A Unified Approach to Dynamic Matching and Barter Exchange

John P. Dickerson

CMU-CS-16-127

September 2016

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:

Tuomas Sandholm, Chair
Avrim Blum
Zico Kolter
Ariel Procaccia
Craig Boutilier, Google
Alvin Roth, Stanford University

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This work was supported by the National Science Foundation under grants IIS-0905390, IIS-096457, IIS-1065251, IIS-1320620, IIS-1350598, IIS-1546752, and CCF-1101668, CCF-1116892, CCF-1215883, CCF-1415460, by the Army Research Office under award W911NF-16-1-0061, by an NDSEG Fellowship awarded through the Army Research Office, a Facebook Fellowship, and a Siebel Scholarship, and used the Extreme Science and Engineering Discovery Environment (XSEDE) at the Pittsburgh Supercomputing Center (PSC), which is supported by National Science Foundation grant OCI-1053575. We also acknowledge Intel Corporation and IBM for machine gifts.

The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government, or any other entity.

Keywords: matching, barter exchange, market design, dynamic matching markets, kidney exchange, combinatorial optimization

To my parents,
for their unwavering support
in my quest to become a doctor
(but not that kind of doctor)

Abstract

The exchange of indivisible goods without money addresses a variety of constrained economic settings where a medium of exchange—such as money—is considered inappropriate. Participants are either matched directly with another participant or, in more complex domains, in barter cycles and chains with many other participants before exchanging their endowed goods. This thesis addresses the design, analysis, and real-world fielding of dynamic matching markets and barter exchanges.

We present new mathematical models for static and dynamic barter exchange that more accurately reflect reality, prove theoretical statements about the characteristics and behavior of these markets, and develop provably optimal market clearing algorithms for models of these markets that can be deployed in practice. We show that taking a holistic approach to balancing efficiency and fairness can often practically circumvent negative theoretical results. We support the theoretical claims made in this thesis with extensive experiments on data from the United Network for Organ Sharing (UNOS) Kidney Paired Donation Pilot Program, a large kidney exchange clearinghouse in the US with which we have been actively involved.

Specifically, we study three competing dimensions found in both matching markets and barter exchange: uncertainty over the existence of possible trades (represented as edges in a graph constructed from participants in the market), balancing efficiency and fairness, and inherent dynamism. For each individual dimension, we provide new theoretical insights as to the effect on market efficiency and match composition of clearing markets under models that explicitly consider those dimensions. We support each theoretical construct with new optimization models and techniques, and validate them on simulated and real kidney exchange data. In the cases of edge failure and dynamic matching, where edges and vertices arrive and depart over time, our algorithms perform substantially better than the status quo deterministic myopic matching algorithms used in practice, and also scale to larger instance sizes than prior methods. In the fairness case, we empirically quantify the loss in system efficiency under a variety of equitable matching rules.

Next, we combine all of the dimensions, along with high-level human-provided guidance, into a unified framework for learning to match in a general dynamic model. This framework, which we coin FUTUREMATCH, takes as input a high-level objective (e.g., "maximize graft survival of transplants over time") decided on by experts, then automatically (i) learns based on data how to make this objective concrete and (ii) learns the "means" to accomplish this goal—a task that, in our experience, humans handle poorly. We validate FUTUREMATCH on UNOS exchange data and make policy recommendations based on it.

Finally, we present a model for liver exchange and a model for multi-organ exchange; for the latter, we show that it theoretically and empirically will result in greater social welfare than multiple individual exchanges.

Acknowledgments

This thesis could not have happened without the constant support, guidance, and occasional tough love of my advisor, Tuomas Sandholm. It has been a pleasure working with Tuomas in pure research, in the commercial arena, and everywhere in between; I've learned as much by just being around him as I have by directly interacting with him. Tuomas invested an incredible amount of time into my personal success and well being not just at CMU, but also in my greater academic career—something that will hopefully continue until I, too, can write "tenured" in front of "professor!" I am especially grateful to him for the late nights spent editing papers over shrimp, jalapeño, and garlic pizza, for allowing me to freely pursue my own research projects both within CMU and with myriad collaborators outside the university, for supporting my travel to more than three dozen workshops and conferences, and—in general—for the constant mentorship and advising over the last six years.

Ariel Procaccia has been like a shadow advisor to me—as I start my career as an assistant professor, I'll be using his trajectory as an (unachievable) set of goal posts. It's clear how much Ariel cares about both research as well as his students and collaborators; that's something I both appreciate and plan to emulate. Zico Kolter shares my interest in optimization and sustainability, and gave valuable advice regarding the stochastic optimization portions of this thesis. Avrim Blum, Craig Boutilier, and Al Roth round out a supportive thesis committee with a staggering amount of expertise in a variety of areas: Avrim's naturally understated presence makes his consistently insightful comments all the more impactful; Craig's grasp on academic AI/ML research and where it slots into practical application in industry is something I hope to rely on in the future; and Al's deep knowledge of both the theory and practice of market design has informed this thesis and my greater research agenda—not to mention a variety of real-world matching markets. I'd especially like to thank Al for his advice and support during my academic job search.

Beyond my thesis committee, I've been fortunate to have many strong mentors and advisors in my past and present. V.S. Subrahmanian is, more than anyone else, the reason I initially pursued an academic career. Working in his lab both during and immediately after my undergraduate degree showed me what a productive, happy, and well-run academic group can be like: his lab had the right mix of individual freedom and direction from on high, older students mentoring younger students, collaboration within and without the lab, and focus on building both theoretically-sound but practically-useful frameworks and systems. V.S. has consistently supported me, from hunting for a graduate school to applying for fellowships to interviewing for academic jobs; on top of this, his informal advice has helped guide much of my career. Prior to this, Amitabh Varshney gave me some of my first "what is academic research" talks during my summer in his lab, and introduced me to a wide variety of interdisciplinary research opportunities. Finally, while an undergraduate, working with Meesh helped me recognize the joy of teaching, and that not all students learn in the same way.

I've been fortunate to share great times, delicious food, and in-depth conversa-

tions with many friends in and around CSD. Listed alphabetically, thank you to: Sarah Allen, for teaching me to appreciate modern CS theory; Alex Beutel, for formally defining data mining over many responsible nights; Noam Brown, for daydreaming with me about a better life in finance; Anca Dragan, for creating deceptive robots and bringing about humanity's downfall; Nico Feltman, for somehow being awake whenever I was, no matter where I was; Sam Gottlieb, for her reverent sense of humor and for teaching me to love eating vegan—kind of; Alex Grubb, for representing Pittsburgh n'at; Nika Haghtalab, for great research in AI (or theory? (or ML? (or AGT?))); Sid Jain, for not taking my bet against Trump—I'm happy to still be solvent; Shiva Kaul, for being on the Pareto frontier; Alex Kazachkov, for discovering a real-world use for p-intersection digraphs—finally—and for the croissants; Elie Krevat, for bringing good taxi service to Pittsburgh; Akshay Krishnamurthy, for the active learning for matching research we're going to do; Jayant Krishnamurthy, for following through on the purchase of great URLs, and helping to run the CSD party house; Christian Kroer, for initiating our healthy gambling habit on paper acceptances—may we both never be stuck in the sand hole; Dave Kurokawa, for being a consistent running partner; Jamie Morgenstern—now, I know I don't have to tell you how committed I am to community service, but hopefully that's over for the both of us—and also for taking academia seriously with me 100% of the time, really; David Naylor, for teaching us about serifs; Abe Othman, for co-founding our successful social-mobile-hyperlocal startup, and for the continuing mentorship; Fei Peng, for writing well-tested and fully-debugged production code with me; Ben Plaut, for being the absolute best undergraduate advisee I can imagine—good luck at Stanford, and thanks for the kidney; Wolf Richter, for calm and constant community leadership (backed up by gainz); Kevin Waugh, for the bass cannon and his zen-like calmness on the soccer pitch; Gabe Weisz, for keeping CSD's beer palate respectable; David Witmer, for his love of natural problems; John Wright, for being a leading exemplar of Johns; and Erik Zawadzki, for How Lee and—in general—a love of food. May we all meet again in Valhalla.

Sharing the Aweffice with top-notch officemates (OMs) like Anna Gommerstadt and Vagelis Papalexakis has been great for life, if not always for work. Vagelis is a champion and an achiever—we'll both crush it. Anna and I will both hit our running goals someday, and then will get a well-deserved brunch. I might even change first.

I cannot overestimate the value of the support received from Deb Cavlovich, Cleah Schlueter, and Charlotte Yano over the last six years. I'd likely have accidentally dropped or been kicked out of the program years ago were it not for Deb's constant guardian angel abilities to make sure everything was in order. I've enjoyed getting to know Cleah over this last year, and thank her especially for handling all of my recommendation letter requests during the academic job market. Charlotte's sense of humor—light, chipper, and never inappropriate—matches well with mine; I've enjoyed the sage insights she's shared about life at CMU, and life in general, and the hearty laughs we've had while staring into the abyss.

In the greater CMU community, Lizzie Silver has redefined what it means to

be a philosopher, and fed me delicious things along the way. Lucy Erickson, Anna Fisher, Rony Patel, and Erik Thiessen have opened my eyes to the world of psychology research—for better or for worse. Computers still make dramatically more sense than humans, at least for now.

Outside of CMU, I've had the pleasure of meeting many smart and fun folks at various stages of their academic careers. Sanmay Das has been a great friend, mentor, and collaborator, especially during my recent academic job search. I look forward to further collaboration with Sanmay and Zhuoshu Li. Nick Mattei and Yonatan Bisk have been excellent conference co-explorers and restauranteers. The ongoing collaboration Tuomas and I have with Chen Hajaj, Avinatan Hassidim, and David Sarne has been a lot of fun; I look forward to interacting with Chen more now that we're both in the same time zone. I'd like to thank David Manlove and James Trimble for hosting an excellent trip to Glasgow, and to James especially for PIEF (pronounced "peef") and PICEF (like "bicep"). That trip also drilled home just how seriously the Scots take their whisky (with no "e")—an important life lesson. Finally, Vanina Martinez, Rob Patro, and Gerardo Simari gave excellent early-stage mentorship to a pre-graduate-school me; thank you all.

Interacting with the United Network for Organ Sharing (UNOS) paired donation pilot program has been a great learning experience; I'd especially like to thank Ruthanne Leishman, Elizabeth Sleeman, Darren Stewart, and the rest of the UNOS KPD Pilot Program staff. Furthermore, thanks to Ross Anderson, Kristiaan Glorie, Xenia Klimentova, Nicolau Santos, and Ana Viana for valuable discussions regarding static kidney exchange clearing, and for making available their kidney exchange software for the purposes of conducting our experimental evaluation in Chapter 3.

I've had the pleasure of maintaining close friendships with a number of people who have, for whatever reason, stuck around for the duration of this program. Katrina "Teammate aka TM" LaCurts has been a constant source of everything from witty one-liners to deep life advice; even if we never live in the same city again, we'll always be on the same team. Thank you, Becca Ahrnsbrak, for the steady stream of cat pictures. Stephen Denney, Omar Hashem, and Geoff Roebuck have been great friends for roughly 20 years, from back in the LAN party and airsoft days. I'm still not convinced we've changed all that much since then.

Finally, none of this could've happened without the constant support of my immediate family. Mom, Dad, Michael, and Jen—thank you all!

Contents

List of Figures

List of Tables

There may be many people wanting, and many possessing those things wanted; but to allow of an act of barter, there must be a double coincidence, which will rarely happen.

– William Stanley Jevons

1 Introduction

The exchange of indivisible goods without money addresses a variety of constrained economic settings where a medium of exchange—such as money—is considered repugnant [181] or otherwise inappropriate. One example setting is the allocation of donor organs to needy patients; while the act of organ donation is laudable, the provision of monetary payments to those who donate is often societally unacceptable and, in nearly all countries, illegal. In general, the removal of a medium of exchange forces markets operating in such settings to rely on the "double coincidence of wants" [121], where two participants must both exist simultaneously and be in possession of a good that the other desires. This is typically quite restrictive, making the design and operation of efficient markets without money theoretically and computationally challenging.

This thesis addresses the design, analysis, and—when appropriate—real-world fielding of matching markets and barter exchanges through the lens of computer science. We focus on the creation of new mathematical models for these markets that more accurately reflect reality, proofs of statements about the characteristics of these markets in theory, and the development of provably optimal clearing algorithms for these models that can be deployed in practice. We show that taking a holistic approach to balancing efficiency and fairness can often practically circumvent negative theoretical results. We support the theoretical claims made in this thesis with extensive experiments on data from the United Network for Organ Sharing (UNOS) Kidney Paired Donation Pilot Program, a large kidney exchange clearinghouse in the US with which we have been actively involved.

Thesis statement

Competing dimensions—equity, efficiency, and computational tractability—in dynamic matching markets and barter exchanges can be balanced holistically through computational optimization methods and informed by random graph models.

The rest of this chapter introduces matching markets and barter exchange (§1.1), gives a brief history of worldwide kidney exchange (§1.2), introduces basic terminology used throughout the thesis (§1.3), and overviews the structure and high-level contributions of this thesis to matching market design (§1.4).

1.1 Matching markets and barter exchange

In matching problems, a central clearinghouse pairs agents with other agents, transactions, or contracts. While such matching—deciding who gets what, and why [182]—has been a central part of human society for millennia, the formal mathematical study of matching markets is only slightly over half a century old, having been initiated by Gale and Shapley [95] in 1962. Since then, however, matching market designers hailing from Economics, Operations Research, and Computer Science have used a growing academic literature on matching to aid in the design and fielding of a variety of markets. For motivation, we now briefly overview recent applications of matching market design, leaving the specification of a formal mathematical model until Section 1.3. For a more in-depth overview of the theory and application areas of traditional matching markets, we redirect the reader to a classic survey by Roth and Sotomayor [183] and to more recent surveys by Abdulkadiroğlu and Sönmez [5], Manlove [153], and Sönmez and Ünver [197].

Two-sided matching markets. In two-sided matching, both sides of the market have preferences over the other side. The classical example of a fielded two-sided matching market is the National Resident Matching Program (NRMP) in the United States; here, graduating medical students are matched as residents to training hospitals with limited capacity [180]. Other fielded classes of two-sided markets include the following:

- School choice [4], where students are matched to schools with limited capacity, like those implemented in Boston [2], New York City [1], and Israel [49];
- Labor markets, where workers are matched to firms (e.g., via an online service like Upwork [214] or more traditional ad hoc application processes); and
- Rideshare and taxi services, where customers in need of transportation match with willing drivers (e.g., through a service like Uber [211]).

Here, as in the markets discussed below, a key goal of the market designer is to promote efficient matching by implementing a mechanism that provides market thickness—e.g., in the rideshare case, ensuring that enough customers participate to make driver-side entrance worthwhile, and vice versa—without too much market congestion [166]. We discuss building thickness in a centralized market in greater depth in Chapters 6 and 8.

Given a set of participants, a designer also cares about handling other forms of short-term uncertainty—e.g., in the labor market case, determining whether a worker and firm truly matched and, if so, the quality of that match—and, finally, actually being capable of computing an efficient matching in an appropriate amount of time. We discuss principled methods to compute efficient matchings—with and without considering such uncertainty—in Chapters 3, 4, 5, and 8.

One-sided matching markets. In one-sided matching, only one side of the market has preferences over the other [195]. This type of market often occurs when agents are being matched with objects. Well-known classes of fielded one-sided matching markets include the following:

- Housing allocation, where humans are assigned to houses [116], or to rooms in public housing projects with limited capacity [127, 128, 205];
- Deceased-donor organ allocation, where needy patients are matched to cadaveric organs that align with their medical needs [37, 56, 202, 223]; and
- Barter exchange, where agents exchange their endowed items by participating in cyclic or chain-based swaps; we discuss this in greater depth below.

In these examples—as in the other matching market types we will discuss—there is a clear need to balance fairness with economic efficiency, a topic we address more rigorously in Chapters 7, 8, and 9.

Barter exchange. In barter exchange, agents enter the market with an initial endowment and then choose to enter into exchanges with other agents. These markets lack the inherent "sidedness" of the two-sided markets above; rather, any participant can—if deemed appropriate by both parties—be matched with any other participant. Numerous barter exchanges are fielded, such as:

- House exchange, where participants seek to swap homes (e.g., Intervac [119] and Best House Swap [38]);
- Room exchange, where college roommates simultaneously "trade up" to better roommates (e.g., The Room Exchange at the University of Maryland [206]);
- Book exchange, where participants swap books after reading them (e.g., Read It Swap It [177]);
- Shoe exchange, where participants who require only a single shoe or two shoes of different sizes due to injury, disease, or genetic disorder can swap shoes with similar participants (e.g., the National Odd Shoe Exchange [164]);
- Shift exchange, where nurses or other shift workers swap shifts (typically an ad hoc process, as seen in ZenDesk's ShiftPlanning software [222]);
- Digital goods exchange, where agents swap digital goods *but retain their original (digital) endowment* even after the exchange occurs (e.g., Factual [92] in the US or Datatang Technology [66] in China, as discussed by Fang et al. [93]); and
- General barter exchange, where participants can swap different classes of goods and services (e.g., Swap.com [203], TradeAway [210], BarterQuest [31], and others).

In this thesis, we focus on the application area of *kidney exchange*, a recent innovation that matches patients in need of a kidney to willing living donors. Chronic kidney disease is a life-threatening health issue that affects millions of people worldwide; its societal burden is likened to that of diabetes [165]. Damage from kidney disease can cause irreparable loss of organ function and, eventually, kidney failure. Such failure requires either continual dialysis or an organ transplant to sustain life. Demand for kidneys is far greater than supply [106]. Although receiving a deceased-donor kidney is a possibility, as of mid-2016, there are roughly 100,000 people on the US national

waiting list for kidneys,[1] with 35,038 added in 2015 while only 17,881 left due to receiving a new organ—from either a deceased or living donor. *Kidney exchange* allows patients who suffer from terminal kidney failure, and have been fortunate enough to find a willing but incompatible kidney donor, to swap donors. Indeed, it may be the case that two donor-patient pairs are incompatible, but the first donor is compatible with the second patient, and the second donor is compatible with the first patient; in this case a life-saving match is possible. As we discuss formally in Section 1.3, sequences of swaps can even take the form of long cycles or chains.

1.2 A brief history of kidney exchange

The idea of kidney exchange was presented in 1986 by Rapaport [176]; important ethical discussion was given by Ross et al. [179]. South Korea performed the first kidney exchange in 1991 [140], followed by the first European exchange (in Switzerland) in 1999 [207]; these swaps were organized on a small scale and by hand. The first organized exchange market for kidneys, the New England Paired Kidney Exchange (NEPKE), started in 2003–2004 in the US [184, 185, 186], while the first "nationwide" exchange was launched in the Netherlands in 2004 [67]. NEPKE has since ceased operations; its pool was merged into the United Network for Organ Sharing (UNOS) US-wide kidney exchange, which started in 2010 and now includes 66% of the US transplant centers.

There are also two large private kidney exchanges in the US, the National Kidney Registry (NKR) and the Alliance for Paired Donation (APD). They typically only work with large transplant centers. Transplant centers can be part of multiple exchanges; Appendix E presents preliminary results quantifying the effect on social welfare of this multi-registration and competition between exchange clearinghouses. NKR makes their matching decisions manually and APD uses a combination of algorithmic and manual decision making. There was also another large private kidney exchange, the Paired Donation Network (PDN), which has ceased operations. In addition, there are several smaller private kidney exchanges in the US. They typically only involve one or a couple of transplant centers. These include an exchange at Johns Hopkins University and a single-center exchange at the Methodist Specialty and Transplant Hospital in San Antonio.

Furthermore, there are now established kidney exchanges in the Netherlands, Canada, the United Kingdom, Portugal, Australia, and Israel. International exchanges have been performed on a case-by-case basis; recent work by Akbarpour et al. [10] investigates a theoretical model of financing international exchange, but a fielded implementation of a large-scale organized international exchange is not on the immediate horizon.

Kidney exchanges started with just using 2-cycles before also allowing 3-cycles and altruist-initiated chains [187]. Since 2006, kidney exchanges have also incorporated never-ending chains, where the last donor in a chain serves as an altruist in a later match run to initiate a new chain [178]. This approach is now included at least in the three leading kidney exchanges (UNOS, NKR, and APD). Chapter 3 covers the methods used to clear exchanges under these different constraints, as well as the effect on economic efficiency of restrictions on matching structures.

[1] http://optn.transplant.hrsa.gov.

A brief overview of theoretical work in kidney exchange. Roth et al. [184, 186, 188] set the groundwork for large-scale organized kidney exchange. These papers explored what efficient matchings in a steady-state kidney exchange would look like; extensions by Ashlagi et al. [23], Ashlagi and Roth [19], Sönmez and Ünver [199], and Ding et al. [82] address shortcomings in those theoretical models that appeared as kidney exchange became reality. We overview this literature more and then augment those models substantially in Chapters 2, 3, 4, 5, 7, 9, and Appendix E. Abraham et al. [6], Biró et al. [41], Manlove and O'Malley [152], Glorie et al. [100], Constantino et al. [63], Anderson et al. [16], and Klimentova et al. [134] discuss complexity results related to the clearing problem and present practical approaches to optimally clearing actual exchanges; we overview this literature more and present novel contributions in this space in Chapters 3, 4, 5, 6, and 8. Game-theoretic models of kidney exchange, where transplant centers are viewed as agents with a private type consisting of their internal pools, were presented and explored by Ashlagi and Roth [19], Toulis and Parkes [209], Ashlagi et al. [25], Carvalho et al. [55], and Blum et al. [45]; we have ongoing work in this area with Hajaj et al. [104], but do not include it in this thesis. Various forms of dynamism, like uncertainty over the possible future participants in the pool [9, 24, 213] or uncertainty over the existence of particular potential transplants [14, 15, 43, 44, 75] have been explored from both an economic and algorithmic efficiency point of view; we discuss this literature in greater depth and our contributions to this space in Chapters 5, 6, and 8.

1.3 Preliminaries

In this section, we introduce basic mathematical terminology and a model of barter exchange that will be used throughout this thesis. Most chapters augment the model in this section. As is common throughout this thesis, we present the model as a kidney exchange, where patients bring willing but incompatible donors to a large waiting pool; however, this model applies to any general barter exchange where agents enter with an endowment.

We begin by describing the creation of a *compatibility graph* representing the space of possible swaps among n agents (patients), each with an endowed item (willing donor). The construction of the graph is based on traits of the candidates and donors. We then describe the *clearing problem*, a formalization of the process used to determine an optimal set of swaps.

1.3.1 Compatibility graph

We begin by encoding an n-patient kidney exchange as a directed graph $G(n)$. Construct one vertex for each incompatible candidate-donor pair. Add an edge e from one candidate-donor vertex v_i to another v_j, if the candidate at v_j can take a kidney from the donor at v_i. (Edges e can also be associated with weights w_e, representing the utility to v_j of obtaining v_i's item.)

Within the compatibility graph, a cycle c represents a possible swap, with each vertex in the cycle obtaining the item of the next vertex. A matching is a collection of *disjoint* cycles; no vertex can give out more than one item (e.g., more than one kidney). Cycles ensure that donors give items if and only if their patients receive organs. Fielded kidney exchanges also gain great utility

through the use of *chains* [22, 96, 178]. An altruistic donor initiates a chain by donating his organ to a patient, whose paired donor donates her organ to another patient, and so on.

Figure 1.1 gives an example kidney exchange compatibility graph. Possible cycles include, for example, the 2-cycles $\langle (d_2 \to p_3), (d_3 \to p_2) \rangle$ and $\langle (d_5 \to p_6), (d_6 \to p_5) \rangle$, or the longer 3-cycle $\langle (d_2 \to p_5), (d_5 \to p_3), (d_3 \to p_2) \rangle$. Additionally, a single altruistic donor a exists in the pool and is willing to give his or her kidney to a patient, whose paired donor will then donate a kidney to a compatible patient in the pool (for example, via the chain $\langle (a \to p_1), (d_1 \to p_4), (d_4 \to p_7), (d_7 \to \cdot) \rangle$, with the final donor d_7 either donating to the deceased donor waiting list or remaining in the pool as a future altruistic donor).

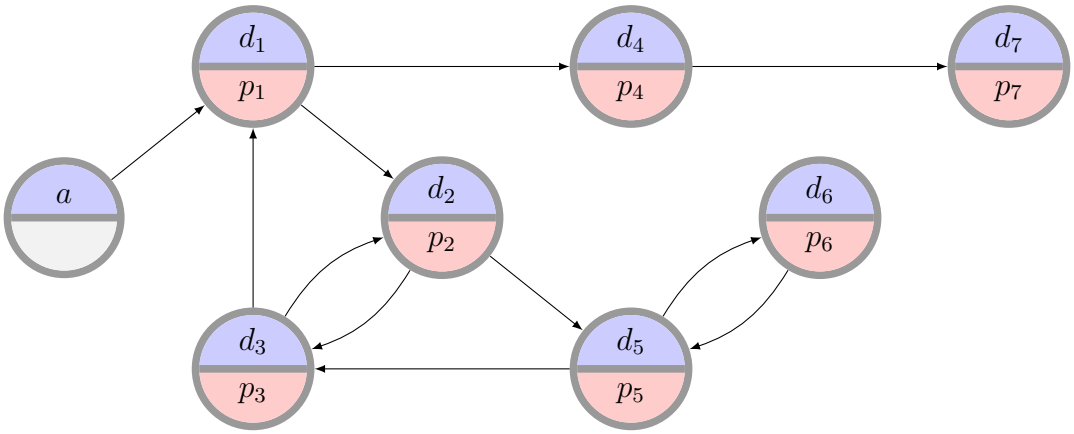


Figure 1.1: An example compatibility graph.

1.3.2 The clearing problem

The *clearing problem* is that of finding a maximum-cardinality matching consisting of disjoint chains—possibly with some length cap K—and cycles of length at most some small constant L. The cycle-length constraint is crucial since all operations in a cycle have to be performed simultaneously. Were this not the case, a donor might back out after his incompatible partner has received an organ. This backing out is legal because, in nearly all countries including the US, it is illegal to form a binding contract over the exchange of organs. The availability of operating rooms, doctors, and staff causes long cycles to be unexecutable. As is the practice in the US-wide kidney exchange and most other real kidney exchanges, we let $L = 3$. Chains need not be limited in length; were a donor to renege before giving an organ but after his paired patient had received the organ, then no remaining pair in the pool has lost its "bargaining chip"—although the collapse of the chain is not desired.

Lower length caps on both cycles and chains may have practical advantages. For example, shorter cycles are less likely to fail after the algorithmic matching but before transplantation, and thus may lead to improved matching in practice [75, 147]. Some fielded kidney exchanges explicitly favor shorter cycles over longer ones [152]. We address this dimension of kidney exchange in Chapter 5, but do not discuss it further in this section.

In the small example compatibility graph shown in Figure 1.1, with $L = 3$, a maximum cardinality matching *without* chains (i.e., with $K = 0$) includes five pairs via the 3-cycle and 2-cycle:

$$\{\langle(d_1 \to p_2), (d_2 \to p_3), (d_3 \to p_1)\rangle, \langle(d_5 \to p_6), (d_6 \to p_5)\rangle\}.$$

With chains of unbounded length (i.e., by letting K increase with $|V|$) the cardinality of the maximum matching increases to seven pairs. This is achieved by starting a chain by way of the altruist a, as well as by using the same two 2-cycles as before:

$$\{\langle(a \to p_1), (d_1 \to p_4), (d_4 \to p_7), (d_7 \to \cdot)\rangle, \langle(d_2 \to p_3), (d_3 \to p_2)\rangle, \langle(d_5 \to p_6), (d_6 \to p_5)\rangle\}.$$

Quantifying the gain in match size due to the addition of altruistic donors, both in terms of a random graph model and via realistic simulation, is addressed in Chapters 2; methods for clearing exchanges in practice in the presence of long but finite chains are discussed in Chapters 3, 4, and 5.

While the clearing problem can be solved easily on very small graphs like that in Figure 1.1, it quickly becomes intractable on larger graphs. The standard algorithmic method for optimally clearing kidney exchanges is integer programming [188]; we discuss methods for clearing in much greater depth in Chapter 3, but briefly overview a basic approach here. Formally, denote the set of all chains of length no greater than K and all cycles of length no greater than L by $\mathcal{C}(L, K)$. Let $|c|$ represent the number of candidate-donor pairs in a cycle or chain c. Then, given binary indicator variables $x_c \in \{0, 1\}\ \forall c \in \mathcal{C}(L, K)$, we must solve the following integer linear program:

$$\max \sum_{c \in \mathcal{C}(L,K)} |c|\, x_c \qquad s.t. \qquad \sum_{c: v \in c} x_c \leq 1 \quad \forall v \in V$$

The clearing problem with any fixed $L > 2$ is NP-complete [6, 41], APX-hard [40], and is constant-factor inapproximable [150]. The cases $L = 2$ with no chains and $L = \infty$ can be solved in polynomial time. Significantly better (i.e., higher cardinality) results are found with $L = 3$ over $L = 2$, so solving the NP-complete version of the problem is necessary in practice [188]. The problem, at least with respect to kidneys, can be solved optimally in practice at the steady-state nationwide scale using a specialized tree search algorithm based on the branch-and-price framework for integer programming [6, 75, 80]. That is one major contribution of this thesis and is discussed in greater depth in Chapter 3.

More generally, define a *matching* M to be any collection of disjoint cycles and chain—possibly obeying length restrictions or other arbitrary business constraints—in the graph G. Then, given the set of all legal matchings $\mathcal{M}$, the general batch clearing problem is to find a matching $M^* \in \mathcal{M}$ that maximizes some utility function $u : \mathcal{M} \to \mathbb{R}$. Formally:

$$M^* = \arg\max_{M \in \mathcal{M}} u(M)$$

We attend to this more general version of the clearing problem in Chapters 5 and 7; we also extend this further to the dynamic setting, where the clearinghouse wishes to maximize some objective over time, in Chapters 6 and 8. Next, we overview the general structure of this thesis and its contributions.

1.4 Contributions & structure of this thesis

In this section, we discuss the structure of this thesis, along with the contributions of each chapter and the overall contributions of the methods presented in the thesis. The thesis is divided into five major parts.

Clearing deterministic exchanges. Part I of this thesis consists of Chapters 2, 3, and 4 and addresses the clearing problem in static, deterministic barter exchanges. A clearinghouse that does not consider the future when matching in the present is said to solve the *static* clearing problem; we relax this myopic outlook in Chapters 6 and 8. Similarly, a clearinghouse that does not consider uncertainty over the existence of market participants, nor uncertainty over the existence of potential trades between participants, is said to solve the *deterministic* clearing problem; we relax this outlook in Chapter 5. Yet, solving the static, deterministic clearing problem is of both theoretical interest and practical importance, as it most accurately mimics the current reality of most fielded kidney exchanges' matching policies.

Chapter 2 begins by presenting the first of two *random graph* models of kidney exchange; it augments the model of Ashlagi and Roth [19] to include altruistic donors, and then determines a matching that only requires short cycles and chains to achieve global efficiency (in the large, with high probability). Experimental evidence in the initial generative model of kidney exchange due to Saidman et al. [191] supports this theoretical intuition. This theoretical result would hint that simple matching strategies could be used in a large fielded kidney exchange, instead of needing to rely on more elaborate mathematical programming techniques; yet, since publication, the reality of kidney exchange has been shown to align with neither that initial theoretical model nor the initial simulators. Chapter 2 shows on real data from the UNOS exchange that long chains can be helpful in the static, deterministic setting; this is now an accepted viewpoint in the kidney exchange community, and motivates the following two chapters' focus on optimization methods for clearing exchanges with longer chains.

Chapter 3 addresses methods for solving the static, deterministic clearing problem to optimality. It begins by overviewing previous approaches, most of which are integer-programming-based solvers relying on either *constraint generation* or *branch and price*. It goes on to identify a bug in the previously leading branch-and-price-based solvers and proves a hardness result showing that these papers' approaches are likely not to work in general. To counter this negative result, it presents new models for kidney exchange that scale *substantially* better than the prior approaches when clearing exchanges with long, finite chains—including the now provably non-optimal prior leading solvers! Some of these new models are compact (polynomial with respect to the input graph), while others are not; for the latter models, it presents a branch-and-price-based approach such that the pricing problem is solved in polynomial time. It also presents various theoretical results comparing the tightness of linear programming relaxations in different models.

Chapter 4 presents a novel model for kidney exchange in which, under a light assumption that is true in reality, the clearing problem is solvable in polynomial time. While the polynomial-time algorithm presented in that chapter is not yet empirically tractable in practice, the new model is quite general and can be used to address some recent advances in kidney exchange—for example,

the idea that patients and donors can pay some monetary or health cost to either temporarily or permanently change (e.g., via immunosuppression) their initial attributes to increase the likelihood of finding a match. We also provide experimental validation that our new model can accurately mimic real kidney exchange.

Contribution

- Some prior state-of-the-art clearing engines were not correct; indeed, their approaches are likely intractable by way of a new hardness result;
- We present the most scalable optimal clearing engine for barter exchange with short cycles and long but finite chains, which accurately mirrors fielded exchanges; and
- A new model for kidney exchange can be used to computationally efficiently clear exchanges, and can model new medical developments in kidney exchange.

Coauthors and relevant publications: Dickerson, Procaccia, and Sandholm [74], Dickerson, Kazachkov, Procaccia, and Sandholm [79], Dickerson, Manlove, Plaut, Sandholm, and Trimble [80], Plaut, Dickerson, and Sandholm [174, 175]

Managing uncertainty in dynamic matching environments. Part II of this thesis consists of Chapters 5 and 6 and relaxes the "static" and "deterministic" descriptors on the clearing problem addressed in Part I. In practice, kidney exchange is *dynamic*, with vertices and edges in the compatibility graph arriving and departing over time; we address this in Chapter 6. Similarly, the actual existence of edges in a compatibility graph is not known with perfect certainty; indeed, patients' and donors' attributes change over time and are also tested at variable levels of detail and accuracy. Chapter 5 addresses this stochasticity from two complementary points of view.

Chapter 5 presents the *failure-aware* model of kidney exchange, which explicitly considers edge failure probability in the clearing optimization problem. It shows that we can significantly increase the number of successfully matched vertices (i) in theory, in a sparse random graph model due to Ashlagi et al. [23]; (ii) on real data from kidney exchange match runs between 2010 and 2014; (iii) on synthetic data generated via a model of dynamic kidney exchange. It presents a novel branch-and-price-based optimal clearing algorithm specifically for this probabilistic exchange clearing problem and shows that this new solver scales well on large simulated kidney exchange data, unlike prior clearing algorithms. It shows experimentally that taking failed parts from an initial match and instantaneously *rematching* them with other vertices still in the waiting pool can result in significant gains. It also shows that the compact formulations of Chapter 3 can be extended to the failure-aware setting under a simple assumption, and presents a polynomial-time branch-and-price-based solver in this model.

Following the presentation of the failure-aware model, Chapter 5 then addresses a complementary approach for dealing with uncertainty in the existence of edges via *pre-match* edge testing. Here, the problem is to identify some small set of edges to test for existence prior to the match run such that the proposed algorithmic match will—after edge failures—be close in size to the omniscient match that knows in advance the existence of all potential edges. Cast as a stochastic

matching (for the 2-cycles-only case) or stochastic k-set packing (for cycles and chains with a maximum size of k) problem, experiments show that even a very small number of pre-match rounds of edge testing—where at most one incoming edge per vertex can be tested per round—would result in tremendous practical efficiency gains.

Chapter 6 addresses the *dynamic* barter exchange problem, where vertices and edges arrive and depart over time, and the clearinghouse's problem is to maximize some utility function in the long run, possibly at the expense of immediate utility. In this chapter, we introduce a natural, general policy parameterization approach and techniques for operationalizing it. Specifically, we propose to learn *potentials* of elements (e.g., vertices, edges, cycles, and so on) of the current problem. The potential represents an estimate of how much that element can contribute to the objective in the future. The potentials can be viewed as policy parameters to be optimized using a black box program; in the experiments of Chapter 6 and, later, Chapter 8, we learn them using parameter tuning [113, 114]. Then, at run time, we simply run any offline matching algorithm (e.g., [6, 14, 16, 63, 75, 78, 80, 100, 134, 147, 174]) at each time period, but subtracting out in the objective the potentials of the elements used up in the matching. This causes the batch optimizer—which is traditionally myopic, as discussed in Part I—to take the future into account without suffering a run-time cost.

While potentials are a general technique for learning to match, we apply them to the kidney exchange domain. We theoretically compare the power of using potentials on increasingly large elements: vertices, edges, cycles, and the entire graph (optimum). Then, experiments show that by learning vertex potentials, our algorithm matches more patients than the current practice of clearing myopically. It scales to exchanges orders of magnitude beyond those handled by prior algorithms for this (unsimplified) dynamic problem. (Potentials later form a large part of the FutureMatch framework, presented in Part III.)

Contribution

- Uncertainty over the existence of potential trades in exchanges should be taken into consideration explicitly in the clearing optimization;
- Even a small amount of pre-match edge existence testing would result in tremendous efficiency gains in practice; and
- Considering long-term exchange dynamics via learning potentials for graph elements results in further efficiency gains.

Coauthors and relevant publications: Blum, Dickerson, Haghtalab, Procaccia, Sandholm, and Sharma [44], Dickerson, Procaccia, and Sandholm [73, 75]

Balancing equity and efficiency in dynamic matching environments. Part III, consisting of Chapters 7 and 8, addresses the effect on various notions of fairness that the efficiency gains from Parts I and II exert on the overall match structure. Chapter 7 focuses on improving access to kidneys for highly-sensitized, or hard-to-match, patients. Toward this end, we formally adapt a recently introduced measure of the tradeoff between fairness and efficiency—the *price of fair-*

ness [35, 53]—to the standard kidney exchange model. We show that the price of fairness in the standard theoretical model—that model due to Ashlagi and Roth [19] and augmented previously in Chapter 2—is small. We then introduce two natural definitions of fairness and formally define them in the standard *deterministic* model of kidney exchange, and in the failure-aware model presented in Chapter 5. The chapter concludes with an empirical exploration of the tradeoff between matching more hard-to-match patients and the overall utility of a utilitarian matching, on data from the UNOS nationwide kidney exchange and simulated data from each of the standard kidney exchange distributions.

Chapter 8 presents FUTUREMATCH, a framework for learning to match in a general dynamic model. FUTUREMATCH takes as input a high-level objective (e.g., in the context of kidney exchange, "maximize graft survival of transplants over time") decided on by experts, then automatically (i) learns based on data how to make this objective concrete and (ii) learns the "means" to accomplish this goal—a task, in our experience, that humans handle poorly. Chapter 8 gives an instantiation of FUTUREMATCH that uses data from all live kidney transplants in the US since 1987 to learn the quality of each possible match; it then learns the potentials of elements in the graph (as discussed in Chapter 6), translates these to weights, and performs a computationally feasible batch matching that incorporates dynamic, failure-aware considerations through the weights (using the failure-aware solver presented in Chapter 5). We validate FUTUREMATCH on UNOS exchange data. It results in higher values of the objective. Furthermore, even under economically inefficient objectives that enforce equity, it yields better solutions for the *efficient* objective (which does not incorporate equity) than traditional myopic matching that uses the efficiency objective.

Contribution

- It is possible to theoretically and empirically quantify tradeoffs between "fair" and "efficient" matching rules;
- In kidney exchange, the "price" to using a fair rule can be quite low; and
- FUTUREMATCH is a system that learns to match in a dynamic setting, taking into account high-level value judgments from human experts while automatically creating a low-level matching policy.

Coauthors and relevant publications: Dickerson and Sandholm [70, 71], Dickerson, Procaccia, and Sandholm [77, 78]

New paradigms for barter exchange. Part IV, consisting of Chapter 9, addresses novel directions in generalized organ exchange. While fielded kidney exchanges see huge benefit from altruistic kidney donors, a significantly higher medical risk to the donor deters similar altruism with livers. This chapter explores the idea of large-scale *liver* exchange, and shows on demographically accurate data that vetted kidney exchange algorithms can be adapted to clear such an exchange at the nationwide level. It then proposes cross-organ donation where kidneys and livers can be bartered for each other. In two adaptations of random graph models—the first due to Ashlagi and Roth [19] (augmented in Chapters 2 and 7) and the second due to Ashlagi et al. [23] (used in Chap-

ter 5)—it shows theoretically that this *multi-organ* exchange provides linearly more transplants than running separate kidney and liver exchanges; that linear gain is a product of altruistic kidney donors creating chains that thread through the liver pool. We support this result experimentally on demographically accurate multi-organ exchanges.

Contribution

- From a technology perspective, we are ready to field a liver exchange; and
- Combining kidney and liver exchanges into a *multi-organ* exchange would result in substantial gains in theory and in practice relative to separate exchanges.

Coauthors and relevant publications: Dickerson and Sandholm [72]

Conclusions, open problems, and the appendices. We conclude the thesis with Part V, which consists of conclusions and closing thoughts in Chapter 10 and a list of open problems in Chapter 11. Both theoretical and experimental open problems are given for general matching, general barter exchange, and—specifically—for kidney exchange. For each open problem, we provide a list of good "starting papers" that an interested researcher could read to get a good foothold on the history and current state of the art of each problem.

This thesis also includes five appendices. Appendix A contains additional theoretical results, proofs, and tabulated experimental data for the position-indexed formulations of barter exchange presented in Chapter 3. Appendix B gives the complete, formal proof of one of the main theoretical results from Chapter 4. Appendix C provides additional experimental results for the adaptive prematch edge testing techniques described in Chapter 5. Appendix D gives details on the generator we created to seed our realistic liver and multi-organ exchange experimental results in Chapter 9. Finally, Appendix E details ongoing work with Sanmay Das and Zhuoshu Li at Washington University in St. Louis and Tuomas Sandholm at CMU; here, we extend a recent theoretical model due to Akbarpour et al. [9] of a single dynamic matching market to the case where *two* markets compete for agents who potentially co-register in both markets. This is motivated by, e.g., overlapping rideshare markets like Uber and Lyft, or overlapping kidney exchanges like APD, NKR, and UNOS in the United States. Our interest in this ongoing work is to quantify the effect on social welfare that this competition creates; we show for a portion of the parameterization space of our model that the loss in social welfare can be high relative to a single clearinghouse.

PART I:

Optimization Methods for Optimal Batch Clearing of Static, Deterministic Exchanges

All models are wrong—but some are useful.

– George Box

A map is not the territory it represents, but, if correct, it has a similar structure to the territory, which accounts for its usefulness.

– Alfred Korzybski

2

Efficient clearing via random graph models

In this chapter, we address the clearing problem by way of a stylized random graph model of kidney exchange. Specifically, we extend the first random graph model of kidney exchange, due to Ashlagi and Roth [19], to include altruistic chains, a recent innovation for barter exchanges that has been widely adopted for kidneys, but was—and, in many ways, still is—poorly understood. Indeed, the research presented in this chapter set policy at the UNOS kidney exchange, effective starting in 2012; however, many aspects of the effect on long-term efficiency of chains are still not well understood from a theoretical point of view (with notable recent work by Anderson et al. [15], Ashlagi et al. [23], and by the author [75], to be discussed in Chapter 5). We include this chapter to motivate the need for non-trivial mathematical-programming-based approaches to clearing real kidney exchanges as a complement to relying on purely theoretical analysis, an interaction we discuss in greater depth throughout the rest of this thesis.

Section 2.1 reports results from early match runs in 2011 at the UNOS kidney exchange; these real-world results clearly show the benefit of integrating chains into the clearing process. Section 2.2 formalizes the theoretical benefit of chains as a kidney exchange scales to the large, and Section 2.3 experimentally determines exactly what "large" means—critically, both of these models operate in a *dense* stylized model of kidney exchange that was popular early on [191], but turned out to not mimic the present-day reality of fielded kidney exchange. Section 2.4 studies the dynamics of kidney exchange over time, using an extension over the prior state-of-the-art dynamic model to more accurately represent the realities of modern kidney exchange. We conclude with some thoughts on the implications of these theoretical and experimental results, as well as areas where they did not end up aligning with the current reality of kidney exchange. The rest of this thesis focuses on building more accurate theoretical representations of kidney exchange, as well as on scalable techniques to learn implementable matching policies that truly mimic reality.

Related Publications

An early version of this work appeared at AAMAS-12 via a collaboration between Dickerson, Procaccia, and Sandholm [74].

2.1 Early results from the UNOS exchange

The UNOS nationwide kidney exchange pilot went live with 77 transplant centers in October 2010 and initially matched using only 2- and 3-cycles. Starting in May 2011, chains were incorporated into the UNOS pilot program [163, 178, 187]. As discussed in Section 1.3, each chain is initiated by an *altruistic* donor—that is, a donor who enters the pool, without a candidate, offering to donate a kidney to any needy candidate in the pool. Chains start with an altruist donating a kidney to a candidate, whose paired donor donates a kidney to another candidate, and so on. Chains can be longer than cycles in practice because it is not necessary (although desirable) to carry out all the transplants in a chain simultaneously; indeed, unlike in a cycle, if a chain breaks by some donor backing out, the chain merely stops, but no patient-donor pair is out their "bargaining chip" (donor kidney). A major question in fielded kidney exchanges is to what extent planning for longer chains at each match run affects short- and long-term efficiency.

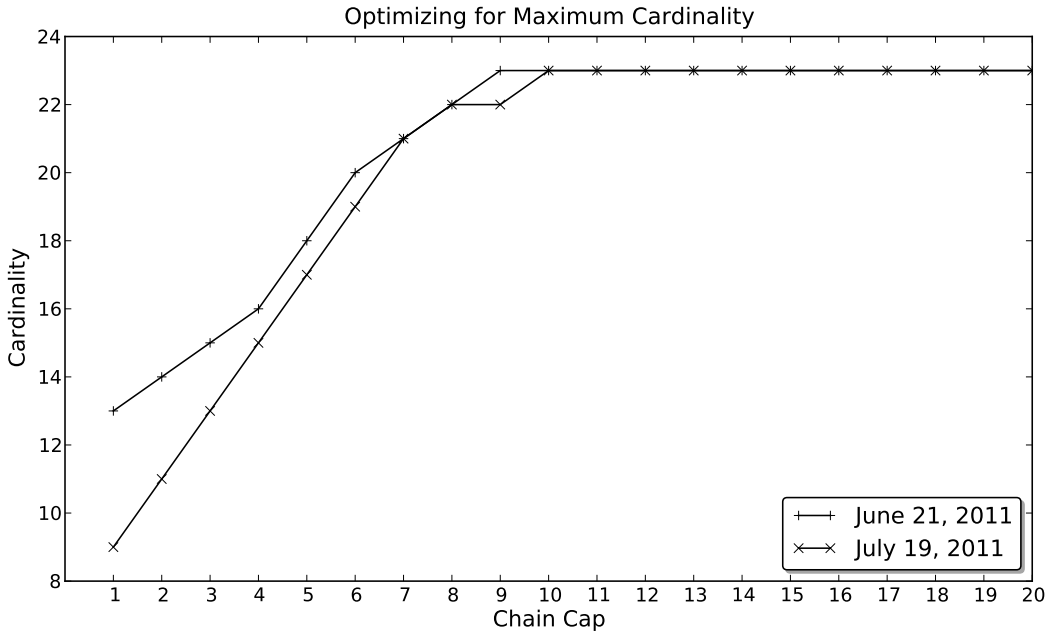


Figure 2.1: Real data from the June/July 2011 UNOS match runs, optimized for maximum cardinality.

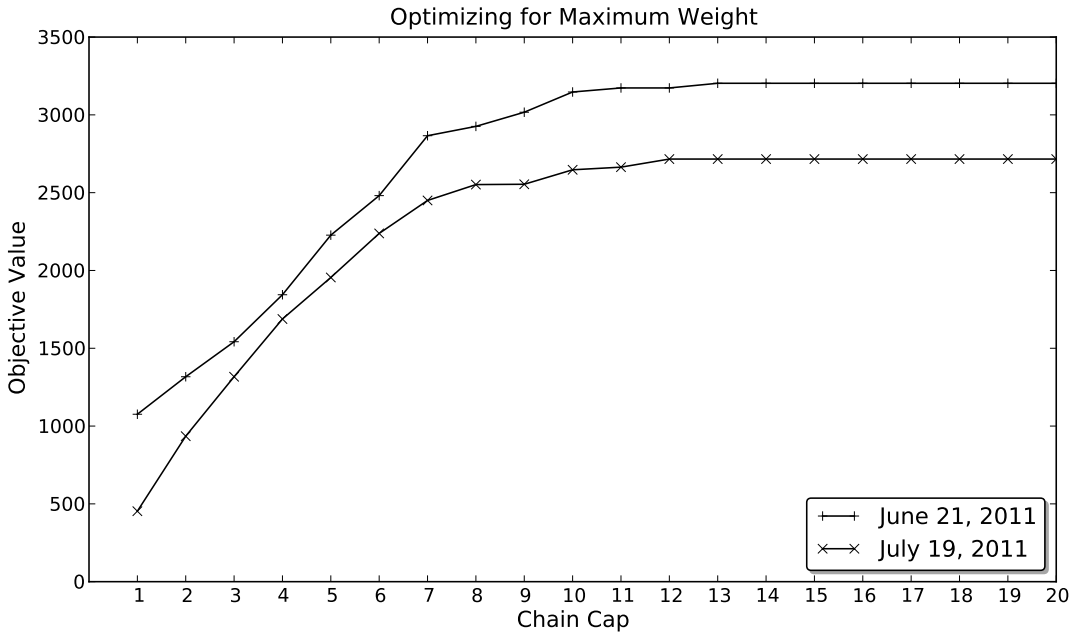


Figure 2.2: Real data from the June/July 2011 UNOS match runs, optimized for maximum total weight.

Figures 2.1 and 2.2 show results for two real matches, for June and July 2011. To show the efficacy of chains, we varied the chain cap from 1 (i.e., the altruistic donor donates directly to the deceased waiting list) to 20. In Figure 2.1, we maximize the cardinality of the final matching. That is, we ignore edge weights and assume all compatible matches are equally good, and determine the matching that allocates kidneys to the most candidates. The size of the matching increases significantly with chains up to length 9 (June) or 10 (July). Critically, with long chains we match 1.77 (June) and 2.55 (July) times the number of candidates than would have been matched with 2− and 3-cycles alone. We note that Ashlagi et al. [23] independently report similar findings from real-world data sets from the NKR exchange.

The improvement from long chains is even more drastic when the edge weights are taken into account, as is the case in the real UNOS match run. Figure 2.2 shows that in June of 2011, chains of length up to 13 increase the objective value, while chains of length up to 12 increase the objective of the matching in July. Overall, incorporating chains increases the objective value to 2.98 (June) and 6.00 (July) times that of chains only (with a cycle cap of 3).

It is important to note that the structure of the compatibility graph, $G(n)$ in this early pilot program is special and, in many ways, computationally fortuitous—especially in the very early months of the UNOS exchange, when the pool was still quite small. The UNOS pool at that time—as well as now—consisted mainly of *highly-sensitized* patients—that is, patients that are difficult to match based on their tissue type. Intuitively, these patients were too hard to match regionally and in prior runs of the national exchange—so the input graph is very sparse. At the time, with

a less sensitized and thus denser pool, clearing engines did not exist that could even solve the current problem size (with long chains) because the input graph $G(n)$ is not as sparse. (This is no longer a problem due to work we will present in Chapter 3). Luckily, in the next section, we show theoretical results stating that in large kidney pools drawn from the full set of candidates (i.e., not just highly sensitized ones), long chains will have negligible effect on the overall cardinality of the matching with high probability. Therefore, one may not need to consider long chains in the clearing—*if* the pool ends up being drawn this way.

If sufficient to achieve efficiency, short chains would be further desirable in practice because short chains are (1) computationally dramatically more tractable for the clearing algorithm (there are fewer of them), (2) logistically easier to administer, and (3) less likely to fail due to a positive crossmatch or some non-simultaneous donor backing out (these two issues will be discussed later in this chapter, and in much greater depth in Part II).

2.2 Theoretical bounds on chains in a dense model

In this section, we prove that using chains of length more than 3 provides no benefit in large, dense, random, unweighted candidate pools. We will prove this result in an adaptation of the first common model of kidney exchange due to Ashlagi and Roth [19], based on a 2006 paper from the medical community due to Saidman et al. [191]. We begin by describing the model.

2.2.1 Necessary background & the "dense" model of kidney exchange

The need for kidney exchange exists due to the myriad of immunological incompatibilities that can be present between a candidate and any potential donor. For instance, the *blood type* of a donor kidney can result in acceptance or outright rejection in a possible candidate. At a high level, human blood is split into four types—O, A, B, and AB—based on the presence or absence of the A and B proteins. While other complications may arise, a type O kidney can be transplanted into any candidate; type A and B kidneys can be transplanted into A and B candidates respectively, or an AB candidate; and type AB kidneys are limited to only type AB candidates. Therefore, some candidates are more difficult to match with a random donor than others. O-candidates are the hardest to match because only O-type kidneys can be given to them. Similarly, O-donors are the easiest to match.

With this in mind, candidate-donor pairs in the matching pool can be labeled based on their blood types using the *ABO model*; it is one of two *de facto* models for theoretical market design work on kidney exchange (see, e.g., [19, 20, 54, 78, 209, 213]). An *under-demanded* pair is any pair such that the donor is not ABO-compatible with the candidate. Furthermore, if these pairs contain only type A and B blood (e.g., the candidate is type A and the donor is type B), the pair is called *reciprocal*. Any pair in the pool such that the donor is ABO-compatible with the candidate is called *over-demanded*. Furthermore, if a donor and candidate share the same blood type, they are a *self-demanded* pair. Intuitively, under-demanded and reciprocal pairs are "harder" to match than over-demanded and self-demanded pairs. In the ABO model, all compatible transplants are

considered to be equally good (i.e., those edges have weight 1 each) and typically results in the ABO model are derived in the limit, when the number of pairs of each kind approaches infinity.

If blood type compatibility were the only requirement for a successful kidney donation, over-demanded and self-demanded pairs would have no need to enter the exchange pool because they could simply conduct the transplant within the pair. However, further complications force their hand: the people in a pair are usually incompatible due to tissue type. Tissue type, in particular what is known as HLA type, is measured as a combination of six proteins. Each potential candidate and potential donor must be tested for preformed antibodies against these six proteins; this needs to be done at least once a month because the antibody state of a person changes over time. An increase in the mismatches between donor and candidate HLA types decreases the likelihood of a successful kidney transplant, and can render a donor and candidate incompatible. These kinds of blood tests where measurements are taken separately from the donors and the patients are called *virtual crossmatch* for reasons that will become obvious in the next paragraph.

An important challenge is that medical knowledge is incomplete: even if a patient and donor are compatible based on the virtual crossmatch (so there is an edge in the input graph), in reality they might not be compatible (i.e., the edge might not be usable). This is determined days before the operation by conducting a test called a *crossmatch*: blood from the patient and blood from his/her planned donor are mixed together and if the mixture coagulates, they are incompatible. Such an unfortunate, but very common, occurrence is called a *positive crossmatch*. Positive crossmatch-sensitive models have only recently begun to appear in the literature, and had not included a study of chains at the time [19, 43, 82, 209]; later work by Ashlagi et al. [23] and by the author [75] (discussed in Chapter 5) did.

We will say that if an altruist donates directly to the deceased-donor waiting list, that constitutes a chain of length 1. If an altruist donates to a pair, whose donor donates to the deceased-donor waiting list, that constitutes a chain of length 2. If an altruist donates to a pair, whose donor donates to a pair, whose donor donates to the waiting list, that constitutes a chain of length 3, and so on. We are now ready to prove the main theoretical result of this chapter.

2.2.2 Short chains suffice (in theory)

In this section, we use the canonical dense model for generating kidney exchange data [19]. It works as follows. We start with $G(n)$, a large compatibility graph representing a kidney exchange as described above. The set of n incompatible patient-donor pairs is partitioned into subsets $V_{X\text{-}Y}$ of type X-Y, for each combination of blood types X and Y of the patient and donor respectively. For each blood type X we denote the set of altruistic donors with that blood type by V_X, but make no assumptions about the size of these sets. We assume that a donor and a patient who are blood type compatible are tissue type incompatible with constant probability $\bar{\gamma}$, corresponding to the virtual crossmatch described above. The frequency of each blood type X is denoted by μ_X.

We are now ready to state our main theoretical result. It extends the recent results of Ashlagi and Roth [19] to the setting with chains.

Theorem 1. *Assume that $\bar{\gamma} < 2/5$, $\mu_O < 3\mu_A/2$, and $\mu_O > \mu_A > \mu_B > \mu_{AB}$. Then with high probability $G(n)$ has an efficient allocation (i.e., one that saves as many patients as possible) that uses only cycles of length at most 3 and chains of length at most 3.*

The proof follows from three lemmas. The first lemma is a trivial simplification and extension of Lemma 9.5 of Ashlagi and Roth [19], which is a generalization of a classic theorem by Erdös and Rényi. To understand the lemma, denote by $G(n,p)$ a random graph with n vertices where an edge exists between two vertices with probability at least p. For a vector $\vec{\alpha} = (\alpha_1, \ldots, \alpha_r)$ where $\alpha_i \geq 0$ for $i = 1, \ldots, r$ let $G(\vec{\alpha}, n, p)$ be an r-partite graph with r sets of vertices $V_1, \ldots, V_r$ where $|V_i| = \alpha_i \cdot n$ for $i = 1, \ldots, r$, and a directed edge between $v \in V_i$ and $v' \in V_{i+1}$ for $i = 1, \ldots, r-1$, or between $v \in V_r$ and $v' \in V_1$, exists with probability at least p. A *perfect allocation* in a graph $G(n,p)$ matches all the vertices; a perfect allocation in $G(\vec{\alpha}, n, p)$ (consisting of cycles of length r) matches all the vertices in the smallest vertex set V_i for $i = \text{argmin}_j |V_j|$.

Deviating from [19], define $G'(\vec{\alpha}, n, p)$ similarly to $G(\vec{\alpha}, n, p)$, except that there are no edges between V_r and V_1. An allocation in $G'(\vec{\alpha}, n, p)$ consists of chains of length r that originate in a vertex in V_1. As before, a perfect allocation in $G'(\vec{\alpha}, n, p)$ matches all the vertices in the smallest vertex set V_i for $i = \text{argmin}_j |V_j|$.

Lemma 1 (Ashlagi & Roth [19])**.** *Let $p > 0$. Then $G(n,p)$ admits a perfect allocation that uses cycles of length at most 3 with high probability. In addition, for any vector $\vec{\alpha}$ as above, the random graphs $G(\vec{\alpha}, n, p)$ and $G'(\vec{\alpha}, n, p)$ admit a perfect allocation with high probability.*

Using Lemma 1, we can assume that if we single out several large groups of vertices (in a large random compatibility graph) that correspond to blood type compatible pairs, there will be sufficiently many edges to admit a perfect matching. For example, if there are large sets of AB-O pairs, O-A pairs, and A-AB pairs, then with high probability we can find an allocation that consists of 3-cycles that matches all the vertices in the smallest set. Even if we consider several such allocations sequentially, by applying the union bound we can see that they all exist with high probability. This essentially allows us to assume in the proof of the next lemma that any two vertices that are blood type-compatible are connected by an edge.

Lemma 2. *Let $G(n)$ be a random graph that admits the following allocation:*

1. *Every self-demanded pair is matched in 2-way or 3-way cycles with other self-demanded pairs.*
2. *Every B-A pair is matched in a 2-way cycle with an A-B pair.*
3. *Every A-B pair that is not matched to a B-A pair is matched in a 3-way cycle with an O-A pair and an A-AB pair.*
4. *For $X \in \{A, B\}$, every over-demanded pair X-O is matched in a 2-way cycle with an O-X pair.*

Then with high probability $G(n)$ admits an efficient allocation that uses cycles of length at most 3 and chains of length at most 3.

Proof. We complete the allocation described in the lemma's statement to an efficient allocation. Figure 2.3 visualizes the augmented allocation; regular edges are assumed by the lemma's formulation while dashed edges are added during this proof. Let V^1 be the set of vertices not matched by the initial allocation. First, as many A-donors as possible donate to A-AB pairs and as many B-donors as possible donate to B-AB pairs (shown in Figure 2.3 by dashed edges from A-altruists to A-AB pairs and from B-altruists to B-AB pairs). In both cases, one of the two vertex sets will be exhausted. More formally, using Lemma 1 we find a perfect allocation for the subgraph induced

by V_A^1 and $V_{\text{A-AB}}^1$, and similarly we find a perfect allocation for the subgraph induced by V_B^1 and $V_{\text{B-AB}}^1$.

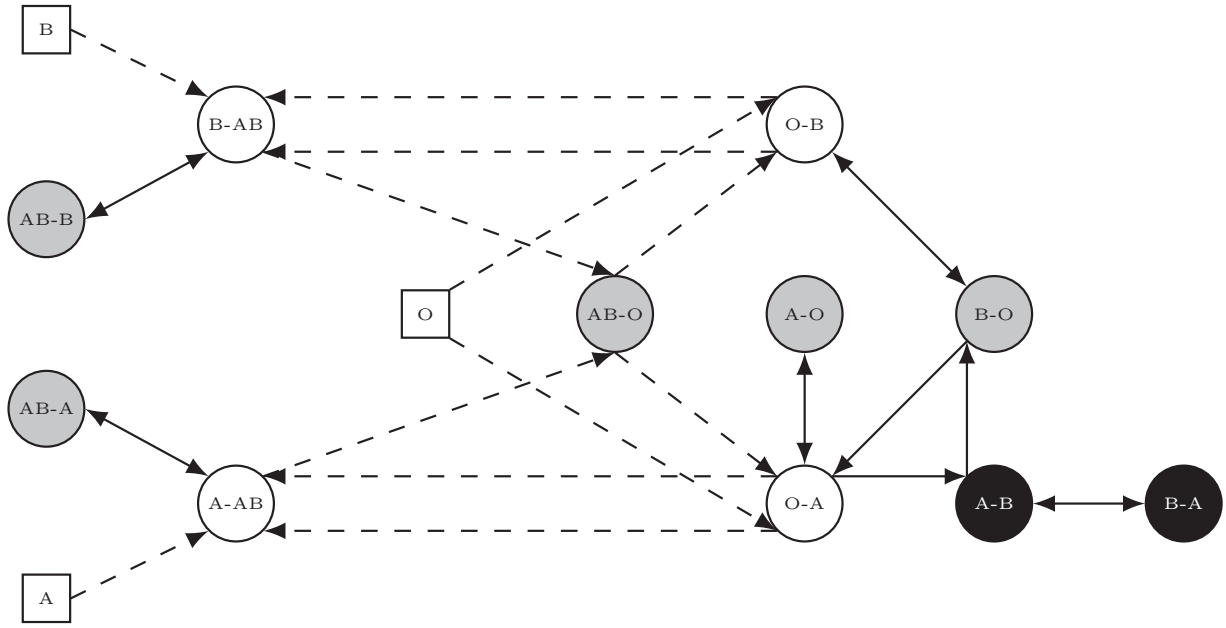


Figure 2.3: Accompanying figure to Lemma 2. Altruists are shown as rectangles; candidate-donor pairs as ovals. Over-demanded pairs are gray, under-demanded are white, and reciprocal pairs are black. Regular edges appear in the lemma's formulation and dashed edges are constructed in the proof.

Let V^2 be the vertices not matched by previous allocations. We first find as many 3-way $(\text{ABO}, \text{O-A}, \text{A-AB})$ cycles as possible, that is, we find a perfect allocation for the subgraph induced by V_{ABO}^2, $V_{\text{O-A}}^2$, and $V_{\text{A-AB}}^2$. It may be the case that $V_{\text{A-AB}}^2 = \emptyset$. Let V^3 be the set of vertices not matched by previous allocations. Next we find a perfect allocation with 3-way $(\text{ABO}, \text{O-B}, \text{B-AB})$ cycles. It may be the case that $V_{\text{ABO}}^3 = \emptyset$ or $V_{\text{B-AB}}^3 = \emptyset$.

Let V^4 be the vertices not matched by previous allocations. The next component in the constructed allocation matches as many O-donors as possible in chains of length 3 of the form $(\text{O}, \text{O-A}, \text{A-AB})$ and then $(\text{O}, \text{O-B}, \text{B-AB})$. This is done sequentially as above. Finally, we match the remaining O-donors and AB-O pairs with remaining under-demanded pairs via chains of length 2 or 2-way cycles (not shown in Figure 2.3).

Each of the allocations constructed above exists with high probability; thus (by applying the union bound) they all exist with high probability. To complete the proof, we argue that our construction gives rise to an efficient allocation. Since under our construction all over-demanded, self-demanded, and reciprocally demanded pairs are matched, it is sufficient to show that no allocation can match more under-demanded pairs.

Following Ashlagi and Roth [19], when vertex v participates in an exchange with under-demanded vertex v' we say that v *helps* v'. Self-demanded and reciprocally demanded pairs cannot

help under-demanded pairs without involving donors or over-demanded pairs. Similarly, AB-donors cannot help under-demanded pairs. In addition, only two types of vertices can help two under-demanded pairs: AB-O pairs can participate in cycles with one of O-A and O-B and one of A-AB and B-AB, and O-donors can start a chain with the same types. Any other vertex can help at most one under-demanded pair, and in particular over-demanded pairs of type X-$Y \neq \text{ABO}$ can only help under-demanded vertices of type Y-X.

Now, A-donors can only help A-AB pairs, and B-donors can only help B-AB pairs. Therefore, it is optimal to match these donors with their respective under-demanded pairs. Finally, in our constructed allocation as many AB-O pairs and O-donors as possible are helping two under-demanded pairs each, while the rest are helping one under-demanded pair each. □

The following lemma directly follows from Proposition 5.2 of [19], and holds under the assumptions of Theorem 1.

Lemma 3 (Ashlagi & Roth [19])**.** *$G(n)$ has an allocation as in Lemma 2, up to symmetries between A-B pairs and B-A pairs, with high probability.*

2.2.3 Discussion

Theorem 1 follows from the proofs of the three lemmas in Section 2.2.2. The theorem itself is motivated by the recent work of Ashlagi and Roth [19]. One has to be careful, though, not to use the exact allocation constructed in Proposition 5.2 of their paper as a starting point for the efficient allocation that involves altruistic donors. Indeed, given that $|V_{\text{A-B}}| \geq |V_{\text{B-A}}|$, Ashlagi and Roth match AB-O pairs in cycles $(\text{ABO}, \text{O-A}, \text{A-AB})$. However, because we are essentially making no assumptions regarding $|V_A|$ and $|V_B|$, it may be the (admittedly extreme) case that there are many (say an infinite supply) of A-donors, few B-donors, few O-donors, and a large number of unmatched under-demanded pairs of type O-B and B-AB. In that case we would rather have the A-donors donate to A-AB pairs while creating cycles $(\text{ABO}, \text{O-B}, \text{B-AB})$. Therefore, we must match AB-O pairs only *after* matching altruistic donors.

The presence of (even short) chains allows us to avoid a negative property of the efficient allocation constructed by Ashlagi and Roth [19]: that it never matches O-AB pairs. These are, in a sense, the "most" under-demanded pairs in that their candidates are hardest to match, while their donors are least capable of finding a match. In our allocation, AB-O pairs and O-donors that cannot participate in 3-cycles can donate to O-AB pairs without affecting the size of the matching. More precisely, if there are sufficiently many donors to fully match one of the sets $V_{\text{O-A}}$ and $V_{\text{A-AB}}$, and one of the sets $V_{\text{O-B}}$ and $V_{\text{B-AB}}$, then an efficient allocation can match O-AB pairs.

Independent work by Ashlagi et al. [23] attempts to explain the observed benefit of longer chains by considering a theoretical model with highly sensitized patients—a model we will introduce and extend in Chapters 5 and 9. Specifically, the probability of tissue type compatibility is allowed to decrease with the size of the graph n. Among other results, it is shown that for any k there exists a small enough probability of compatibility such that chains of length $k+1$ are strictly better than chains of length k. However, to even derive such a statement for chains of length 5 versus chains of length 3, the probability must be as small as c/n for some constant c, whereas

intuitively this probability should be a constant that does not depend on n. Hence, despite the elegance of their results, the assumptions underlying their model may be hard to justify.

2.3 Experimental validation

The theoretical results from Theorem 1 are strong in that they limit the utility of chains to those of length 3 or fewer—as the graph grows to infinity. In this section we study the disconnect between that theorem and the real-world results from the early UNOS kidney match runs (shown in Figures 2.1 and 2.2; we provide more recent results with long chains on UNOS graphs in the succeeding chapters).

There are three potential reasons for this disconnect: (1) the theory applies in the large, and the UNOS exchange is not yet large enough for the theory to have taken hold, (2) the model that each blood type compatible edge fails tissue type compatibility independently and with equal probability is a poor model of the (highly sensitized) UNOS pool, and (3) the theory assumes all edges have equal weight, while in the UNOS exchange, edges are weighted.

The discrepancy between the theory and the fielded results cannot be explained solely by the fact that the theory model uses unweighted edges while the real UNOS data has edge weights. If that were the main difference, we would see the curves in Figure 2.1 reach their maxima at a chain cap of 3. This is not the case. So, we see that even if all the weights were binary, long chains would produce a significant benefit in practice. The difference can, in part, be attributed to the highly structured and very small UNOS pool. This is a product of the newness of the UNOS pilot program and other kidney exchange programs; as the exchanges mature, we may expect the compatibility graph's structure to converge to one more similar to our theoretical model.

In reality, the input graph $G(n)$ cannot grow infinitely; specifically, in kidney paired donation, it has been estimated that in steady state the fully fielded nationwide exchange will have around $10{,}000$ pairs at any one time. In this section, we experimentally determine just how large the candidate pool needs to be for the chain cap prescribed by Theorem 1 to apply—under the assumption that a steady-state pool would also be dense.

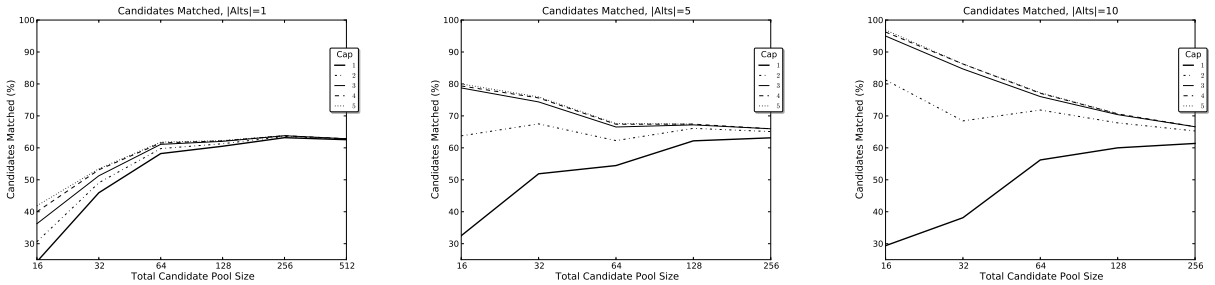
The minimum size of this compatibility graph needed for the theory to take hold depends on the probability distribution of blood and HLA types in the candidate and altruist pools, the number of candidates in the graph, and the number of altruists. We will vary both the number of candidates and altruists, but choose to focus only on blood and HLA types representative of the US population (which serves the current nationwide kidney exchange).

Here we generate candidate-donor pairs and altruists via the most commonly used dense data generator for steady-state kidney exchange, by Saidman et al. [191]. This generator incorporates the blood types from the ABO model discussed earlier. It also incorporates an abstract model of tissue types to compute a type of score that quantifies the likelihood of a specific candidate being tissue type compatible with a random donor. In other words, this tissue type model is more refined than assuming all blood type compatible edges are tissue type incompatible with equal probability.

2.3.1 Increasing the candidate pool size

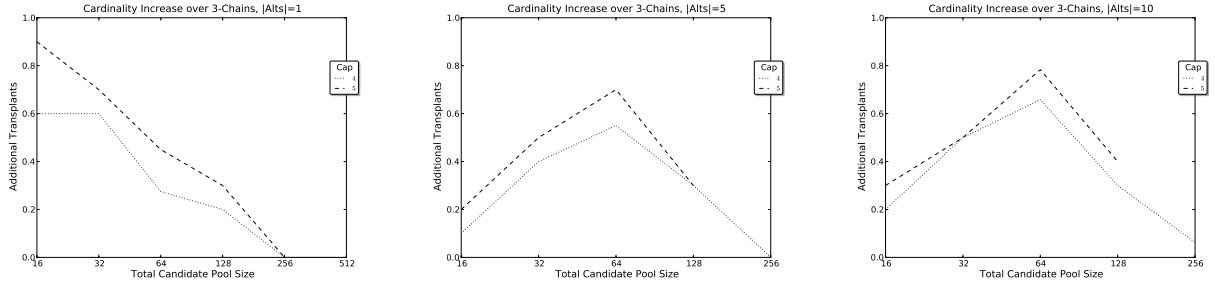
In the first set of experiments, we explore the effect of a large number of *candidates* on the efficacy of long chains. We hold the number of *altruists* constant at 1, 5, or 10 for each experiment.

Figures 2.4a, 2.4b, and 2.4c show that larger pools match a higher percentage of candidates, leveling out at roughly 62% in compatibility graphs with a couple hundred candidates. At a high level, this is a strong argument for a national kidney exchange to replace the set of smaller regional exchanges; see [184] for similar arguments. These figures also make a case for the inclusion of chains in pools at both the regional and national level. Figure 2.4b shows that, for generated pools of size 256, the optimal matching with a chain cap of 1 (i.e., altruists donating directly to the deceased waiting list, avoiding the paired candidate pool entirely) matches nearly 4% fewer candidates overall than matching with a chain cap of 3. The case is more drastic as the number of altruists increases; for instance, Figure 2.4c shows a 5% decrease on compatibility graphs of the same size. The effect of altruists on the pool is discussed further in the next section.



(a) #Altruists=1 **(b)** #Altruists=5 **(c)** #Altruists=10

Figure 2.4: Total percentage of candidates matched as #candidates increases across various chain caps, for various numbers of altruistic donors.



(a) #Altruists=1 **(b)** #Altruists=5 **(c)** #Altruists=10

Figure 2.5: Expected match cardinality increase over 3-chains for 4- and 5-chains, for various numbers of altruistic donors.

From above, we can now ignore matchings that only include chains of length 1 and 2; capping

chains at either of these levels would result in fewer candidates being matched. Figures 2.5a, 2.5b, and 2.5c show the expected number of extra transplants resulting from matches incorporating chains of length 4 and 5, compared to only considering chains of up to length 3. Clearly, the maximum number of additional transplants offered by increasing the chain cap by 1 is proportional to the number of altruists present in the graph. For example, for a graph with a altruists, incorporating 5-chains can provide a benefit of at most $2a$ matches over incorporating at most 3-chains; similarly, increasing the cap from 3 to 4 results in at most a extra matches. Figures 2.5a and 2.5b show that at pool sizes of 256 with $a = 1$ and $a = 5$, the expected number of additional transplants for either 4- or 5-chains is nil (over 100 generated compatibility graphs). Figure 2.5c shows similar results while exemplifying another behavior: as the number of altruists increases, the size of the pool required so that limiting the mechanism to 3-chains is satisfactory increases. This behavior is explored further in the next section.

Figures 2.5b and 2.5c initially show an *increase* in the utility of longer chains as the graph size moves from very small (e.g., 16 candidates) to slightly larger (e.g., 32–64 candidates).[1] This is a side effect of the number of altruists present relative to the size of the pool. With a high enough ratio of altruists to candidates, altruists can "flood" the matching, an idea explored further in the next section.

All of the experiments validate the theory: there seems to clearly be a pool size beyond which long chains do not help.

2.3.2 Increasing the number of altruists

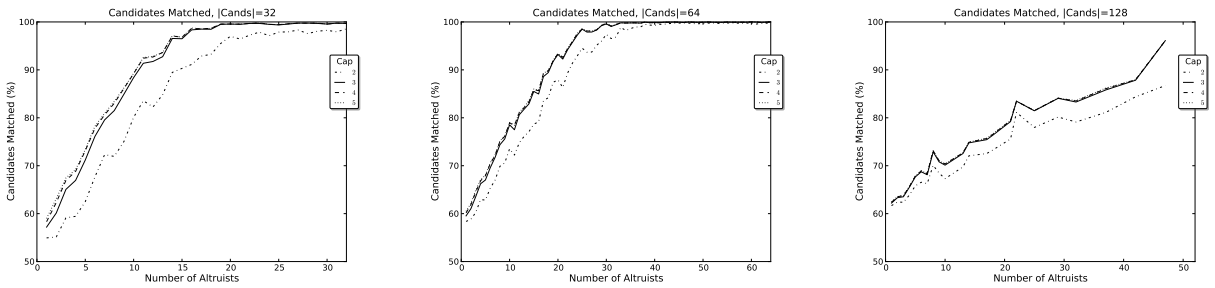
In the previous subsection, we held the number of altruists constant while increasing the size of the candidate pool. We now explore the opposite, allowing ever increasing numbers of altruists to enter candidate pools of constant size.

As the number of altruists increases relative to the size of the candidate pool, the expected number of candidates matched rises to 100%, as shown in Figures 2.6a, 2.6b, and 2.6c. This full flooding of the pool to create a complete matching, while interesting, is not presently a realistic scenario; all three tested compatibility graph sizes would require around 50% as many altruists as candidates in the pool (Figure 2.6c has the x-axis cut short). In our experience with UNOS, the number of altruists is typically around 5% the size of the candidate pool—but, as we discuss in the later chapters that cover dynamic kidney exchange, this can be a function of match cadence and policy. Increasing this number could feasibly change as the exchange grows in size and publicity, paying special notice to the ethical issues that arise in coercion of possible donors [219].

2.4 Dynamic kidney exchange

In the chapter thus far, we have studied static models. We now discuss the dynamics of a kidney exchange running month to month; that cadence has since increased at the UNOS exchange and at

[1] In Figure 2.5c, the computational demands of this experiment precluded us from extending the dotted line past 128 candidates—a clear demonstration of why more scalable algorithms like those presented in Chapters 3 and 4 are needed, as present-day kidney exchanges are larger than the simulated pools in these experiments!



(a) #Candidates=32 **(b)** #Candidates=64 **(c)** #Candidates=128

Figure 2.6: Total percentage of candidates matched as #altruists increases across various chain caps, for various numbers of candidate-donor pairs.

many other exchanges, which we discuss more in Chapters 6 and 8. These early dynamic experiments are included to whet the reader's appetite, to introduce some of the additional complexities handled by the work in the rest of this thesis, and for posterity's sake in that they set early policy at the UNOS exchange and informed future research on both chains and dynamic matching in general.

2.4.1 Augmenting the model

We augment the model in several ways to make it capture the nuances that have arisen in practice.

Dynamics. Most of the work in kidney exchange has focused on a single-shot optimization on a static pool. This deviates from reality in that matching should occur *dynamically*. In reality, candidates arrive and depart from the pool. Even with dialysis, only 12% of patients survive 10 years [215]; this gives us the monthly death rate we use in our experiments. Timeliness in matching is clearly important. Our experimental results, discussed later, perform matching over 24 months using a changing kidney pool.

We provide an in-depth review of related work in dynamic matching and dynamic kidney exchange in Chapter 6—before going on to discuss fully dynamic clearing—but overview one discussion from the medical community that was ongoing at the time of this work: chain execution policies. Work by Gentry et al. [97] on simulated data and Ashlagi et al. [22] on real-world data explores the trade-offs between two types of chain execution polices. The first chain type is executed in its entirety in one time period, with the leftover donor donating to the waiting list. An alternative is to split long chains into segments with intra-segment simultaneous transplants, but the segments execute one after another. The left over donor (aka *bridge donor*) from one segment then serves as a virtual altruist for the next segment. These two types of chains perform differently under the presence of *renege rates*—that is, when a bridge donor decides to leave the pool before donating a kidney. However, no reliable quantification of a renege rate exists due to the infancy of kidney exchanges.

While Gentry et al. [97] do not explicitly consider chain caps, Ashlagi et al. [22] do; they experimentally show that longer (up to length 6) chains can, in fact, help. Our work uses a similar model with single-shot execution chains and, importantly, takes into account the policies of the UNOS nationwide kidney exchange (as they were at the time—that policy has since changed due to this work and other work that we discuss later in this thesis). As we will show, this addition results in different matching behavior. We now discuss these UNOS-specific additions to the model.

Individual crossmatch sensitivity. As exemplified in the real, highly-sensitized UNOS candidate pool, candidates can have widely varying susceptibility to incompatibilities in kidney donation. The dense model due to Saidman et al. [191] from the previous section has a rather realistic view of virtual crossmatch failures, and we use that model here.

In addition, here we do (non-virtual) crossmatches for all the planned transplants just before the transplant takes place, as in reality. This is again done using the Saidman et al. [191] generator. It provides for each candidate a probability that the candidate is tissue type compatible with a random person. We use that probability to draw crossmatch success versus failure. If the crossmatch fails, the transplant cannot proceed. If it is part of a cycle, the cycle does not execute; the pairs in the cycle go back in the pool. The failed edge is permanently removed from the compatibility graph $G(n)$.

Crossmatching has a significant effect on the size of the "real" matching—a topic we explore extensively in Parts II and III, first in isolation in Chapter 5, and then as part of a holistic matching framework in Chapter 8. Briefly, assume an optimal matching (pre-crossmatch) yields a 3-cycle. If *any* crossmatch fails between a candidate and potential donor, the *entire* cycle must be thrown away—since we cannot force a donor to give a kidney if his accompany candidate does not receive one. Even more drastic is the case of chains: if, for example, a pre-crossmatch matching yields a 20-chain, no transplants after the first crossmatch failure can be performed.

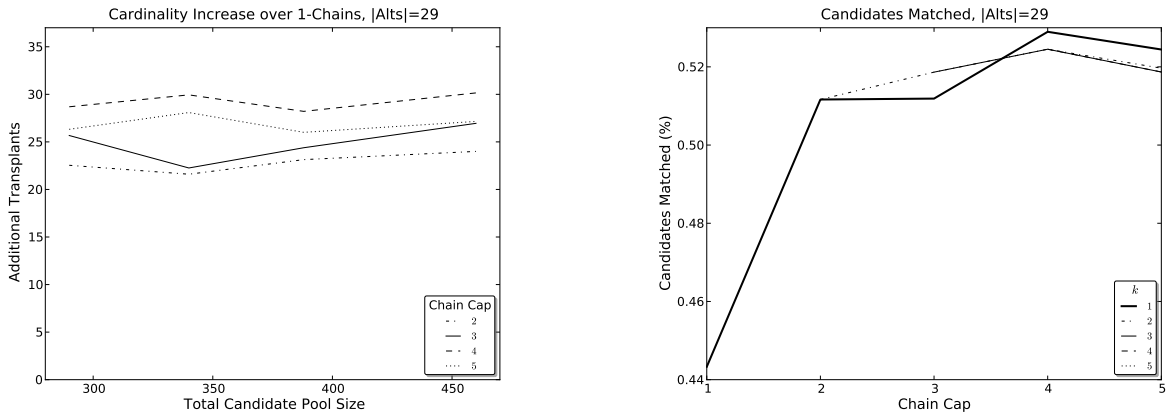
Because of this special case for chains, real-world exchanges have enacted policies for the acceptance or rejection of chains based on their length and the quality of the altruistic donor. O-type altruists are highly valued, as they can (potentially) donate to any blood type, so short chains enabled by O-type altruists should (potentially) be rejected in favor of longer chains in the future. Our experiments follow current UNOS policy which, along with some special cases discussed below, states that (i) chains started by non-O-type altruists are always executed, while (ii) chains triggered by an O-type altruist are executed only if they can be executed to length at least 5 (before there is a crossmatch failure). We will experiment with varying the value away from 5; we will call this parameter k.

Altruists are allowed choices. In the event that an O-type chain is shorter than length 5, the UNOS policy allows for the altruist to decide that the chain be executed anyway. This is due to the fact that altruists do not want to stay in the candidate pool indefinitely, but rather want to move on with their lives and other plans. In UNOS's experience running kidney exchange, altruists typically do not wish to stay active in the pool for more than three months—instead opting to donate directly to the deceased donor waiting list. While exact data on this phenomenon are too sparse at the moment, our experiments use the anecdotal rates (received through UNOS): 75% probability of an altruist requesting execution of a short chain, and a monthly altruist exit rate that corresponds to

an expected presence of two months in the pool for each altruist. Our model executes each chain in a single time segment.

2.4.2 Experimental results

We now present preliminary results simulating dynamic kidney exchange under the model described above. Figure 2.7a shows the expected increase in transplants when including chains over the cycles-only approach. The x-axis describes the total number of candidates available during at least one time period over the entire simulation; between 15 and 20 candidates arrive every time period and between 1 and 2 altruists arrive every time period. The initial pool (i.e., the pool at time $t = 0$) is seeded with between 50 and 100 candidates and 5 altruists. These settings roughly mimicked the state of the UNOS pilot program at the time; it has since grown substantially.



(a) Expected improvement of n-chains over 1-chains (over 24 months), for increasing pool arrival rates.

(b) Expected percentage of candidates matched with 1- to 5-chains, varying k in the UNOS chain execution policy.

Figure 2.7: Dynamic kidney exchange experiments using a myopic, deterministic clearing engine.

The results both remain true and (appear to) deviate from the theory in a number of ways. The benefit of using chains is immediately obvious; in all cases, even using only 2-chains increases the total number of transplants by 20 or more. However, in this new setting, chains of length at most 3 (at least for the tested pool sizes, number of altruists, etc) do *not* provide equivalent benefit to longer chains. While 3-chains do provide a net gain over 2-chains, considering longer chains helps—sometimes by nearly 10 additional transplants. This increase is surprising because, intuitively, longer chains are less likely to be executed in full (and thus likely to be canceled by the UNOS policy) due to low crossmatch probability. Not executing a chain is dangerous because altruists leave the pool entirely if they remain unmatched for more than a few months.

The results above can be explained by considering the effect of time on an evolving small-scale pool of candidates. Over time, highly sensitized candidates will build up in the pool, since they are often significantly harder to match—both because they have fewer connected edges in

the generated compatibility graph and because they are more likely to fail during the crossmatch. Through the real-world results detailed in Section 2.1, we have seen that the utility of (long) chains increases tremendously in the presence of a small, highly sensitized pool. In Figure 2.7a, chains of length greater than 3 are able to serve highly sensitized candidates because they do not need to "close" the chain, as is the case with a cycle.

Surprisingly, allowing the optimizer to use chains of up to length 5 is strictly worse than constraining it to chains of length at most 4 (while a cap of 4 is better than 3). This suggests that there is diminishing benefit to longer and longer chains, and at the same time there is increasing risk of crossmatch failure (and thereby altruists leaving and candidates dying) with increasing chain cap. The experiments here suggest that in the dynamic setting with these pool sizes (i.e., not in the very large), a chain cap of 4 is best. In follow-up work discussed in Chapters 5, 6, and 8, we formally explore how to more precisely strike this tradeoff from a theoretical and experimental point of view; this chapter's experiments reflect the fielded practice at the time.

We now expand our preliminary experiments to include the chain execution policy from UNOS (see Section 2.4.1), and we will vary k (between 1 and the chain cap). Intuitively, a higher k will prevent "wasting" a valuable O-altruist on short chains, favoring waiting for a longer, higher-scoring chain instead. Figure 2.7b shows the effect of varying k as we increase the chain length cap. When considering only short chains, a higher k increases the total number of transplants. In contrast, when chains of length 4 and 5 are considered, it appears better to reduce k. The drop in overall utility from allowing *only* long chains to execute is due to altruists' propensity to leave the pool; if an altruist is not used in an executed chain within a few time period, he/she is likely to leave the pool (and thus be "wasted" by going straight to the deceased donor waiting list instead of saving some lives in the pool first). Following the appearance of our initial paper presenting this work [74], UNOS removed the rule that chains triggered by an O-type altruist are executed only if they can be executed to length at least 5. Our experimental results were the reason for this change in policy.

2.5 Conclusions & present-day implications

In this chapter, we considered altruist-initiated chains, a recent innovation in barter exchanges that has seen wide adoption in regional and national kidney exchange. We showed that a real kidney exchange can benefit, in a static, deterministic model of the world, from long chains that are executed simultaneously. We then showed that, in the large, the benefit from chains longer than 3 becomes negligible (with high probability) on random compatibility graphs drawn from distributions that would mimic a steady-state kidney exchange drawing from the complete real-world population. We supported these theoretical results by experiments using a state-of-the-art dense instance generator due to Saidman et al. [191] to allow us to experiment on larger instances than exist in current kidney exchanges. The theoretical results take hold in exchanges orders of magnitude smaller than the expected steady-state of the nationwide kidney exchange *assuming* the pool mimics a uniform draw from the United States population.

Finally, we experimented in the dynamic setting where the exchange clears every month. Computational complexity precluded experiments in the large for the dynamic setting—which was fine

for this initial work, because at the time kidney exchange pools were small—but in medium-sized pools a chain cap of 4 was best (and strictly better than 5). At any given point in our largest dynamic simulations, 100–150 candidates were present in the pool—others had already been matched, had died, or had not entered the simulation yet. We showed in Section 2.3 that, at such a small size and with so many altruists, we cannot expect 3-chains to suffice.

Thoughts on this work, five years down the line. We performed this research within the first year of the initial fielding of the UNOS kidney exchange, and thus many of the assumptions made did not turn out to be true as that exchange—and, indeed, the rest of the kidney exchange world—evolved. In many ways, this work introduced more questions than it answered, thus motivating further research in both the theory and practice of kidney exchange. How do we create theoretical models that actually mimic fielded exchanges? How do we take short-term uncertainty (e.g., that due to crossmatch failure) into account? Can we quantify its effect on efficiency, and mitigate that effect? Will more efficient clearing engines harm specific classes of agents (e.g., highly-sensitized patients) participating in the exchange, and can we quantify and mitigate that? Should we match with an eye toward the future, or just maximize our utility function now? If so, how can we do this scalably? How do we balance all of this in a way humans can understand? We aim to answer many of these questions in this thesis.

First, though, we note that even the basic static, deterministic experimental results in this chapter suffered from scalability issues due to chains; indeed, fielded kidney exchanges (including UNOS) hit scalability problems as early as 2011, also due to the use of long chains in pools of increasing size. We address this in the succeeding chapter by introducing a method for clearing exchanges with short cycles and long chains that is dramatically more scalable than any prior approach.

A compact formulation of a MIP may have a weak LP relaxation. Frequently, the relaxation can be tightened by a reformulation that involves a huge number of variables.

– Cynthia Barnhart *et al.*

3

Optimal batch clearing of deterministic barter exchanges

3.1 Preliminaries & prior approaches to optimal clearing of large barter exchanges

In this section, we briefly overview the two leading high-level approaches to solving integer program (IP) models of the kidney exchange clearing problem. For a more in-depth coverage of prior integer programming approaches to the kidney exchange problem, we redirect the reader to the recent survey by Mak-Hau [151]. Following these preliminaries, in Section 3.2 we present a bug in one state-of-the-art approach to solving the clearing problem, and show that a core facet of that solver—solving the "pricing problem"—is theoretically harder to perform than previously assumed. Following this, Section 3.3 presents a novel approach to optimally clearing exchanges that is both correct and dramatically faster than prior approaches.

The first optimization-based approach to clearing kidney exchanges was described in the seminal work of Roth et al. [186]; here, only 2-cycles were considered—that is, a pair could only swap with a single other pair, disallowing longer cyclic swaps and any chains. In this case, the problem reduces to classical matching; Roth et al. [186] used Edmonds' algorithm [85] to clear such exchanges in polynomial time. Segev et al. [192] also used Edmonds' algorithm to investigate the effect on match size of a national versus set of regional exchanges under the 2-cycles-only paradigm. In follow-up work considering both 2- and 3-cycles, Roth et al. [188] give a basic IP formulation—which we will refer to later as the "edge formulation"—that could optimally solve very small exchanges without long chains, but was not scalable.

The two fundamental IP models for kidney exchange are the *cycle formulation*, which includes one binary decision variable for each feasible cycle or chain, and the aforementioned *edge formula-*

tion, which includes one decision variable for each compatible pair of agents [6, 188]. In the cycle formulation, the number of constraints is sublinear in the input size, but the number of variables is exponential. In the basic edge formulation, the number of variables is linear but the number of constraints is exponential. Optimally solving these models has been an ongoing challenge for the past decade. We overview both approaches here.

3.1.1 Constraint generation & the edge formulation

Constraint-generation-based approaches to kidney exchange have all variables of the appropriate model in memory from the start, but bring in the constraints of the model incrementally. A basic constraint generation form of the kidney exchange problem uses a decision variable for each edge (i.e., only $O(|V|^2)$ variables) in the compatibility graph and solves a flow problem such that unit flow into a vertex exists if and only if unit flow out of that vertex also exists [6, 188]. This relaxed form of the full problem with only a polynomial number of constraints will not obey cycle or chain caps, so constraints of that form are added until an optimal solution to the relaxed problem is also feasible with respect to cycle and chain caps.

Anderson et al. [16] built the leading constraint-generation-based IP solver for the kidney exchange problem. Their solver builds on the prize-collecting traveling salesperson problem [28], where the problem is to visit each city (patient-donor pair) exactly once, but with the additional option to pay some penalty to skip a city. They maintain decision variables for all cycles of length at most L, but build chains in the final solution from decision variables associated with individual edges. Then, an exponential number of constraints is required to prevent the solver from including chains of length greater than K; these are generated incrementally until optimality is proved. This algorithm is particularly effective for solving instances where the cycle cap is 3 and there is no cap on the length of chains, but it is outperformed by branch-and-price-based approaches if a finite cap on chains is used, as is typically done in practice. We will compare our new solvers against this algorithm, and others, in Section 3.3.

Constantino et al. [63] introduced the first two compact IP formulations for kidney exchange, where *compact* means that the counts of variables and constraints are polynomial in the size of the input. Their *extended edge formulation* was shown empirically to be effective in finding the optimal solution where the cycle cap is greater than 3, particularly on dense graphs. However, each of the compact formulations introduced in their work has a weaker linear program (LP) relaxation than the cycle formulation, discussed below, even in the absence of altruistic donors.

The EE-MTZ model due to Mak-Hau [151], another compact formulation, uses the variables and constraints of the extended edge formulation to model cycles and a variant of the Miller-Tucker-Zemlin model [160] for the traveling salesperson problem to model chains. The same paper introduces the exponentially-sized SPLIT-MTZ model, which adds redundant constraints to the edge formulation in order to tighten the LP relaxation (LPR).

3.1.2 Branch and price & the cycle formulation

Given a set of vertices $V = P \cup A$, the number of cycles of length at most L is $O(|P|^L)$, the number of uncapped chains is exponential in $|P|$ if $A \neq \emptyset$, and the number of capped chains of length (here,

defined to be the number of constituent edges) at most K is $O(|A||P|^K)$. Let $\mathcal{C}(L, K)$ represent the set of cycles of length at most L and chains of length at most K. With one decision variable per cycle and chain $c \in \mathcal{C}(L, K)$, it is not clear that an integer program *model* cannot even be *written* to main memory—much less solved—for even moderately-sized graphs. Indeed, Abraham et al. [6] could not write down the full model for instances as small as 1000 patient-donor pairs for $\mathcal{C}(3, 0)$, while Dickerson et al. [74]—the initial work discussed in Chapter 2—could not write down the full model for instances as small as 256 pairs with just 10 altruists for $\mathcal{C}(3, 4)$. Thus, any solver must maintain at most a reduced model (i.e., subset of columns and rows in the constraint matrix) in memory.

Branch and price is a combination of standard branch and bound with column generation that searches for and proves the optimality of a solution to an IP while maintaining only a reduced model in memory [30]. For kidney exchange, the idea is as follows [6]. (We will loosely use "cycles" to refer to both cycles and chains, except when explicitly distinguished. This is consistent because both are represented as decision variables in the model, and because a chain is equivalent to a cycle with an additional "dummy" zero-weight back-edge to an altruistic donor.) First, start with some relatively small number of, or no, "seed" cycle variables in the model, and solve the linear program (LP) relaxation of this reduced model. Next, generate *positive price* cycles—variables that might improve the solution when brought into the model. For the maximum-weight clearing problem, the price of a cycle c is given by $\sum_{(u,v)\in c}(w_{(u,v)} - \delta_u)$, where δ_u is the dual value of vertex u in the LP.

The pricing problem is to generate one or more positive price cycles to bring into the model, or prove that none exist. While any positive price cycles exist at the current node in the branch and bound search tree, optimality has not been proven for the LP. Solving the pricing problem can be expensive in its own right, as we discuss in Sections 3.2 and 3.3, and later in a more general model in Chapter 5. Once there are no more positive price cycles, if the LP solution is integral, optimality is proved at that node in the search tree. However, if the LP is fractional, branching occurs. Abraham et al. [6] branched on individual cycles c, creating one subtree that includes c in the final solution and a second subtree that explicitly does not, and recursing in this way. (Our solvers in Section 3.3 will necessarily uses more complex branching, when necessary.) These branches are then explored in depth-first order until a provably optimal solution is found.

A number of kidney exchange algorithms use the cycle formulation with branch and price [6, 75, 100, 134, 174]. These have been some of the fastest algorithms to date for the kidney exchange problem and are deployed in countries like the United States and the Netherlands; we will build on them in the succeeding chapters.

3.2 Pricing cycles and chains in deterministic kidney exchange

In this section, we discuss solving the *pricing problem* used in branch-and-price-based approaches to clearing large kidney exchanges. The pricing problem is to determine whether there exists a positive price (i.e., promising) column. Once no more positive price columns exist, optimality has been proven for that node in the branch-and-bound search tree, and the search can proceed further in the tree. Quickly solving the pricing problem at nodes in the search tree is important

for overall runtime. Recently, it was shown that determining whether a positive price *cycle* exists can be solved in polynomial time [100, 174]. Both Glorie et al. [100] and Plaut et al. [174] also use a variant of their cycle-pricing algorithms for chains. In this section, we show that not only are those latter algorithms incorrect, but the underlying problem—determining whether a positive price *chain* exists—is, in fact, NP-complete.

Related Publications

Initial versions of some of the work in this section appeared at AAAI-16, and an early version of the rest is available on arXiv; it is an ongoing collaboration between Plaut, Dickerson, and Sandholm [174, 175].

3.2.1 The pricing problem in kidney exchange

We begin with the standard model of static, deterministic kidney exchange, which is modeled as a directed *compatibility graph* $G = (V, E)$. In this section and the next, we explicitly partition the set of all vertices V into P and A, where vertices in P represent patient-donor pairs and vertices in A represent altruistic donors (aka "non-directed donors"). Under this partitioning, for each $u, v \in P$, the edge (u, v) exists if the donor of pair u is compatible with the patient of pair v. Similarly, for each $a \in A$ and $v \in P$, the edge (a, v) exists if altruist a is compatible with the patient of pair v. These edges may also have weights, representing the relative value of a potential transplant.

In this section and the next, formally, we define a chain as a path beginning at an altruist, and the length of a cycle or chain to be the number of *edges* it contains. The weight of a cycle or chain is then the sum of its constituent edge weights.

As discussed informally in Section 3.1.2, the basic cycle formulation of the kidney exchange problem is defined as follows. Let $\mathcal{C}(L, K)$ be the set of all cycles of length at most L and chains of length at most K. Then, given a binary indicator variable $x_c \in \{0, 1\}$ for each $c \in \mathcal{C}(L, K)$, we must solve the following integer linear program:

$$\max \sum_{c \in \mathcal{C}(L,K)} w_c \, x_c \qquad s.t. \qquad \sum_{c : v \in c} x_c \leq 1 \quad \forall v \in V$$

We now formally define the pricing problem in the context of kidney exchange. The pricing problem is to determine whether there exists a positive price cycle or chain. The price of a cycle or chain c is $\sum_{(u,v) \in c} w_{(u,v)} - \sum_{v \in c} \delta_v$, where $w_{(u,v)}$ is the weight of edge (u, v), and δ_v is the dual value of vertex v in the linear program (LP) relaxation. The initial branch-and-price-based formulation due to Abraham et al. [6] solved the pricing problem by explicitly considering every possible cycle and chain via an exhaustive depth-first search, taking time exponential in $\max(L, K)$. This became a problem in practice as K increased to allow for longer chains. Recently, Glorie et al. [100] showed how determining whether a positive price cycle in the compatibility graph $G = (V, E)$ exists is equivalent to finding a negative weight cycle in a *reduced graph* $G' = (V, E')$, where each edge $e' = (u, v) \in E'$ exists if and only if $(u, v) \in E$, and e' has *reduced weight* $r_{(u,v)} = \delta_v - w_{(u,v)}$.

A similar equivalence holds for chains. We must be careful, however, since the number of vertices in a chain exceeds the number of edges by 1. We now define $r_{(u,v)}$ as follows:

$$r_{(u,v)} = \begin{cases} \delta_v - w_{(u,v)} & u \in P \\ \delta_u + \delta_v - w_{(u,v)} & u \in A \end{cases}$$

Since an outgoing edge from an altruist will only ever be used in a chain, this ensures that a chain has positive price in G if and only if it has negative weight in the reduced graph G'.

3.2.2 Counterexample to two prior algorithms

In this section, we provide counterexamples to the pricing algorithms of both Glorie et al. [100] and the proposed fix to that algorithm by Plaut et al. [174]. Both previous algorithms use Bellman-Ford-style search in the reduced graph, initiated from each altruist as the source, to find negative-weight chains. Ideally, we would like to find the *shortest* paths using each vertex at most once, but this is NP-hard in the presence of negative cycles. As discussed by Plaut et al. [174], this is via reduction from the Hamiltonian cycle problem: set all edge weights to -1 and ask if the shortest path from a source u to any neighbor v such that $(v, u) \in E$ is of weight $1 - |V|$. However, we need not find the shortest paths beginning at each altruist: we only need to determine whether there exists any negative path starting at *any* altruist.

In the presence of negative cycles, traditional Bellman-Ford may generate paths with internal loops, which are invalid in our context. Plaut et al. [174] handle this by preventing Bellman-Ford from looping during execution. As a result, the generated paths may not be the shortest, and a given negative chain may not be found.

The full pseudocode for the method of Plaut et al. [174] is given as Algorithm 1, and is an adaptation of the polynomial pricing algorithm provided by Glorie et al. [100]. In Algorithm 1, for a fixed source, let $d_i(v)$ represent the computed distance from that source to v after the ith step of the algorithm, where $d_0(v)$ represents the distances before any steps are performed. Distance is defined as the sum of the edge weights in the computed path. Let L and K be the maximum allowable cycle and chain lengths, respectively. Finally, let A be the set of altruist donors and let P be the set of donor-patient pairs. The function GETNEGATIVECYCLES is called with the reduced graph $G = (V, E)$, cycle cap L, and chain cap K.

For the version of Algorithm 1 for cycles, Plaut et al. [174] show that although there may be negative cycles that are not found, at least one negative cycle will be found, if any exist. The proof of the version of the algorithm for chains (i.e., Algorithm 1 as written here) is incorrect in general, however, as it implicitly assumes that the chain length cap and cycle length cap are equal.

Plaut et al. [174] gave a counterexample to the algorithm of Glorie et al. [100]. Figure 3.1 gives a counterexample to the algorithm of Plaut et al. [174]; this is also a counterexample to the original algorithm due to Glorie et al. [100].

For cycle length cap $L = 3$ and chain length cap $K = 5$, there are no valid negative cycles in the reduced graph, and there is a single valid negative chain in the graph: $(a, p_5, p_2, p_3, p_4, p_1)$. Although (p_1, p_2, p_3, p_4) is a cycle with negative weight, it exceeds the cycle length cap of $L = 3$, and thus is invalid.

Algorithm 1 (Incorrect) pricing of cycles and chains via an adapted Bellman-Ford search.

function GETNEGATIVECYCLES($G = (V, E), L, K$)
  $\mathcal{C} \leftarrow \emptyset$ ▷ Accumulator set for negative weight cycles
  for each $s \in V$ **do**
    $N \leftarrow s \in A \; ? \; K - 1 : L - 1$ ▷ Set maximum step number based on chain or cycle cap
    $pred_0(v) = \emptyset \;\; \forall v \in V$
    $d_0(s) = 0$ ▷ Distance from source to source is zero
    $d_0(v) = \infty \;\; \forall v \neq s \in V$ ▷ Distance at step 0 to other vertices is infinite
    for $i \in \{1, \ldots, N\}$ **do**
      $d_i(v) = d_{i-1}(v) \;\; \forall v \neq s \in V$
      $pred_i(v) = pred_{i-1}(v) \;\; \forall v \neq s \in V$
      for each $(u, v) \in E$ **do**
        if $v \notin$ TRAVERSEPREDS$(u, pred, i - 1)$ **then** ▷ Avoid loops in path
          if $d_{i-1}(u) + w(u, v) < d_i(v)$ **then** ▷ If this step decreases the distance to node
            $d_i(v) \leftarrow d_{i-1}(u) + w(u, v)$ ▷ Update to shorter distance
            $pred_i(v) \leftarrow (u, i - 1)$ ▷ Store correct predecessor
    for each $v \neq s \in V$ **do** ▷ Find negative weight cycles with s as the source
      if $d_N(v) + w(v, s) < 0$ **then**
        $\mathcal{C} \leftarrow \mathcal{C} \cup$ TRAVERSEPREDS$(v, pred, N)$
  return $\mathcal{C}$
function TRAVERSEPREDS($v, pred, n$)
  $c \leftarrow []$ ▷ Start with an empty list (representing a cycle or chain)
  $curr \leftarrow v$
  while $curr \neq \emptyset$ **do** ▷ Until we reach the source node ...
    $c \leftarrow curr + c$ ▷ Add predecessor to path
    $(u, i) \leftarrow pred_n(curr)$ ▷ Get predecessor of predecessor
    $curr \leftarrow u; \;\; n \leftarrow i$
  return c

In the second iteration of the algorithm due to Plaut et al. [174], vertex p_2 would store as its most promising predecessor the path (a, p_1, p_2) with weight $w\left[(a, p_1, p_2)\right] = 0$, instead of (a, p_5, p_2) with less promising weight $w\left[(a, p_5, p_2)\right] = 1$. However, this causes the algorithm to miss the overall negative chain that would be found otherwise at iteration 5, since it cannot reuse vertex p_1 (and thus cannot use the sole negative-weight edge with sink p_1). Critically, even though the path (a, p_5, p_2) was not promising at an earlier iteration, following it instead of the more immediately promising (a, p_1, p_2) would have led to a negative-weight chain—in this case, the only negative-weight chain. The initial algorithm due to Glorie et al. [100] would also incorrectly return that no negative-weight chains exist, by similar reasoning.

This shows a correctness error in both the algorithms of Glorie et al. [100] and the proposed fix due to Plaut et al. [174]. Next, we show that, in general, such polynomial-time approaches are

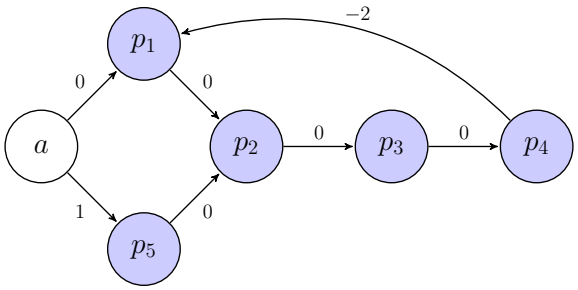


Figure 3.1: Example where the algorithm of Plaut et al. [174] fails to find a negative chain for $L = 3$ and $K = 5$, although one exists. Here, vertex a is an altruist ($a \in A$) and the rest of the vertices are incompatible donor-patient pairs ($\{p_1, \ldots, p_5\} \in P$).

hopeless: determining whether a positive price chain exists is NP-complete.

3.2.3 Main result: this form of pricing is not possible in polynomial time

We define the *negative chain problem* as follows: given a directed graph $G = (V, E)$, where $V = P \cup A$, is there a path through P (using each vertex at most once) of negative weight, using at most K edges, and starting at some vertex $a \in A$? We call such a path a negative chain.

Theorem 2. *The negative chain problem is NP-complete.*

Proof. The negative chain problem is trivially in NP: simply sum the edge weights in a proposed path and check its sign. To show NP-hardness, we reduce from the directed Hamiltonian path problem. Given some graph $H = (V, E)$, the directed Hamiltonian path problem asks whether there exists a directed path that visits each vertex exactly once. Let $n = |V|$ and $V = \{v_1, \ldots, v_n\}$. Construct the graph G as follows: set $w_e = -1$ for each $e \in E$, and add a vertex a with an edge (a, v) with $w_{(a,v)} = n - 2$ for each $v \in V$. Figure 3.2 gives an example of the construction of the graph G. Let $P = V$, $A = \{a\}$, and $K = n$.

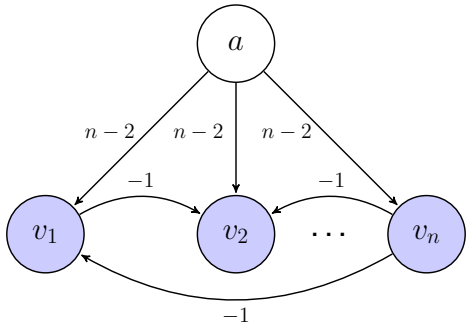


Figure 3.2: Example construction for the proof of Theorem 2.

Suppose h is a Hamiltonian path in H starting at v_i. Let $c = (a, v_i) \cup h$. Since h has exactly $n-1$ edges, c contains n edges, thereby satisfying the length constraint. Since h visits each $v \in V$

exactly once and never visits a, c visits each vertex in G at most once. Finally, since h has weight $1-n$, c has weight $n-2+1-n=-1$. Therefore c is a negative chain in G.

Suppose c is a negative chain in G. Then c must begin at a, so we can write $c=(a,v_i)\cup h$, for some $v_i \in V$ and path h. Let m be the number of edges in h. Then $w_c = n-2-m$. Since $w_c < 0$, we have $m > n-2$. Since c can use each vertex at most once, we have $m \leq n-1$. Therefore $m = n-1$. Because c visits each vertex at most once, h visits each vertex at most once. Since h has $n-1$ edges, h visits every vertex in V exactly once, making it a valid Hamiltonian path in H. □

The general pricing problem (where both cycles and chains are included) is to determine whether there exists a positive price (negative weight) cycle of length at most L or a positive price (negative weight) chain of length at most K. Note that solving the general pricing problem does not necessarily solve the negative chain problem. If X is the set of negative chains and Y is the set of negative cycles, the general pricing problem is to determine whether $X \cup Y = \emptyset$. The negative chain problem is to determine whether $X = \emptyset$: however, determining whether $X \cup Y = \emptyset$ does not necessarily determine whether $X = \emptyset$.

To show that the general pricing problem is NP-hard, we modify the above construction by expanding each edge in H to a series of $\max(L,1)$ edges whose weights sum to -1. Then any cycle in G has length at least $2L$, which violates the length constraint for $L \geq 2$. For $L < 2$, there are no valid negative cycles regardless. Since there are no valid negative cycles in G, the general pricing problem becomes equivalent to the negative chain problem. Finally, we set $K = n \cdot \max(L,1)$ to ensure that any chain satisfying the length cap in the original construction remains valid. Therefore, the general pricing problem is also NP-hard. Since the general pricing problem is also trivially in NP, it is NP-complete.

3.2.4 Hardness in the branch and price context

The negative chain problem in general is NP-complete. However, it could be that the instances which arise in the branch and price context necessarily have a certain structure, and form a subclass which is not NP-complete. We now show that this is not the case.

Theorem 3. *The negative chain problem, restricted to instances which occur in a kidney exchange branch and price search tree, is NP-complete.*

Proof. The problem remains trivially in NP. For NP-hardness, we will show that the hardness proof of Theorem 2 can be achieved using only instances that occur as reduced graphs in branch and price search trees for kidney exchange instances.

For an arbitrary graph $H = (V,E)$, where $V = \{v_1 ... v_n\}$, construct G as in the proof of Theorem 2. Construct the graph G' as follows: starting from G, add vertices p_1 and p_2, and add edges (a,p_1) and (a,p_2), each with weight 0. Set $P = V \cup \{p_1,p_2\}$, $A = \{a\}$, and $K = n$. This adds exactly two chains to G, both with weight 0. Thus the set of negative chains remains unchanged. Therefore using G' instead of G in the proof of Theorem 2 preserves correctness.

Construct the kidney exchange instance (not a reduced graph) G_0 as follows: use the same vertices and edges as G', set $w_{(a,p_1)} = w_{(a,p_2)} = n-1$, and set all other edge weights to 1.

Figure 3.3 gives an example of the construction of the graph G_0. We will now show that G_0 yields G' as a reduced graph.

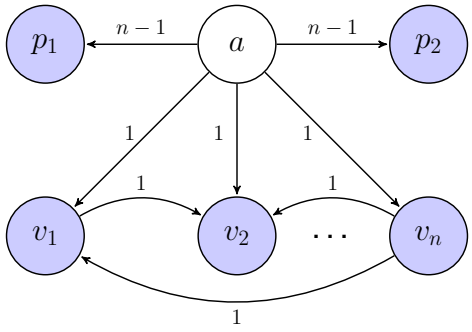


Figure 3.3: An instance of a kidney exchange compatibility graph that could arise in practice and that yields a reduced graph equivalent to that used in the proof of Theorem 2.

The weights in the reduced graph depend on the dual values in the LP. The dual value of a constraint is equal to the increase in the objective value of the LP optimum if that constraint were relaxed by one unit, while all other constraints remain unchanged. In our case, each constraint corresponds to a particular vertex, and ensures that that vertex is used at most once. The dual value of vertex v refers to the dual value of the constraint corresponding to vertex v. Therefore the dual value of a vertex v is the potential increase in the LP objective value if v were allowed to be used at most twice, instead of at most once.

At the root of the search tree, if the only variables in the initially-seeded reduced model are the two chains (a, p_1) and (a, p_2), any optimal solution to the LP is either $\{(a, p_1)\}$, $\{(a, p_2)\}$, or a convex combination of the two. The LP objective value is $n - 1$. If vertex a could be used twice instead of once, both (a, p_1) and (a, p_2) could be fully included in the solution, increasing the objective value by $n - 1$ to $2n - 2$. Therefore, the dual value of vertex a is $n - 1$. Since none of vertices $\{v_1, \ldots, v_n\}$ are used by any variable in the model, allowing them to be used more than once would have no effect on the objective value of the LP. Therefore, $\delta_{v_i} = 0$ for $1 \leq i \leq n$. Similarly, p_1 and p_2 each appear in a single chain, so allowing them to be used multiple times would yield no benefit. Therefore $\delta_{p_1} = \delta_{p_2} = 0$.

Recall that the weight of an edge (u, v) in the reduced graph is $r_{(u,v)} = \delta_v - w_{(u,v)}$ if $u \in P$, and $r_{(u,v)} = \delta_u + \delta_v - w_{(u,v)}$ if $u \in A$. Therefore $r_{(a,p_1)} = \delta_a + \delta_{p_1} - w_{(a,p_1)} = n - 1 + 0 - (n - 1) = 0$. By symmetry, $r_{(a,p_2)} = 0$. For each $e = (v_i, v_j) \in E$, $r_e = \delta_{v_j} - w_e = 0 - 1 = -1$. Finally, $r_{(a,v_i)} = \delta_a + \delta_{v_i} - w_{(a,v_i)} = n - 1 + 0 - 1 = n - 2$. Therefore G_0 yields G' as a reduced graph.

Therefore, the negative chain problem, even restricted to instances which occur in a kidney exchange branch and price search tree, is NP-complete. □

The proof of Theorem 3 relies on weighted edges, so one may wonder if this result extends to the unweighted case, where every edge in a kidney exchange instance has weight 1. The kidney exchange instance G_0 can be made unweighted by expanding each edge e with weight $w_e > 1$—in this case, just (a, p_1) and (a, p_2)—to w_e edges, each with weight 1. This is possible because all edge weights in G_0 are integers. An example of this construction is given in Figure 3.4.

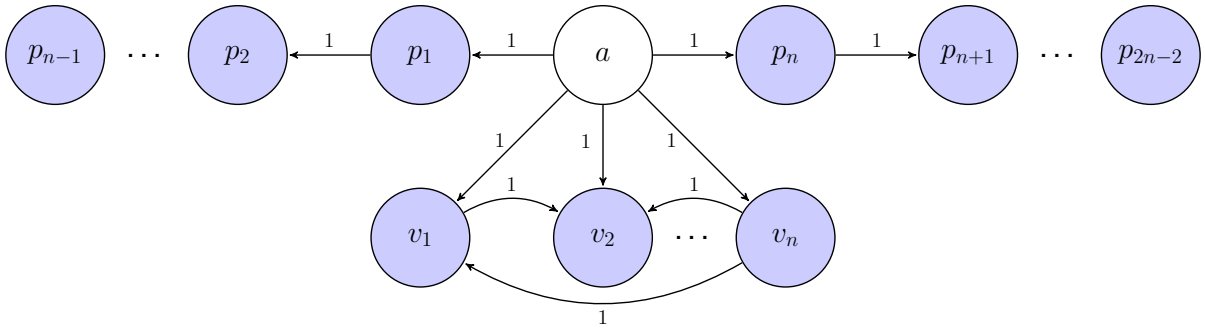


Figure 3.4: Translation of the weighted compatibility graph of Figure 3.3 to an unweighted version; the proof of Theorem 3 then applies to this case almost directly.

Allowing vertex a to be used twice instead of once would still increase the LP objective value by $n-1$, keeping the dual value of vertex a unchanged. Thus, the set of negative chains remains unchanged, so this modification preserves the correctness of the proof of Theorem 3. Therefore the result of Theorem 3 holds even if only unweighted kidney exchange instances are considered.

3.2.5 Implications

We discussed branch-and-price-based approaches to the kidney exchange problem, and showed that solving the pricing problem for chains, and thereby the pricing problem for cycles and chains jointly, is NP-complete. This shows a correctness error in two leading branch-and-price-based solvers. The results apply to other barter exchanges as well, as long as they use chains (potentially with cycles as well).

Our hardness results show that a different approach for handling chains is necessary. Next, we introduce models where chains are represented by position-indexed edge variables. Since there are only a polynomial number of edge variables, they can be fully enumerated, removing the need for branch and price for chains. Cycles in these models can still be handled via branch and price, or via a different scheme.

3.3 Position-indexed formulations for kidney exchange

This section presents new scalable integer-programming-based approaches to optimally clearing large kidney exchanges, including two models which can comfortably handle chain caps greater than 10 for pools of sizes substantially larger than those in, e.g., Chapter 2. We focus specifically on the realistic setting of small cycle caps L and large—but finite—chain caps K. We introduce three integer program formulations for the kidney exchange problem, two of which are compact. Model size (i.e., memory footprint) often constrains today's kidney exchange solvers; critically, our models are typically much smaller than the prior state of the art *while managing to maintain tight linear program relaxations (LPRs)*—which in practice is quite important to proving optimality quickly.

In Section 3.3.1, we introduce the *position-indexed edge formulation (PIEF)*, a model for the kidney exchange problem with only cycles that is substantially smaller than, yet has an LPR equivalent to, the model with the tightest LPR for the cycles-only version of the problem [6, 188]. Section 3.3.1 presents the *position-indexed chain-edge formulation (PICEF)* which compactly brings chains into the model via a polynomial number of decision variables; the number of cycle decision variables is exponential in just the maximum *cycle* length (which is typically only 3 or 4 in fielded exchanges). To address that latter exponential reliance on the cycle length, we also present a branch-and-price-based implementation of PICEF. Finally, in Section 3.3.3 we present the *hybrid position-indexed edge formulation (HPIEF)*, which combines PIEF and PICEF to yield a compact formulation.

Throughout, we prove new results regarding the tightness of the LPRs of our models relative to the current state of the art. The tightness of these relaxations hints that our formulations will be competitive in practice; toward that end, we provide extensive experimental evidence that they are. In particular, we show that at least one of PICEF and HPIEF is faster than the best solver from all those provably-optimal solvers contributed in earlier papers that we evaluated in 96.41% of instances considered, with the speed-ups being most evident for larger instance sizes and larger chain caps. In Section 3.3.5, we use real and generated data from two nationwide kidney exchange programs—one in the UK, and one in the US—to compare our formulations against other competitive solvers [6, 16, 134, 174]. Our new formulations are on par or faster than all other solvers, outperforming all other solvers by orders of magnitude on many problem instances.

Finally, while we focus on the maximum-cardinality and maximum-weight cycle and chain cover problems here, we note that our models can be extended to work with alternative objects. One alternative objective for the kidney exchange problem is maximizing the *expected* number of transplants subject to post-match edge and vertex failures [12, 75, 172]; we describe how our model can be extended to this setting at the end of this chapter. Some fielded exchanges like those in the UK and the Netherlands use lexicographic optimization of a hierarchy of objectives [100, 152]; we note that our models would work under simple augmentation in these settings as well.

Related Publications

Some of the work in this section appeared at EC-16, and the rest can be found on arXiv; it is a collaboration between Dickerson, Manlove, Plaut, Sandholm, and Trimble [80, 81].

We especially would like to thank Ross Anderson, Kristiaan Glorie, Xenia Klimentova, Nicolau Santos, and Ana Viana for valuable discussions regarding this work and for making available their kidney exchange software for the purposes of conducting our experimental evaluation in Section 3.3.4.

3.3.1 PIEF: Position-Indexed Edge Formulation

We begin by presenting the first of our three new IP formulations, the *position-indexed edge formulation* (PIEF). PIEF is a natural extension of the extended edge formulation (EEF) of Constantino et al. [63]. For this formulation, we assume that the problem instance contains no altruistic donors;

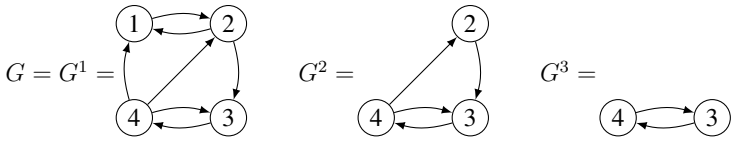


Figure 3.5: A kidney exchange instance G with $|A| = 0$ and $|P| = 4$, along with graph copies G^1 ($= G$), G^2, and G^3. $G^4 = (\{4\}, \{\})$ is not shown.

HPIEF (the hybrid PIEF) in Section 3.3.3 is a compact generalization of this formulation which can be used for instances with altruistic donors.

The PIEF, like the EEF, uses copies of the underlying compatibility digraph $G = (V, E)$. For each vertex $l \in V$, let $G^l = (V^l, E^l)$ be the subgraph of G induced by $\{i \in V : i \geq l\}$. The PIEF ensures that at most one cycle is selected in each copy, and that a cycle selected in graph copy G^l must contain vertex l.

The first directed graph in Figure 3.5 is an instance with four patients which we will use as an example in this section. The figure shows graph copies G^1 ($= G$), G^2, and G^3. The remaining graph copy, $G^4 = (\{4\}, \{\})$, contains no edges and is not shown.

The main innovation of the PIEF formulation is the use of edge *positions* to index variables; the position of an edge in a cycle is defined as follows. Let $c = (e_1, \ldots, e_{|c|})$ be a cycle represented as a list of edges in E. Further, assume that we use the unique representation of c such that e_1 leaves the lowest-numbered vertex involved in the cycle. For $1 \leq i \leq |c|$, we say that e_i has position i.

We define $\mathcal{K}(i, j, l)$, the set of positions at which edge (i, j) is permitted to be selected in a cycle in graph copy G^l. For $i, j, l \in V$ such that $(i, j) \in E^l$, let

$$\mathcal{K}(i,j,l) = \begin{cases} \{1\} & i = l \\ \{2, \ldots, L-1\} & i, j > l \\ \{2, \ldots, L\} & j = l. \end{cases}$$

Thus, an edge may be selected at position 1 in graph copy l if and only if it leaves vertex l, and any edge selected at position L in graph copy l must enter l.

Now, create a set of binary decision variables as follows. For $i, j, l \in P$ such that $(i, j) \in E^l$, create variable x^l_{ijk} for each $k \in \mathcal{K}(i, j, l)$. Variable x^l_{ijk} takes the value 1 if and only if edge (i, j) is selected at position k of a cycle in graph copy G^l. Returning to our example instance and letting $L = 3$, we give x^1_{342} as an example of a variable in the model; this represents the edge $(3, 4)$ being used in position 2 of a cycle in graph copy 1. In full, the set of variables created for this instance is x^1_{121}, x^1_{212}, x^1_{213}, x^1_{232}, x^1_{342}, x^1_{412}, x^1_{413}, x^1_{422}, x^1_{432} (in graph copy G^1), x^2_{231}, x^2_{342}, x^2_{422}, x^2_{423}, x^2_{432} (in graph copy G^2), x^3_{341}, x^3_{432}, and x^3_{433} (in graph copy G^3).

The following integer program finds the optimal cycle packing.

$$\max \quad \sum_{l \in V} \sum_{(i,j) \in E^l} \sum_{k \in \mathcal{K}(i,j,l)} w_{ij} x^l_{ijk} \tag{3.1a}$$

$$\text{s.t.} \quad \sum_{l \in V} \sum_{j:(j,i) \in E^l} \sum_{k \in \mathcal{K}(j,i,l)} x^l_{jik} \leq 1 \qquad i \in V \tag{3.1b}$$

$$\sum_{\substack{j:(j,i) \in E^l \wedge \\ k \in \mathcal{K}(j,i,l)}} x^l_{jik} = \sum_{\substack{j:(i,j) \in E^l \wedge \\ k+1 \in \mathcal{K}(i,j,l)}} x^l_{i,j,k+1} \qquad \begin{array}{l} l \in V, \\ i \in \{l+1, \ldots, n\}, \\ k \in \{1, \ldots, L-1\} \end{array} \tag{3.1c}$$

$$x^l_{ijk} \in \{0,1\} \qquad l \in V, (i,j) \in E^l, k \in \mathcal{K}(i,j,l) \tag{3.1d}$$

The objective (3.1a) is to maximize the weighted sum of selected edges. Constraint (3.1b) is the capacity constraint for vertices: for each vertex $i \in V$, there must be at most one selected edge whose target is i. Constraint (3.1c) is the flow conservation constraint. For each graph copy G^l, each vertex i in G^l except the lowest-numbered vertex, and each edge position $k < L$, the number of selected edges at position k with target i is equal to the number of selected edges at position $k+1$ with source i. Constraint (3.1d) ensures that no fractional solutions are selected. Theorem 17 in Section A.1.1 establishes the correctness of the PIEF model.

We note that PIEF is not the first IP model for a directed-graph program to use position-indexed variables; Vajda [216] uses position-indexed variables for subtour elimination in a model for the traveling salesman problem (TSP). Vajda's model is substantially different from the PIEF; most notably, graph copies are not required for Vajda's model because any TSP solution contains exactly one cycle.

Further reducing the size of the basic PIEF model

We now present methods for reducing the size of the PIEF model while maintaining provable optimality. These reductions are performed as a polynomial-time preprocess (prior to solving the NP-hard kidney exchange clearing problem), and thus may result in practical run time improvements.

Basic reduced PIEF. In typical problem instances, there are many (i,j,k,l) tuples such that x^l_{ijk} takes the value zero in any assignment that satisfies constraints (3.1b)-(3.1d). For example, suppose that $L = 4$ and that Figure 3.6 is graph copy G^1. Edge $(6,7)$ cannot be chosen at position 3 of a cycle, since the edge only appears in one cycle and it is at position 2 of that cycle. Hence, if we could eliminate variable x^1_{673} from the integer program it would not change the optimal solution. Similarly, all variables for the edge $(3,4)$ within this graph copy could be eliminated, since this edge does not appear in any cycle of length less than 5.

Following the approach used by Constantino et al. [63] for the extended edge formulation, we eliminate variables as follows. For $i, j \in V^l$, let d^l_{ij} be the length of the shortest path in terms of

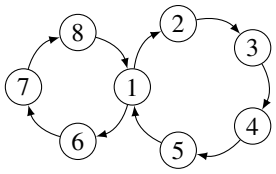


Figure 3.6: A graph copy where the reduced PIEF decreases the number of variables in the integer program

edges from i to j in G^l. For $(i,j) \in E^l$, let

$$\mathcal{K}^{\text{red}}(i,j,l) = \{k : 1 \le k \le L \wedge d^l_{li} < k \wedge d^l_{jl} \le (L-k)\}.$$

For any $k \notin \mathcal{K}^{\text{red}}(i,j,l)$, no cycle in graph copy G^l of length less than or equal to L contains (i,j) at position k, since either there is no sufficiently short path from l to i or there is no sufficiently short path from j to l. Note that $\mathcal{K}^{\text{red}}(i,j,l) \subseteq \mathcal{K}(i,j,l)$. We can substitute $\mathcal{K}^{\text{red}}(i,j,l)$ for $\mathcal{K}(i,j,l)$ in (3.1a)-(3.1d), yielding a smaller integer program—*PIEF-reduced (PIEFR)*—with the same optimal solution.

Elimination of variables at position 1 and L: the PIEFR^2 formulation. In the PIEFR model with $L \ge 3$, variables at position 1 are redundant, since $x^l_{lj1} = 1$ if and only if $x^l_{ji2} = 1$ for some i. Similarly, variables at position L are redundant; $x^l_{jlK} = 1$ if and only if $x^l_{ij(K-1)} = 1$ for some i. We can eliminate variables at positions 1 and L from PIEFR as follows. Define a modified weight function w': for all $i,j,l \in P$ such that $(i,j) \in E^l$ and all $k \in \{2,\ldots,L-1\}$, let

$$w'(i,j,k,l) = \begin{cases} w_{ij} + w_{li} & k = 2 \\ w_{ij} + w_{jl} & k = L-1 \\ w_{ij} & \text{otherwise.} \end{cases}$$

With this weight function, a selected edge (i,j) at position 2 of a cycle in E^l contributes to the objective value its own weight plus the weight of the implicitly selected edge (l,i). An edge (i,j) at position $L-1$ of a cycle in E^l contributes its own weight plus the weight of of the implicitly selected edge (j,l).

For $i,j,l \in P$ such that $(i,j) \in E^l$, define the restricted set of permitted edge positions:

$$\mathcal{K}^{\text{red}^2}(i,j,l) = \mathcal{K}^{\text{red}}(i,j,l) \setminus \{1,L\}.$$

For $i,j,l \in P$ such that $(i,j) \in E^l$, and for each $k \in \mathcal{K}^{\text{red}^2}(i,j,l)$, create a binary variable x^l_{ijk}.

The following IP, denoted PIEFR^2 (PIEF reduced twice), is solved.

$$\max \quad \sum_{l \in V} \sum_{(i,j) \in E^l} \sum_{k \in \mathcal{K}^{\text{red}^2}(i,j,l)} w'(i,j,k,l) x^l_{ijk} \quad \text{subject to} \tag{3.2a}$$

$$\sum_{l \in V} \left(\sum_{j:(j,i) \in E^l} \sum_{k \in \mathcal{K}^{\text{red}^2}(j,i,l)} x^l_{jik} + \sum_{\substack{j:(i,j) \in E^l \wedge \\ 2 \in \mathcal{K}^{\text{red}^2}(i,j,l)}} x^l_{ij2} \right) + \sum_{\substack{h,j:j \neq i \wedge \\ (h,j) \in E^i \wedge \\ K-1 \in \mathcal{K}^{\text{red}^2}(h,j,i)}} x^i_{hj(L-1)} \leq 1 \quad i \in V \tag{3.2b}$$

$$\sum_{\substack{j:(j,i) \in E^l \wedge \\ k \in \mathcal{K}^{\text{red}^2}(j,i,l)}} x^l_{jik} = \sum_{\substack{j:(i,j) \in E^l \wedge \\ k+1 \in \mathcal{K}^{\text{red}^2}(i,j,l)}} x^l_{i,j,k+1} \quad \begin{array}{l} l \in V, \\ i \in \{l+1, \ldots, n\}, \\ k \in \{2, \ldots, L-2\} \end{array} \tag{3.2c}$$

$$x^l_{ijk} \in \{0,1\} \quad \begin{array}{l} l \in V, (i,j) \in E^l, \\ k \in \mathcal{K}^{\text{red}^2}(i,j,l) \end{array} \tag{3.2d}$$

The constraints of PIEFR^2 differ from those of PIEFR (3.1b-3.1d) in the following two respects. First, the second term in parentheses in the PIEFR^2 capacity constraint for vertex i (3.2b) ensures that any selected edge (i,j) at position 2 of a cycle in G^l counts towards the capacity for i, since it is implicit that the edge (l,i) is also chosen. The final term on the left-hand side of constraint (3.2b) serves the same function for selected edges at position $L-1$, since an edge at position L is implicitly chosen. Second, the flow conservation constraint (3.2c) is not required for $k \in \{1, L-1\}$, since edges at positions 1 and L are not modeled explicitly in PIEFR^2.

Vertex-ordering heuristic. We can reduce the number of variables in the reduced PIEF model by carefully choosing the order of vertex labels in the digraph G. We have found that relabelling the vertices in descending order of total degree is an effective heuristic to this end. To estimate the effect of this ordering heuristic on model size, we generated the PIEFR^2 model for each of the ten PrefLib instances [154] with 256 vertices and no altruistic donors—these were generated in accordance with the dense model of Saidman et al. [191]. The heuristic reduced the variable count by a mean of 38 percent, and reduced the constraint count by a mean of 60 percent.

PIEF has a tight LPR

We now compare the LPR bound of PIEF to those of other popular IP models. The tightness of an LPR is typically viewed as a proxy for how well an IP model will perform in practice, due to the important role the relaxation plays in modern branch-and-bound-based tree search. In this section, we compare the LPR of PIEF against the IP formulation with the tightest LPR, the *cycle formulation* due to Abraham et al. [6] and Roth et al. [188]; we also note that the formulation is equivalent to that due to Anderson et al. [16] if chains are disallowed. While the number of decision variables in the cycle formulation model is exponential in the cycle cap L, PIEF maintains an LPR that is just as tight, but has far fewer variables if $L > 3$.

LPR of PIEF. We now show that the LPR of PIEF is exactly as tight as that of the cycle formulation. Formally, if A and B are two IP formulations for the kidney exchange problem, we write that A *weakly dominates* B, denoted $Z_A \preceq Z_B$, if for every problem instance, the LPR objective value under A is no greater than the LPR objective value under B. Further, we say that A *strictly dominates* B, denoted $Z_A \prec Z_B$, if $Z_A \preceq Z_B$ and for some problem instance, the LPR objective value under A is strictly smaller than the LPR objective value under B. Finally, we write $Z_A = Z_B$ if $Z_A \preceq Z_B$ and $Z_B \preceq Z_A$. The following result is proved in Appendix A.1.2.

Theorem 4. *$Z_{CF} = Z_{PIEF}$ (without chains).*

3.3.2 PICEF: Position-Indexed Chain-Edge Formulation

Our second new IP formulation, PICEF, uses a variant of PIEF for chains, and—like the cycle formulation—uses one binary variable for each cycle. The idea of using variables for edges in chains and a variable for each cycle was introduced in the PC-TSP-based algorithm of Anderson et al. [16]. The innovation in our IP model is the use of position indices on edge variables, which results in polynomial counts of constraints and edge-variables; this is in contrast to the exponential number of constraints in the PC-TSP-based model.

Unlike PIEF, PICEF does not require copies of G. Intuitively, this is because a chain is a simpler structure than a cycle, with no requirement for a final edge back to the initial vertex.

We define $\mathcal{K}'(i,j)$, the set of possible positions at which edge (i,j) may occur in a chain in the compatibility graph G. For $i,j \in V$ such that $(i,j) \in E$,

$$\mathcal{K}'(i,j) = \begin{cases} \{1\} & i \in A \\ \{2,\ldots,L\} & i \in P \end{cases}.$$

Thus, any edge leaving an altruistic donor can only be in position 1 of a chain, and any edge leaving a patient vertex may be in any position up to the cycle-length cap L, except 1.

For each $(i,j) \in E$ and each $k \in \mathcal{K}'(i,j)$, create variable y_{ijk}, which takes value 1 if and only if edge (i,j) is selected at position k of some chain. For each cycle c in G of length up to L, define a binary variable z_c to indicate whether c is used in a packing.

For example, consider the instance in Figure 3.7, in which $|A| = 2$ and $|P| = 4$. Suppose that $L = 3$ and $K = 4$, and suppose further that each edge has unit weight. The IP model includes variables y_{131}, y_{141}, and y_{241}, corresponding to edges leaving altruistic donors. For each $k \in 2,3,4$, the model includes variables y_{34k}, y_{45k}, y_{56k}, y_{64k}, and y_{65k}, corresponding to edges between donor-patient pairs at position k of a chain. Finally, the model includes z_c variables for the cycles $((4,5),(5,6),(6,4))$ and $((5,6),(6,5))$.

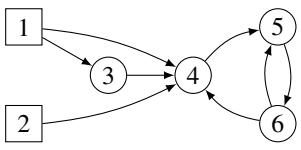


Figure 3.7: An instance with $|A| = 2$ and $|P| = 4$

The following IP is solved to find a maximum-weight packing of cycles and chains.

$$\max \quad \sum_{(i,j)\in E} \sum_{k\in\mathcal{K}'(i,j)} w_{ij} y_{ijk} + \sum_{c\in\mathcal{C}(L,0)} w_c z_c \tag{3.3a}$$

$$\text{s.t.} \quad \sum_{j:(j,i)\in E} \sum_{k\in\mathcal{K}'(j,i)} y_{jik} + \sum_{c\in\mathcal{C}(L,0):i \text{ appears in } c} z_c \leq 1 \qquad i \in P \tag{3.3b}$$

$$\sum_{j:(i,j)\in E} y_{ij1} \leq 1 \qquad i \in A \tag{3.3c}$$

$$\sum_{\substack{j:(j,i)\in E\wedge \\ k\in\mathcal{K}'(j,i)}} y_{jik} \geq \sum_{j:(i,j)\in E} y_{i,j,k+1} \qquad \begin{array}{l} i \in P, \\ k \in \{1,\ldots,L-1\} \end{array} \tag{3.3d}$$

$$y_{ijk} \in \{0,1\} \qquad (i,j) \in E, k \in \mathcal{K}'(i,j) \tag{3.3e}$$

$$z_c \in \{0,1\} \qquad c \in \mathcal{C}(L,0) \tag{3.3f}$$

Inequality (3.3b) is the capacity constraint for patients: each patient vertex is involved in at most one chosen cycle or incoming edge of a chain. Inequality (3.3c) is the capacity constraint for altruists: each altruist vertex is involved in at most one chosen outgoing edge. The flow inequality (3.3d) ensures that patient-donor pair vertex i has an outgoing edge at position $k+1$ of a selected chain only if i has an incoming edge at position k; we use an inequality rather than an equality since the final vertex of a chain will have an incoming edge but no outgoing edge. Theorem 18 in Section A.2.1 establishes the correctness of the PICEF model.

We now give an example of each of the inequalities (3.3b–3.3d) for the instance in Figure 3.7. For $i = 4$, the capacity constraint (3.3b) ensures that $y_{141} + y_{241} + y_{342} + y_{343} + y_{344} + z_{((4,5),(5,6),(6,4))} \leq 1$. For $i = 1$, the altruist's capacity constraint (3.3c) ensures that $y_{131} + y_{141} \leq 1$. For $i = 5$ and $k = 2$, the chain flow constraint (3.3d) ensures that $z_{452} + z_{652} \geq z_{563}$; that is, the outgoing edge $(5,6)$ can only be selected at position 3 of a chain if an incoming edge to vertex 5 is selected at position 2 of a chain.

In our example in Figure 3.7, the optimal objective value is 4. One satisfying assignment that gives this objective value is $y_{131} = y_{342} = z_{((5,6),(6,5))} = 1$, with all other variables equal to zero.

Practical implementation of the PICEF model

We now discuss methods for the practical implementation of PICEF, first by reducing the number of decision variables via a polynomial-time preprocess, and second by tackling the large number of decision variables for cyclic exchanges via a branch-and-price-based transformation of the model.

Reduced PICEF. We can reduce the PICEF model using a similar approach to the PIEF reduction in Section 3.3.1. For $i \in P$, let $d(i)$ be length of the shortest path in terms of edges from some $j \in A$ to i. Since any outgoing edge from i cannot appear at position less than $d(i)+1$ in a chain, we can replace $\mathcal{K}'$ in PICEF with $\mathcal{K}^{\text{red}}$, defined as follows:

$$\mathcal{K}^{\text{red}}(i,j) = \begin{cases} \{1\} & i \in A \\ \{d(i)+1,\ldots,K\} & i \in P. \end{cases}$$

A branch and price implementation of PICEF. We now discuss a method for scaling PICEF to graphs with high cycle caps, or large graphs with many cycles; this method maintains the full set of edge decision variables, but only incrementally considers those corresponding to cycles.

Formally, for $V = P \cup A$, the number of cycles of length at most L is $O(|P|^L)$, making explicit representation and enumeration of all cycles infeasible for large enough instances, as we discussed earlier in this chapter. With one decision variable per cycle, Abraham et al. [6] could not even write the full integer program in memory for instances as small as 1000 pairs, and—once chains were introduced into the picture—our experimental work in Chapter 2 could not even scale to substantially smaller pools.

Section 3.2 discussed how recent work by Glorie et al. [100], and a proposed bugfix to that work by Plaut et al. [174], were both incorrect for the general problem of pricing *both* cycles and chains. However, those algorithms are correct for pricing *only cycles*. In the case of PICEF, we need only price cycles; the number of variables and constraints needed to represent chains is polynomial in the input size and thus that part of the model can be represented in memory completely. Thus, we are able to directly plug in cycle-only pricing algorithms to create a branch and price implementation of PICEF—which we do in our experimental section. Appendix A.4.3 gives the correct version of that cycle pricer as Algorithm 4.

The LPR of PICEF is not as tight

As an analogue to Section 3.3.1, we now compare the LPR of PICEF against the cycle formulation LPR. Unlike in the PIEF case, where Theorem 4 showed an equivalence between the two models' relaxations, we show that PICEF's relaxation can be looser than that of the cycle formulation. Theorem 5 gives a simple construction showing this, while Theorem 6 presents a family of graphs on which PICEF's LPR is arbitrarily worse than that of the cycle formulation. The proofs of both of these results are contained in Section A.2.2.

Theorem 5. *$Z_{CF} \prec Z_{PICEF}$ (with chains).*

Indeed, Theorem 6 shows that the ratio between the optimum objective value for the relaxations of PICEF and the cycle formulation can be made arbitrarily large.

Theorem 6. *Let $z \in \mathbb{R}^+$ be given. There exists a problem instance for which $Z_{PICEF}/Z_{CF} > z$, where Z_{PICEF} is the objective value of the LPR of PICEF and Z_{CF} is the objective value of the LPR of the cycle formulation.*

While the results of Theorems 5 and 6 may be disheartening, in the following section, we give experimental evidence that PICEF (as well as its branch-and-price-based interpretation) perform extremely competitively on real and generated data.

3.3.3 HPIEF: A hybrid formulation

In this section, we present a compact generalization of the PIEF model to kidney exchange graphs with altruistics donors (also called non-directed donors). This stands in contrast to the the PICEF formulation, which has polynomial counts of constraints and edge variables, but an exponential number of cycle variables. By replacing the cycle variables in PICEF with the variables from PIEF and modifying the constraints accordingly, we can create a compact formulation, the *hybrid PIEF (HPIEF)*. Let the variables x^l_{ijl} and the index set $\mathcal{K}(i,j,l)$ be defined as in PIEF. Let the variables y_{ijk} and the index set $\mathcal{K}(i,j)$ be defined as in PICEF. The HPIEF integer program is as follows.

$$\max \quad \sum_{l \in P} \sum_{(i,j) \in E^l} \sum_{k \in \mathcal{K}(i,j,l)} w_{ij} x^l_{ijk} + \sum_{(i,j) \in E} \sum_{k \in \mathcal{K}(i,j)} w_{ij} y_{ijk} \tag{3.4a}$$

$$\text{s.t.} \quad \sum_{l \in P} \sum_{j:(j,i) \in E^l} \sum_{k \in \mathcal{K}(j,i,l)} x^l_{jik} + \sum_{j:(j,i) \in E} \sum_{k \in \mathcal{K}(j,i)} y_{jik} \leq 1 \qquad i \in P \tag{3.4b}$$

Constraints (3.1c), (3.3c), and (3.3d)

$$x^l_{ijk} \in \{0,1\} \qquad l \in P, (i,j) \in E^l, \; k \in \mathcal{K}(i,j,l) \tag{3.4c}$$

$$y_{ijk} \in \{0,1\} \qquad (i,j) \in E, k \in \mathcal{K}(i,j) \tag{3.4d}$$

Inequalities (3.4b) and (3.3c) are the capacity constraints for patients and altruistic donors respectively.

The reductions described in Sections 3.3.1 and 3.3.1 can also be applied to the x^l_{ijk} in HPIEF.

3.3.4 Experimental comparison of state-of-the-art kidney exchange solvers

In this section, we compare implementations of our new models against existing state-of-the-art kidney exchange solvers. To ensure a fair comparison, we received code from the author of each solver that is not introduced in this chapter. We compare run times of the following state-of-the-art solvers:

- BNP-DFS, the original branch-and-price-based cycle formulation solver due to Abraham et al. [6];
- BNP-POLY, a branch-and-price-based cycle formulation solver with pricing due to Glorie et al. [100] and Plaut et al. [174];[1]
- CG-TSP, a recent IP formulation based on a model for the prize-collecting traveling salesman problem, with constraint generation [16];
- PICEF, the model from Section 3.3.2 of this chapter;

[1]Section 3.2 of this chapter showed a correctness bug in both implementations of the BNP-POLY-style solvers due to Glorie et al. [100] and Plaut et al. [174]; for posterity, we still include these run times. Furthermore, we note that the objective values returned by BNP-POLY always equaled that of the other provably-correct solvers on all of our test instances.

- BNP-PICEF, a branch and price version of the PICEF model, as presented in Section 3.3.2 of this chapter;
- HPIEF, the Hybrid PIEF model from Section 3.3.3 of this chapter (which reduces to PIEF for $K = 0$); and
- BNP-DCD, a branch-and-price algorithm using the Disaggregated Cycle Decomposition model, which is related to both the cycle formulation and the extended edge formulation [134].

A cycle-length cap of 3 and a time limit of 3600 seconds was imposed on each run. When a timeout occurred, we counted the run-time as 3600 seconds.

We test on two types of data: real and generated. Section 3.3.4 shows run time results on *real* match runs, including 286 runs from the UNOS US-wide exchange, which (at the time of these runs) contained 143 transplant centers, and 17 runs from the NLDKSS UK-wide exchange, which uses 24 transplant centers. Section 3.3.4 increases the size and varies other traits of the compatibility graphs via a realistic generator seeded by the real UNOS data. We find that PICEF and HPIEF substantially outperform all other models.

Real match runs from the UK- and US-wide exchanges

We now present results on real match run data from two fielded nationwide kidney exchanges: The United Network for Organ Sharing (UNOS) US-wide kidney exchange where the decisions are made by algorithms and software from CMU, and the UK kidney exchange (NLDKSS) where the decisions are made by algorithms and software from Dr. Manlove's group at the University of Glasgow.[2] The UNOS instances used include all the match runs starting from the beginning of the exchange in October 2010 to January 2016. The exchange has grown significantly during that time and chains have been incorporated. The match cadence has increased from once a month to twice a week; that keeps the number of altruists relatively small. We will discuss this more in Chapters 6 and 8. On average, these instances have $|A| = 2$, $|P| = 231$, and $|E| = 5021$. The NLDKSS instances cover the 17 quarterly match runs during the period January 2012–January 2016. On average, these instances have $|A| = 7$, $|P| = 201$, and $|E| = 3272$.

Figure 3.8 shows mean run times across all match runs for both exchanges; Appendix A.5 gives additional statistics like minimum and maximum run times, as well as their standard deviations. Immediately obvious is that the non-compact formulations—BNP-DFS and CG-TSP—tend to scale poorly compared to our newer formulations. Interestingly, BNP-PICEF tends to perform worse than the base PICEF and HPIEF; we hypothesize that this is because branch-and-price-based methods are necessarily more "heavyweight" than standard IP techniques, and the small size of presently-fielded kidney exchange pools may not yet warrant this more advanced technique. Perhaps most critically, both PICEF and HPIEF clear real match runs in both exchanges within seconds.

In the NLDKSS results, the wide fluctuation in mean run time as the chain cap is varied can be

[2]Due to privacy constraints on sharing real healthcare data, the UNOS and NLDKSS experimental runs were necessarily performed on different computers—one in the US and one in the UK. All runs *within* a figure were performed on the same machine, so relative comparisons of solvers within a figure are accurate.

explained by the small sample size of available NLDKSS instances, and the fact that the algorithms other than HPIEF and PICEF occasionally timed out at one hour. By contrast, each of the HPIEF and PICEF runs on NLDKSS instances took less than five seconds to complete. We also note that the LP relaxation of PICEF and HPIEF are very tight in practice; the LPR bound equaled the IP optimum for 614 of the 663 runs carried out on NLDKSS data.

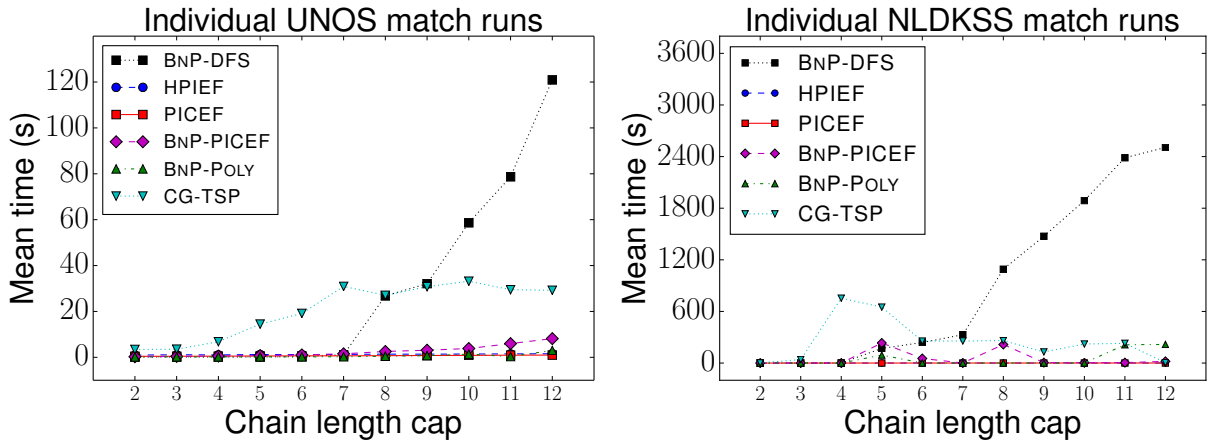


Figure 3.8: Mean run times for various solvers on 286 real match runs from the UNOS exchange (left), and 17 real match runs from the UK NLDKSS exchange (right).

We remark that the BNP-DCD model due to Klimentova et al. [134] was run on all NLDKSS instances where the chain cap K was equal to 0. Larger values of K could not be tested since the current implementation of the model in our possession does not accept altruistic donors (and thus will ignore potential chains) in the input. However for the case that $K = 0$ the BNP-DCD model was the fastest for all NLDKSS instances.

Finally we note that the solver of Glorie et al. [100] was executed on the NLDKSS instances with a chain cap of K, for $0 \leq K \leq 4$. It was found that on average the execution time was 8.9 times slower than the fastest solver from among all the others executed on these instances as detailed at the beginning of Section 3.3.4. PICEF was the fastest solver on 40% of occasions.

Scaling experiments on realistic generated UNOS kidney exchange graphs

As motivated earlier in the paper, it is expected that kidney exchange pools will grow in size as (a) the idea of kidney exchange becomes more commonplace, and barriers to entry continue to drop, as well as (b) organized large-scale international exchanges manifest. Toward that end, in this section, we test solvers on generated compatibility graphs from a realistic simulator seeded by all historical UNOS data; the generator samples patient-donor pairs and altruistic donors with replacement, and draws edges in the compatibility graph in accordance with UNOS' internal edge creation rules.

Figure 3.9 gives results for increasing numbers of patient-donor pairs (each column), as well as increasing numbers of altruist donors as a percentage of the number of patient-donor pairs (each

row). As expected, as the number of patient-donor pairs increases, so too do run times for all solvers. Still, in each of the experiments, for each chain cap, both PICEF and HPIEF are on par or (typically) much faster—sometimes by orders of magnitude compared to other solvers. Appendix A.5 gives these results in tabular form, including other statistics—minimum and maximum run times, as well as their standard deviations—that were not possible to show in Figure 3.9.

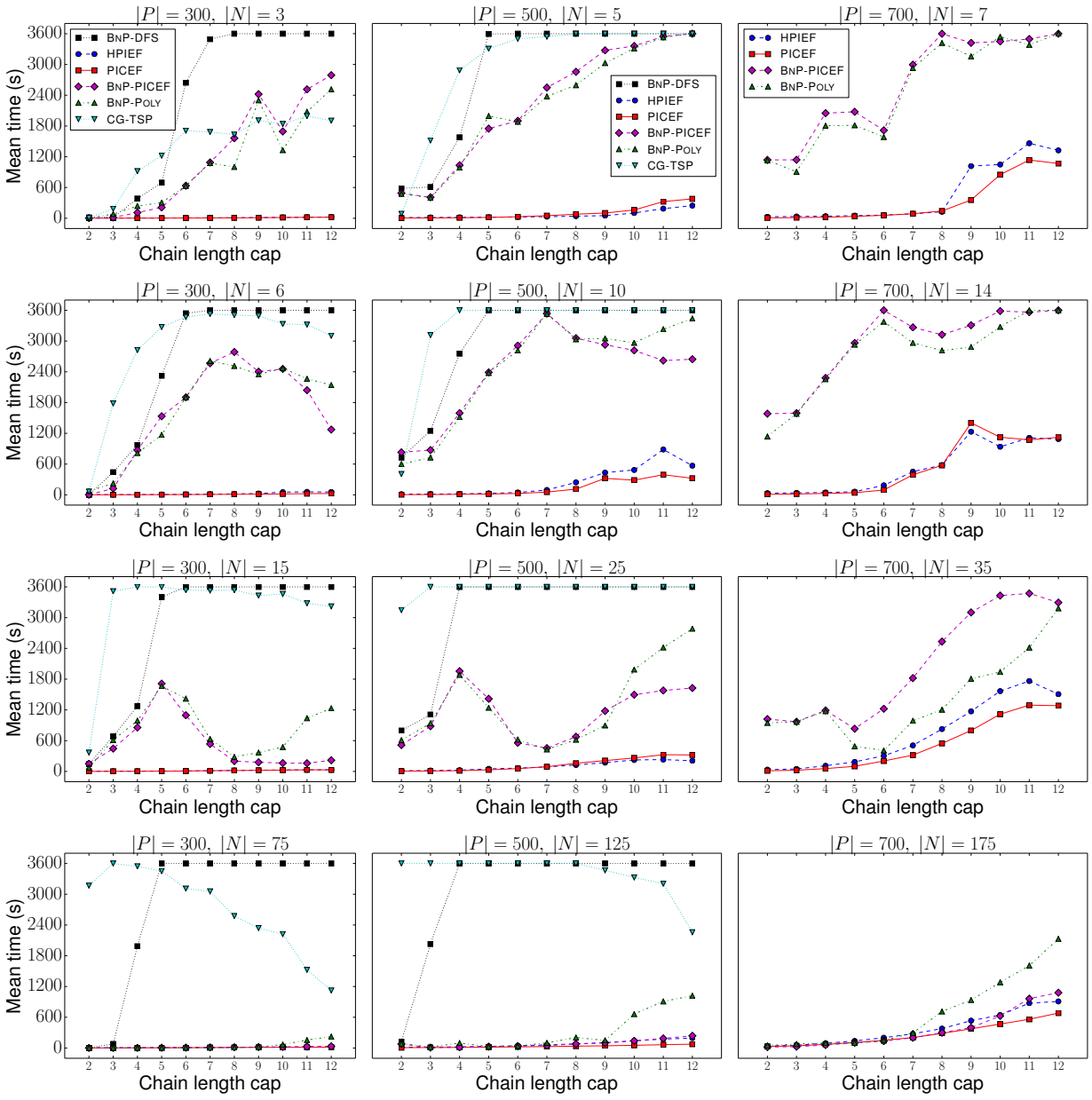


Figure 3.9: Mean run time as the number of patient-donor pairs $|P| \in \{300, 500, 700\}$ increases (left to right), as the percentage of altruists in the pool increases $|A| = \{1\%, 2\%, 5\%, 25\%\}$ of $|P|$ (top to bottom), for varying finite chain caps.

In addition to their increased scalability, we note two additional benefits of the PICEF and HPIEF models proposed in this chapter: reduced variance in run time, and relative ease of implementation when compared to other state-of-the-art solution techniques. In both the real and simulated experimental results, we find that the run time of both the PICEF and HPIEF formulations is substantially less variable than the branch-and-price-based and constraint-generation-based IP solvers. While the underlying problem being solved is NP-hard, and thus will always present worst-case instances that take substantially longer than is typical to solve, the increased predictability of the run time of these models relative to other state-of-the-art solutions—including those that are presently fielded—is attractive. Second, we note that *significant* engineering effort is involved in the creation of custom branch-and-price and constraint-generation-based codes, while both PICEF and HPIEF are implemented with relative ease, relying on only a single call to a black box IP solver.

3.3.5 Failure-aware kidney exchange

Real-world exchanges all suffer to varying degrees from "last-minute" failures, where an algorithmic match or set of matches fails to move to transplantationLeishman et al. [143]. This can occur for a variety of reasons, including more extensive medical testing performed before a surgery, a patient or donor becoming too sick to participate, or a patient receiving an organ from another exchange or from the deceased donor waiting list; we discuss these reasons in greater detail in Chapter 5.

To address these post-match edge failures, our work to be presented in Chapter 5 augments the standard model of kidney exchange to include a success probability p for each edge in the graph. We show how to solve that model using branch and price, where the pricing problem is solved in time *exponential* in the chain and cycle cap. Prior compact formulations—and, indeed, prior "edge formulations" like those due to Abraham et al. [6], Constantino et al. [63], and Anderson et al. [16]—are not expressive enough to allow for generalization to this model. Intuitively, while a single edge failure prevents an entire cycle from executing, chains are capable of incremental execution, yielding utility from the initial altruistic donor to the first edge failure later in the chain. Thus, the expected utility gained from an edge in a chain is dependent on where in the chain that edge is located, which is not expressed in those models.

PICEF for failure-aware matching

With only minor modification, PICEF allows for implementation of failure-aware kidney exchange, under the restriction that each edge is assumed to succeed with equal probability p. While this assumption of equal probabilities is likely not true in practice, there is good reason why a fielded implementation of this model would potentially choose to equalize all failure probabilities: namely, so that already-sick patients—who will likely have higher failure rates—are not further marginalized by this model. (We discuss this marginalization more in Chapter 7.) Thus, given a single success probability p, we can adjust the PICEF objective function to return the maximum expected

weight matching as follows:

$$\max \quad \sum_{(i,j)\in E} \sum_{k\in\mathcal{K}'(i,j)} p^k w_{ij} y_{ijk} + \sum_{c\in\mathcal{C}(L,0)} p^{|c|} w_c z_c \tag{3.5a}$$

Objective (3.5a) is split into two parts: the utility received from edges in chains, and the utility received from cycles. For the latter, a cycle c of size $|c|$ has probability $p^{|c|}$ of executing; otherwise, it yields zero utility. For the former, if an edge is used at position k in a chain, then it yields a p^k fraction of its original weight—that is, the probability that the underlying chain will execute at least through its first k edges.

Failure-aware polynomial pricing for cycles

The failure-aware branch-and-price work we will present in Chapter 5 generalizes the pricing strategy of Abraham et al. [6], and thus suffers from a pricing problem that ran in time exponential in cycle and chain cap. As shown in Section 3.2, Glorie et al. [100] and Plaut et al. [174] gave polynomial pricing algorithms for cycles—but not chains—in the *deterministic* case. Using our algorithm from Plaut et al. [174] as a subroutine (given explicitly as Algorithm 4 in Appendix A), we present an algorithm which solves the failure-aware, or *discounted*, pricing problem for cycles in polynomial time, under the restriction that all edges have equal success probability p.

As discussed in Section 3.2, in the deterministic setting, the price of a cycle c is $\sum_{(i,j)\in c} w_{ij} - \sum_{j\in c} \delta_j$, where w_{ij} is the weight of edge (i,j), and δ_j is the dual value of vertex j in the LP. The *discounted* price of a cycle is $p^{|c|} \sum_{(i,j)\in c} w_{ij} - \sum_{j\in c} \delta_j$. Since the utility of an edge now depends on what cycle it ends up in, we cannot collapse edge weights and dual values without knowing the length of the cycle containing it.

With this motivation, we augment the algorithm to run $O(L)$ iterations for each source vertex: one for each possible final cycle length. On each iteration, we know exactly how much edge weights will be worth in the final cycle, so we can reduce the discounted pricing problem to the deterministic pricing problem.

Pseudocode for the failure-aware cycle pricing algorithm is given by GETDISCOUNTEDPOSITIVEPRICECYCLES. Let w and δ be the edge weights and dual values respectively in the original graph. The function GETNEGATIVECYCLES is the adaptation of the deterministic pricing algorithm due to Glorie et al. [100] that we presented in Plaut et al. [174] which returns at least one negative cycle of length at most K, or shows that none exist; we give that pseudocode explicitly in Appendix A.4.3 as Algorithm 4.

The algorithm of Plaut et al. [174] has complexity $O(|V||A|L^2)$. Considering all $L-1$ possible cycle lengths brings the complexity of our algorithm to $O(|V||A|L^3)$.

Theorem 7. *If there is a discounted positive price cycle in the graph, Algorithm 2 will return at least one discounted positive price cycle.*

Algorithm 2 Polynomial-time failure-aware pricing for cycles.

function GETDISCOUNTEDPOSITIVEPRICECYCLES($G = (V, E), L, p, w, \delta$)
  $\mathcal{C} \leftarrow \emptyset$
  for each $k \in \{2, \ldots, L\}$ **do** ▷ Consider all possible cycle lengths
    $w_k(i, j) \leftarrow \delta_j - p^k w_{ij} \ \ \forall (i, j) \in E$ ▷ Reduction of Glorie et al. [100]
    $\mathcal{C} \leftarrow \mathcal{C} \cup$ GETNEGATIVECYCLES(G, k, w_k)
  return $\mathcal{C}$

3.3.6 Conclusions & final thoughts

In this chapter, we addressed the tractable clearing of kidney exchanges with short cycles and long, but bounded, chains. This is motivated by kidney exchange practice, where chains are often long but bounded in length due to post-match edge failure—which we will discuss in much greater depth in Chapter 5. We introduced three IP formulations, two of which are compact, and favorably compared their LPRs to a state-of-the-art formulation with a tight relaxation. Then, on real data from the UNOS US nationwide exchange and the NLDKSS United Kingdom nationwide exchange, as well as on generated data, we showed that our new models outperform all other solvers on realistically-parameterized kidney exchange problems–often dramatically. We also explored practical extensions of our models, such as the use of branch and price for additional scalability, and an extension to the failure-aware kidney exchange case that more accurately mimics reality.

Beyond the immediate importance of more scalable static kidney exchange solvers for use in fielded exchanges, solvers like the ones presented in this chapter are of practical importance in more advanced—and as yet unfielded—approaches to clearing kidney exchange. In reality, patients and donors arrive to and depart from the exchange dynamically over time [213]. As we will see in Chapters 6 and 8, approaches to clearing *dynamic kidney exchange* often rely on solving the static problem many times [14, 27, 70, 73, 98]; thus, faster static solvers result in better dynamic exchange solutions.

Small changes in problem definition often change the complexity of a problem from being solvable optimally in polynomial time to being NP-hard.

– Samir Khuller

And if you think the bit is a big deal, consider the atom.

– Uber PR

4

A new model for kidney exchange

In this chapter, we take a complementary approach to tackling the complexity of the clearing problem. Rather than choosing to solve the NP-hard clearing problem to optimality with techniques like those described in Chapter 3, we instead introduce a novel model for kidney exchange that explicitly takes into account all attributes of the participating patients and donors in the formulation. Under the assumption that real kidney exchange graphs can be represented using just a constant number of attributes—like blood type, weight, various aspects of medical history, insurance details, and so on—we show that our model permits polynomial-time solutions to central NP-hard problems in general kidney exchange. Inspired by classical results from intersection graph theory, we give conditions on the representation of arbitrary graphs in our model, and generalize to the case where participants are allowed to have a thresholded number of negative interactions between attributes. Noting that real-life kidney exchange graphs are *not* arbitrary, we show on actual data from the United Network for Organ Sharing (UNOS) US-wide kidney exchange that our model permits lossless representation of true graphs with far fewer attributes than the worst-case theoretical results require.

Related Publications

A version of the work in this chapter is available on arXiv, and is an ongoing collaboration between Dickerson, Kazachkov, Procaccia, and Sandholm [79].

4.1 A new model for kidney exchange

In this section, we formalize our model of kidney exchange. We prove that under this model certain well-known NP-hard problems in general kidney exchange are solvable in polynomial time. We

also show that, given a compatibility graph, determining the best set of attributes to change (at some cost) is solvable in polynomial time.

4.1.1 Notation & preliminaries

As in earlier chapters, we represent a kidney exchange by a directed *compatibility graph* $G = (V, E)$. Each patient-donor pair, or unpaired altruistic donor, forms a vertex $v \in V$, and a directed edge exists from one vertex to another if the donor at the former can give to the patient at the latter, i.e., are compatible [184, 185, 186].

We consider a model that imposes additional structure on an arbitrary compatibility graph. For each vertex $v_i \in V$, associate with its constituent donor and patient attribute vectors $\mathbf{d}_i$ and $\mathbf{p}_i$, respectively. Here, the qth element d_i^q of $\mathbf{d}_i$ takes on one of a fixed number of types—for example, one of four blood types (O, A, B, AB), or one of a few hundred standard insurance plans. Then, for $v_i \neq v_j \in V$, we define a *compatibility function* $f(\mathbf{d}_i, \mathbf{p}_j)$, a boolean function that returns the compatibility of the donor of v_i with the patient of v_j.

Given V and associated attribute vectors, we can uniquely determine a compatibility graph $G = (V, E)$ such that $E = \{(v_i, v_j) : f(\mathbf{d}_i, \mathbf{p}_j) = 1 \ \forall v_i \neq v_j \in V\}$. We claim that this model accurately mimics reality, and we later support that claim with strong experimental results on real-world data. Furthermore, under this new model, certain complexity results central to kidney exchange change (for the better), as we discuss next.

4.1.2 The clearing problem is easy (in theory)

We now tackle the central computational challenge of kidney exchange: the clearing problem. As discussed earlier, the clearing problem on general graphs is NP-hard [6, 41]. We show that, in our model, the clearing problem itself is solvable in polynomial time; later, we show that the restrictions imposed by our model to achieve this reduction in complexity do not preclude representation of real kidney exchange graphs.

Formally, we are interested in a polynomial-time algorithm that solves the L-CYCLE-COVER problem—that is, finding the largest disjoint packing of cycles of length at most L. For ease of exposition, in this section we use "cycles" to refer to both cycles and chains; indeed, it is easy to see that altruist donors are equivalent to standard patient-donor pairs with a patient who is compatible with all non-altruist vertices in the pool. Then, a chain is equivalent to a cycle with a "dummy" edge returning to the altruist. Also, again for ease of exposition, we assume the value of a chain of length L is equal to a cycle of length L, due to the donor at the end of the chain giving to a patient on the deceased donor waiting list.

Recall that we are working in a model where each vertex v_i belongs to one of a *fixed* number of types determined solely by its attribute vectors $\mathbf{d}_i$ and $\mathbf{p}_i$. Let Θ be the set of all possible types, and $\theta \in \Theta$ represent one such individual type. Then, with a slight abuse of notation, we can define a type compatibility function $f(\theta, \theta') = 1$ if and only if there is a directed edge between vertices of type θ and θ'. (Note that this captures chains and altruist donors as described above.)

A key observation of this section is that any additional edge structure that is imposed on the graph—such as a cycle cover—would be independent of the identity of *specific* vertices, rather, it

would only depend on their types, as vertices of the same type have the exact same incoming and outgoing neighborhoods. For example, in any cycle cover, if v_i and v_j are two vertices of the same type, we can insert v_j in place of v_i, and v_i in place of v_j, and obtain a feasible cycle cover of the same size. This observation drives our theoretical algorithmic results.

In more detail, for a vector of types $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k) \in \Theta^k$, let us denote $f_C(\boldsymbol{\theta}) = 1$ if and only if $f(\theta_t, \theta_{t+1}) = 1$ for all $t < k$, and $f(\theta_k, \theta_1) = 1$. In other words, $f_C(\boldsymbol{\theta}) = 1$ if every k vertices $i_1, \ldots, i_k$ of types $\theta_1, \ldots, \theta_k$, respectively, are involved in a cycle in the graph. Furthermore, for $L \leq n = |V|$, denote

$$\mathcal{T}(L) = \{\boldsymbol{\theta} \in \Theta^k : k \leq L \text{ and } f_C(\boldsymbol{\theta}) = 1\}.$$

That is, $\mathcal{T}(L)$ contains all vectors of types that induce feasible cycles of length at most L.

Now consider the following algorithm for L-CYCLE-COVER in our model:

Algorithm 3 L-CYCLE-COVER

1. $\mathcal{C}^* \leftarrow \emptyset$
2. **for** every collection of numbers $\{m_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta} \in \mathcal{T}(L)}$ such that $\sum_{\boldsymbol{\theta} \in \mathcal{T}(L)} m_{\boldsymbol{\theta}} \leq n$
   - **if** there exists cycle cover $\mathcal{C}$ such that $\|\mathcal{C}\|_V > \|\mathcal{C}^*\|_V$ and for all $\boldsymbol{\theta} \in \mathcal{T}(L)$, $\mathcal{C}$ contains $m_{\boldsymbol{\theta}}$ cycles consisting of vertices of the types in $\boldsymbol{\theta}$ **then** $\mathcal{C}^* \leftarrow \mathcal{C}$
3. **return** $\mathcal{C}^*$

Here, $\|\mathcal{C}\|_V$ denotes the number of unique *vertices* matched in a cycle cover $\mathcal{C}$. We claim that, in our setting, Algorithm 3 is optimal and computationally efficient.

Theorem 8. *Suppose that L and $|\Theta|$ are constants. Then Algorithm 3 is a polynomial-time algorithm for L-CYCLE-COVER.*

Proof. We start by verifying that Algorithm 3 is indeed optimal. Consider the optimal cycle cover $\mathcal{C}^*$. For each $\boldsymbol{\theta} \in \mathcal{T}(L)$, let $m^*_{\boldsymbol{\theta}}$ be the number of cycles in $\mathcal{C}^*$ that are consistent with the types in $\boldsymbol{\theta}$. Clearly $\sum_{\boldsymbol{\theta} \in \mathcal{T}(L)} m^*_{\boldsymbol{\theta}} \leq n$, as there are only n vertices so there cannot be more than n cycles (in fact, $n/2$ is also a valid upper bound). Therefore, Algorithm 3 considers the collection of numbers $m^*_{\boldsymbol{\theta}}$ in Step 2. Because this collection of numbers does induce a valid cycle cover that is of the same size as $\mathcal{C}^*$, the algorithm would update its incumbent cycle cover if it was not already optimal.

We next analyze the running time of the algorithm. First, note that it is straightforward to check whether the numbers $\{m_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta} \in \mathcal{T}(L)}$ induce a valid cycle cover. Since $\mathcal{T}(L)$ consists only of valid cycles according to the compatibility function f_C, we just need to check that there are enough vertices of type θ to construct all the cycles that require them. This simply amounts to multiplying each $m_{\boldsymbol{\theta}}$ by the number of times type θ appears in $\boldsymbol{\theta}$, and verifying that the sum of these products over all $\boldsymbol{\theta}$ in $\mathcal{T}(L)$ is at most the number of vertices of type θ.

Second, we argue that there is only a polynomial number of possibilities to construct a collection of numbers $\{m_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta} \in \mathcal{T}(L)}$ such that $\sum_{\boldsymbol{\theta} \in \mathcal{T}(L)} m_{\boldsymbol{\theta}} \leq n$. Indeed, this number is at most $(n+1)^{|\mathcal{T}(L)|}$. Moreover, $|\mathcal{T}(L)| \leq L \cdot |\Theta|^L$. Because $|\Theta|$ and L are constants, $|\mathcal{T}(L)|$ is also a constant. The expression $(n+1)^{|\mathcal{T}(L)|}$ is therefore a polynomial in n. □

Even for constant L, the running time of Algorithm 3 is exponential in k. But this is to be expected. Indeed, any graph can trivially be represented using a set Θ of types of size n, where each vertex has a unique type, and a compatibility function f_C that assigns 1 to an ordered pair of types if the corresponding edge exists in G. Therefore, if the running time of Algorithm 3 were polynomial in n *and* k, we could solve the general L-CYCLE-COVER problem in polynomial time—and that problem is NP-hard [6].

4.1.3 Flipping attributes is also easy (in theory)

While patients and donors in a kidney exchange are endowed with an initial set of attributes, it may be possible in practice to—at a cost—change some number of those attributes to effect change in the final matching. For example, the human body naturally tries to reject, to varying degrees, a transplanted organ. Due to this, nearly all recipients of kidneys are placed on immunosuppressant drugs after transplantation occurs.[1] However, *preoperative* immunosuppression can also be performed to increase transplant opportunity—but at some cost to the patient's overall health.

With this in mind, we extend the model of Section 4.1.2 as follows. Associate with each pair of types $\theta, \theta' \in \Theta$ a cost function $c : \Theta \times \Theta \to \mathbb{R}$ representing the cost of changing a vertex of type θ to type θ'. Then, the L-FLIP-AND-CYCLE-COVER problem is to find a disjoint packing of cycles of length at most L that maximizes the size of the packing minus the sum of costs spent changing types. Building on Theorem 8, this problem is also solvable in polynomial time.

Theorem 9. *Suppose that L and $|\Theta|$ are constants. Then L-FLIP-AND-CYCLE-COVER is solvable in polynomial-time.*

Proof sketch. For any type $\theta_i \in \Theta$, there are n_i vertices. Then, for each of the $(|\Theta| - 1)$ choices of which type $\theta \neq \theta_i$ to switch to, choose how many vertices from θ_i will switch to a different type; there are $(n_i + 1)$ choices. Do this for all $|\Theta|$ types, resulting in $\prod_{\theta_i \in \Theta}[(n_i + 1)(|\Theta| - 1)]$ choices. Note that $\sum_{\theta_i \in \Theta}(n_i + 1) = n + |\Theta|$, meaning $\prod_{\theta_i \in \Theta}[(n_i + 1)(|\Theta| - 1)] \leq ((n + |\Theta|) \cdot (|\Theta| - 1)/|\Theta|)^{|\Theta|} \leq (n + |\Theta|)^{|\Theta|}$. Since $|\Theta|$ is a constant, this is polynomial in n; invoking an adaptation of the polynomial time Algorithm 3 that subtracts out $c(\theta, \theta')$ for every vertex that switches from θ to θ', for each of the polynomially-many choices, concludes the proof. □

4.2 A concrete instantiation: thresholding

As motivated in Chapter 1 and Section 4.1, compatibility in real kidney exchange graphs is determined by patient and donor attributes, such as blood or tissue type. In particular, if an attribute for a donor and patient is in conflict, they are deemed incompatible. Motivated by that reality, in this section, we associate with each patient and donor a bit vector of length k, and count incompatibilities based on any shared activated bits between a patient and potential donor.

As a concrete example, consider human blood types. At a high level, human blood contains A antigens, B antigens, both (type AB), or neither (type O). AB-type patients can receive from any donor, A-type (B-type) can receive from O-type and A-type (B-type) donors, and O-type patients

[1] https://www.kidney.org/atoz/content/immuno

can only receive from O-type donors. In our bit model, this is represented with $k = 2$, such that a donor's first (resp. second) bit is set if his blood holds A (resp. B) antigens. and a patient's first (resp. second) bit is set if she cannot receive from blood containing A (resp. B) antigens. Thus, the type space $\Theta = 2^{\{\texttt{has-A},\texttt{has-B}\}} \times 2^{\{\texttt{no-A},\texttt{no-B}\}}$; in general, $|\Theta| = 2^{2k}$.

Formally, unless otherwise stated, throughout this section G will refer to a directed graph with vertex set $V = [n] := \{1, \ldots, n\}$ and edge set E, and with each $i \in V$ associated with two k-bit vectors $\mathbf{d}_i, \mathbf{p}_i \in \{0,1\}^k$. Let $Q_d(i) = \{q \in [k] : \mathbf{d}_{iq} = 1\}$ be the set of conflict bits for the donor associated with vertex $i \in V$, and similarly let $Q_p(i) = \{q \in [k] : \mathbf{p}_{iq} = 1\}$. For $i, j \in V$ such that $i \neq j$, the threshold feasibility function f^t_{thresh} is defined as

$$f^t_{\mathsf{thresh}}(\mathbf{d}_i, \mathbf{p}_j) = \begin{cases} 1 & \text{if } |Q_d(i) \cap Q_p(j)| \leq t, \\ 0 & \text{otherwise.} \end{cases}.$$

Note that $|Q_d(i) \cap Q_p(j)| \leq t$ if and only if $\langle \mathbf{d}_i, \mathbf{p}_j \rangle \leq t$.

Kidney exchange graphs constructed using threshold compatibility functions are closely related to complements of *intersection graphs* [156], which are graphs that have a set associated with each vertex and an edge between two vertices if and only if the sets intersect. Given a nonnegative integer t, the function f^t_{thresh} is related to *p-intersection graphs* [62, 83], in which an edge exists between two vertices if their corresponding sets intersect in at least $p \geq 1$ elements.

In particular, our model is similar to that of *intersection digraphs* [194], or equivalently *bipartite intersection graphs* [105], both also considered in [170]. Both of these have mainly been studied under the assumption that the sets used to represent the graph have the "consecutive ones" property, i.e., each set is an interval from the set of integers. Our model is more general: we do not place such an assumption on the set of conflict bits. Moreover, most treatments of intersection digraphs consider loops on the vertices, whereas in the thresholding model we have defined, whether or not donor i and patient i are compatible is not considered. In addition, the directed and bipartite intersection graph literature has focused on the case that $t = 0$ (in our terminology). To the best of our knowledge, the work in this chapter is the first treatment *p-intersection digraphs*, and certainly their first real-world application.

4.2.1 Existence of small representations

It is natural to ask for what values of t and k can we select vertices with bit vectors $\mathbf{d}_i$ and $\mathbf{p}_i$ of length k such that f^t_{thresh} can create *any* graph of a specific size?

Formally, we say that G is *(k,t)-representable* (by feasibility function f^t_{thresh}) if, for all $i \in V$ there exist $\mathbf{d}_i, \mathbf{p}_i \in \{0,1\}^k$ such that for all $j_1 \in V$, $j_2 \in V \setminus \{j_1\}$, $(j_1, j_2) \in E$ if and only if $f^t_{\mathsf{thresh}}(\mathbf{d}_{j_1}, \mathbf{p}_{j_2}) = 1$.

It is known [89] that any graph can be represented as an intersection graph with $k \leq n^2/4$. In contrast, the next theorem shows that, in our model, $k \leq n$ suffices to represent any graph. It is akin to a result on the *term rank* of the adjacency matrix of $\overline{G}$ [170, Theorem 6.6].

Theorem 10. *Let $G = (V, E)$ be a digraph on n vertices. Let n_1 be the number of vertices with outgoing edges, Let n_2 be the number of vertices with incoming edges, and $n' = \min\{n_1 + 1, n_2 + 1, n\}$. Then G can be $(n', 0)$-represented.*

Proof. We first show that the graph can be $(n_1 + 1, 0)$-represented. Assume without loss of generality that vertices $1, \ldots, n_1$ have outgoing edges. We show how to set $\mathbf{d}_i, \mathbf{p}_i \in \{0,1\}^{n_1+1}$ for each vertex i in V. To set the donor attributes, for each $i \in [n_1]$, let $\mathbf{d}_i$ be e_i, the ith standard basis vector, i.e., the vector of length $n_1 + 1$ with a 1 in the ith coordinate and 0 everywhere else. For $i > n_1$, set $\mathbf{d}_i$ to be e_{n_1+1}. For the patient attributes of vertex $j \in [n]$, for each $i \in [n]$ such that $(i, j) \in E$, set $\mathbf{p}_{ji} = 0$, and set $\mathbf{p}_{ji} = 1$ otherwise. Note that if all the vertices have outgoing edges, then $n_1 = n$ unit vectors suffice. A similar approach works to $(\min\{n, n_2 + 1\}, 0)$-represent G, by using the n_2 unit vectors as the patient vectors of those vertices with incoming edges, and (if needed) one additional unit vector for any remaining vertices. In both of these cases, $\langle \mathbf{d}_i, \mathbf{p}_j \rangle = 0$ if and only if $(i, j) \in E$, which represents G by f^0_{thresh}. □

In particular, Theorem 10 implies that any graph is $(n, 0)$-representable. The next theorem shows a matching lower bound. The same construction and bound also hold if loops are considered [194].

Theorem 11. *For any $n \geq 3$, there exists a graph on n vertices that is not $(k, 0)$-representable for all $k < n$.*

Proof. Define G to be the digraph on n vertices, $V = [n]$, with an edge from vertex i, for each $i \in V$, to every vertex except $i - 1$ (and itself), where vertex n is also referred to as vertex 0.

Assume that G is $(k, 0)$-representable, and consider vertex 1. Since $(1, n) \notin E$, and $(i, n) \in E$ for all $i \notin \{1, n\}$, there exists a conflict bit $q_1 \in Q_d(1) \cap Q_p(n)$ such that $q_1 \notin Q_p(V \setminus \{1, n\})$. More generally, there exists such a conflict bit q_i for all $i \in V$.

We claim that these conflict bits are all unique, which directly implies that $k \geq n$. Indeed, otherwise we can assume that $q_1 = q_i$ for some $i \neq 1$ (without loss of generality, as the graph is symmetric subject to cyclic permutations). But then $(1, i - 1)$ and (i, n) do not appear as edges in G, which is not true for any $i \in V \setminus \{1\}$. □

More generally, it is easy to see that any graph that is $(k, 0)$-representable is also $(k + t, t)$-representable for any $t \geq 0$. Indeed, simply take the $(k, 0)$-representation of the graph, and append t ones to every vector. Together with Theorem 10, this shows that any graph is $(n + t, t)$-representable. However, the lower bound given by Theorem 11 does not extend to $t > 0$. We conjecture that for any n and t there exists a graph that can only be (k, t)-represented with $k = \Omega(n)$—this remains an open question.

4.2.2 Computational issues

Given any real compatibility graph with n vertices, we know by Theorem 10 that we can $(k, 0)$-represent that graph for $k = n$. But, in practice, how large of a k do we actually need?

Various problems related to intersection graphs are NP-complete for general graphs [137, 170], but we work in a setting with additional structure. And while we do not show that finding a (k, t)-representation is NP-hard, we do show that a slightly harder problem, which we refer to as (k, t)-REPRESENTATION WITH IGNORED EDGES, is NP-hard. Given an input of a directed graph $G = (V, E)$, a subset F of $\binom{V}{2}$, and integers $k \geq 1$ and $t \geq 0$, this problem asks whether there exist

bit vectors $\mathbf{d}_i$ and $\mathbf{p}_i$ of length k for each $i \in V$ such that for any $(i, j) \in F$, we have $(i, j) \in E$ if and only if $\langle \mathbf{d}_i, \mathbf{p}_j \rangle \leq t$.

Theorem 12. *The* (k, t)-REPRESENTATION WITH IGNORED EDGES *problem is NP-complete.*

The theorem's nontrivial proof is given in its entirety in Appendix B. Here we give a proof sketch. One major idea is the construction of a *bit-grounding gadget* G_k—a subgraph where the bits are set uniquely (up to permutations) in any valid representation, and can be used to set the bits in other vertices. The gadget has $\binom{k}{2}$ vertices; we prove that there is a unique (up to permutations) $(k, 1)$-representation of G_k, where each donor vector has a unique pair of ones, and similarly for each patient vector. Figure B.1 shows G_4.

Then, we prove NP-hardness by reduction from 3SAT. In the constructed instance of our problem, we set the threshold to 1. The crux of the reduction is to add a vertex for each clause in the given 3SAT formula, where in the patient vector, the bit corresponding to each literal in the clause is set to 1. This can be done by connecting the vertex to the bit-grounding gadget. Moreover, there is a special vertex that does not have outgoing edges to any of the clause vertices. This means that it must have a 1 in a position that corresponds to one of the literals in each clause. A different part of the construction ensures that there is at most a single 1 in the two positions corresponding to a variable and its negation. Therefore, a valid assignment of the donor bits corresponds to a satisfying assignment for the 3SAT formula.

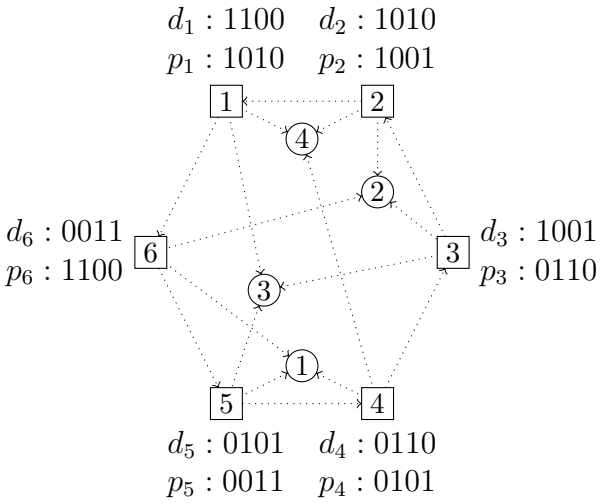


Figure 4.1: Gadget G_4 with a subset of *non*-edges shown; all edges between circle vertices (those in G_4^2) are also not in E.

4.3 Computing small representations of real kidney exchange compatibility graphs

In this section, we test our hypothesis that real compatibility graphs can be represented by a substantially smaller number of attributes than required by the worst-case theoretical results of Section 4.2. We begin by presenting general mathematical programming techniques to determine, given $k, t \in \mathbb{Z}$, whether a specific graph $G = (V, E)$ is (k, t)-representable. We then show on real and generated compatibility graphs from the UNOS US-wide kidney exchange that small k suffices for $(k, 0)$-representation, and conclude by exploring the allowance of greater thresholds t on match size. We find even small thresholds $t > 0$ result in substantial societal gain.[2]

4.3.1 Mathematical programming formulations

Implementation of f^t_{thresh} can be written succinctly as a quadratically-constrained discrete feasibility program (QCP) with $2k|V|$ binary decision variables, given as M1 below.

$$\begin{aligned} \langle \mathbf{d}_i, \mathbf{p}_j \rangle &\leq t && \forall (v_i, v_j) \in E \\ \langle \mathbf{d}_i, \mathbf{p}_j \rangle &\geq (t+1) && \forall (v_i, v_j) \notin E \\ \mathbf{d}_i, \mathbf{p}_i &\in \{0,1\}^k && \forall v_i \in V \end{aligned} \tag{M1}$$

The constraint matrix for this program is not positive semi-definite, and thus the problem is not convex. Exploratory use of heuristic search via state-of-the-art integer nonlinear solvers [46] resulted in poor performance (in terms of runtime and solution quality) on even small graphs. With that in mind, and motivated by the presence of substantially more mature integer *linear* program (ILP) solvers, we linearize M1, presented as M2 below.

$$\begin{aligned} \min \quad & \textstyle\sum_{v_i \in V} \sum_{v_j \neq v_i \in V} \xi_{ij} && \\ \text{s.t.} \quad & d^q_i \geq c^q_{ij} \wedge p^q_j \geq c^q_{ij} && \forall v_i \neq v_j \in V, q \in [k] \\ & d^q_i + p^q_j \leq 1 + c^q_{ij} && \forall v_i \neq v_j \in V, q \in [k] \\ & \textstyle\sum_q c^q_{ij} \leq t + (k-t)\xi_{ij} && \forall (v_i, v_j) \in E \\ & \textstyle\sum_q c^q_{ij} \geq (t+1)\xi_{ij} && \forall (v_i, v_j) \in E \\ & \textstyle\sum_q c^q_{ij} \geq t + 1 - k\xi_{ij} && \forall (v_i, v_j) \notin E \\ & \textstyle\sum_q c^q_{ij} \leq k - (k-t)\xi_{ij} && \forall (v_i, v_j) \notin E \\ & d^q_i, p^q_i \in \{0,1\} && \forall v_i \in V, q \in [k] \\ & c^q_{ij}, \xi_{ij} \in \{0,1\} && \forall v_i \neq v_j \in V, q \in [k] \end{aligned} \tag{M2}$$

M2 generalizes M1; while M1 searches for a feasible solution to the (k, t)-representation problem, M2 searches for the "best" (possibly partially-incorrect) solution by minimizing the total number of edges that exist in the solution but not in the base graph G, or do not exist in the solution but do in G. This flexibility may be desirable in practice to strike a tradeoff between small k and accuracy of representation.

[2] All code for this section can be found at https://github.com/JohnDickerson/KidneyExchange.

Interestingly, neither the fully general ILP nor its (smaller) instantiations for the special cases of feasibility and/or threshold $t = 0$ were solvable by a leading commercial ILP solver [118] within 12 hours for even small graphs, primarily due to the model's loose LP relaxation. Indeed, the model we are solving is inherently logical, which is known to cause such problems in traditional mathematical programming [109]. With that in mind, we note that the special case of $t = 0$ can be represented compactly as a satisfiability (SAT) problem in conjunctive normal form, given below as M3.

$$\begin{aligned} & \bigwedge_{q \in [k]} (\neg d_i^q \vee \neg p_j^q) \quad \forall (v_i, v_j) \in E \\ & (z_{ij}^1 \vee z_{ij}^2 \vee \ldots \vee z_{ij}^k) \wedge \\ & \bigwedge_{q \in [k]} \left[(\neg z_{ij}^q \vee d_i^q) \wedge (\neg z_{ij}^q \vee p_j^q) \right] \quad \forall (v_i, v_j) \notin E \end{aligned} \tag{M3}$$

This formulation maintains two sets of clauses: the first set enforces no bit-wise conflicts for edges in the underlying graph, while the second set enforces at least one conflict via k auxiliary variables $z_{ij}^{\cdot}$ for each possible edge $(v_i, v_j) \notin E$. M3 was amenable to parallel SAT solving [39]. Next, we present results on real graphs using this formulation.

4.3.2 $(k, 0)$-representations of real kidney exchange graphs

Can real kidney exchange graphs be represented by a small number of attributes? To answer that question, we begin by testing on real match run data from the first two years of the United Network for Organ Sharing (UNOS) kidney exchange. We translate each compatibility graph into a CNF-SAT formulation according to M3, and feed that into a SAT solver [39] with access to 16GB of RAM, 4 cores, and 60 minutes of wall time. (Timeouts are counted—conservatively against our chapter's qualitative message—as negative answers.)

Figure 4.2 shows a classical phase transition from unsatisfiability to satisfiability as k increases as a fraction of graph size, as well as an associated substantial increase in computational intractability centered around that phase transition. This phenomenon is common to many central problems in artificial intelligence [59, 108, 218]. Indeed, we see that substantially fewer than $|V|$ attributes are required to represent real graphs; compare with the lower bound of Theorem 11.

Figure 4.3 explores the minimum k required to represent each graph as a function of $|V|$, compared against the theoretical upper bound of $k = |V|$. The shaded area represents those values of k where the SAT solver timed out; thus, the reported values of k are a conservative *upper* bound on the required minimum, which is still substantially lower than $|V|$.

4.3.3 Thresholding effects on matching size

One motivation of this work is to provide a principled basis for optimally "flipping bits" of participants (via, e.g., immunosuppresion) in fielded kidney exchanges, in the hope that additional edges in the compatibility graph will result in gains in the final algorithmic matchings. We now explore this line of reasoning—that is, increasing the t in f^t_{thresh} instead of the k, which is now endogenous to the underlying model—on realistic generated UNOS graphs of varying sizes.

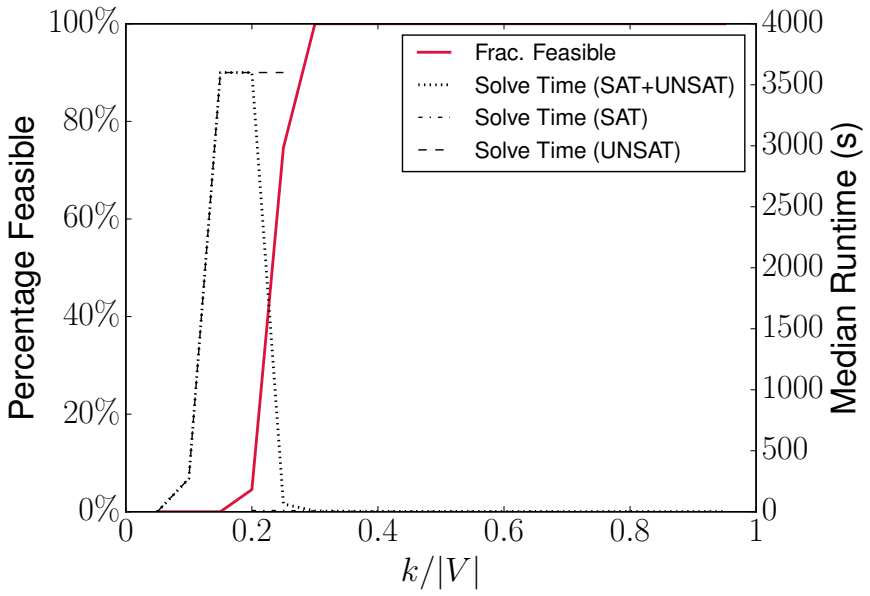


Figure 4.2: Classical hardness spike near the phase transition for $(k, 0)$-representation on real UNOS compatibility graphs.

Figure 4.4 shows the effect on the percentage of patient-donor pairs matched by 2- and 3-cycles as a global threshold t is raised incrementally from $t = 0$ (the current status quo) to $t = 5$. Intuitively, larger compatibility graphs result in a higher fraction of pairs being matched; however, a complementary approach—making the graph denser via even small increases in t—also results in tremendous efficiency gains of 3–4x (depending on $|V|$) over the baseline for $t = 1$, and quickly increasing to all pairs being matched by $t = 5$.

We note that any optimal matching found after increasing a global threshold t could also be created by paying to change at most t bits per vertex in a graph; however, the practical *selection* of the minimum-sized set of at most t bits per vertex such that the size of the resulting optimal matching is equal to that found under the global threshold of t is a difficult two-stage problem and is left as future research. The large efficiency gains realized by moving from f^0_{thresh} to even f^1_{thresh} motivate this direction of research.

4.4 Conclusions & future research directions

Motivated by the increasing size of real-world kidney exchanges, in this chapter, we presented a compact approach to modeling kidney exchange compatibility graphs. Our approach is intimately connected to classical intersection graph theory, and can be viewed as the first exploration and practical application of p-intersection digraphs. We gave necessary and sufficient conditions for losslessly shrinking the representation of an arbitrary compatibility graph in this model. Real compatibility graphs, however, are not arbitrary, and are created from characteristics of the patients and donors; using real data from the UNOS US-wide kidney exchange, we showed that using only

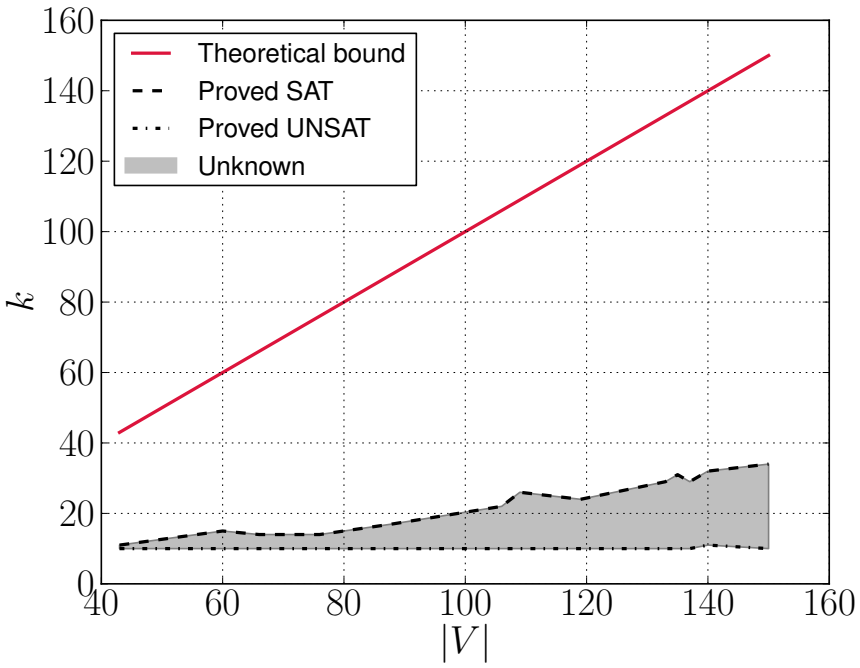


Figure 4.3: Comparison of number of bits (y-axis) required to $(k, 0)$-represent real UNOS compatibility graphs of varying sizes (x-axis). The theoretical bound of $k = |V|$ is shown in red; it is substantially higher than the conservative upper bound of k solved by our SAT solver (upper dotted line).

a small number of attributes suffices to represent real graphs. This observation is of potential practical importance; if all the real graphs can be represented by a constant number of attributes, then central NP-hard problems in general kidney exchange are solvable in polynomial time.

The work in this chapter is still early stage, and is not yet ready to be deployed. For instance, while Algorithm 3 runs in polynomial time, the order of that bounding polynomial is clearly quite large. Furthermore, our exact methods for computing (k, t)-representations are still not scalable; progress in that direction would allow us to shrink the exponent in that polynomial. Yet, we are excited: an alternate representation for kidney exchange, such that the central problem (that of finding the best disjoint cycle and chain packing) is theoretically easier than it is in the status quo model, may allow us to more tractably approach problems in kidney exchange that are as-yet unsolved—and thus not fielded. One of these is the optimal flipping of attributes—in kidney exchange, via immunosuppression—at some cost, either in a static (as we briefly discussed in Section 4.1.3) or dynamic model (left as future work). This is already being discussed at a number of medical centers; a tractable computational approach to supporting that medical decision will be of great practical aid.

This chapter only addresses the representation of static compatibility graphs; in reality, exchanges are dynamic, with patients and donors arriving and departing over time [213]; we discuss this further in Chapter 6. Extending the proposed method to cover time-evolving graphs is of independent theoretical interest, but may also be useful in speeding up the dynamic clear-

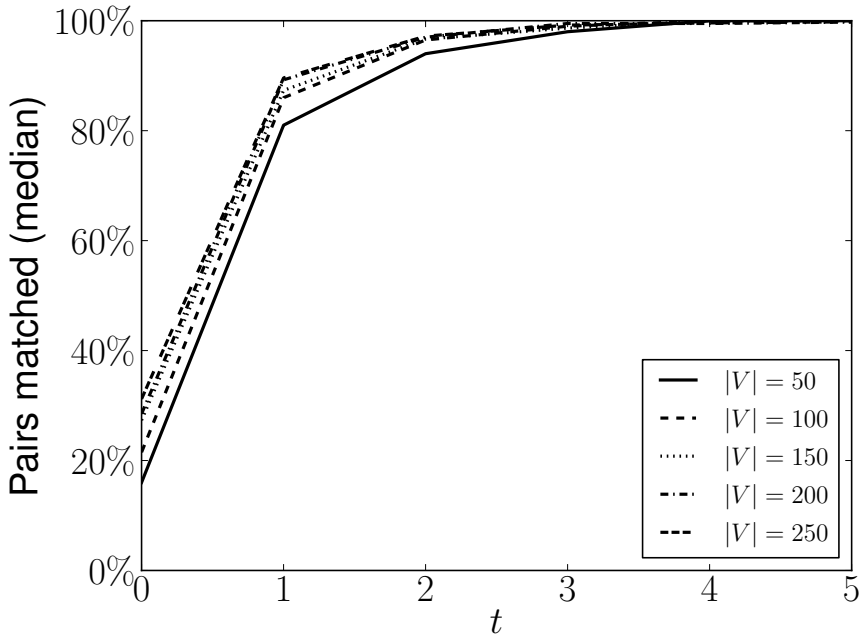


Figure 4.4: Pairs matched (%, y-axis) in generated UNOS graphs of varying sizes (lines), as t increases (x-axis).

ing problem [14, 27, 70, 73, 98]. Better exact and approximate methods for computing (k, t)-representations of graphs would likely be a prerequisite for that line of research. Furthermore, adaptation of the theoretical results to alternate organ models like lung [90, 149], liver [91], and multi-organ [72] exchange (discussed in Chapter 9) would be of practical use.

PART II:

Managing Uncertainty in Dynamic Matching Environments

In real-world applications of Linear Programming, one cannot ignore the possibility that a small uncertainty in the data can make the usual optimal solution completely meaningless from a practical viewpoint.

– Aharon Ben-Tal & Arkadi Nemirovski

5

Optimization in the face of edge failure

This part of the thesis addresses the management of both short-term and long-term uncertainty in kidney exchange. In this chapter, we propose quantitative approaches to one of the main problems kidney exchanges face today: “last-minute" failures. We mean failures before the transplant surgery takes place, not failures during or after it. Amazingly, most planned matches fail to go to transplant! In the case of the UNOS exchange, roughly 90% of matches fail [131], and most matches fail at other kidney exchanges as well [e.g., 22, 48, 143]. There are myriad reasons for these failures, as we will detail in this chapter. We also address last-minute failures, albeit from a complementary point of view where information is gathered before the matching algorithm is run at all. Finally, Chapter 6 addresses longer-term uncertainty in kidney exchange, where vertices and edges arrive and depart over multiple matching periods; we will refer to that as “dynamic” kidney exchange.

To address last-minute failures, in this chapter we propose to move away from the deterministic clearing model used by kidney exchanges today, and discussed extensively in Chapter 3, into a probabilistic model where the input includes failure probabilities on possible planned transplants, and the output is a transplant plan with maximum *expected* value. The probabilistic approach has recently also been suggested by others [e.g., 61, 147]. They used a general-purpose integer program solver (Gurobi) to solve their optimization models. We show that general-purpose solvers do not scale to today’s real kidney exchange sizes. Then we develop a custom branch-and-price-based integer program solver specifically for the probabilistic clearing problem, and show that it scales dramatically better. We also present new theoretical and experimental results that show that the probabilistic approach yields significantly better matching than the current deterministic approach. We conduct experiments both in the static and dynamic setting with (to our knowledge) the most realistic instance generators—one due to [191] and one that we created that uses real data from all the UNOS match runs conducted so far—and simulator to date. Perhaps most interestingly, we show that, even when higher edge failure rates are correlated with other marginalizing character-

istics of a vertex, failure-aware matching can simultaneously increase both the overall number of transplants and the number of transplants to these marginalized patients—in both the static and dynamic settings, on real and simulated data.

Related Publications

An early version of some of the work in this chapter appeared at EC-13, and at medical conferences like ATC-13 and WTC-14, with an extensively augmented version under revision at a journal. It is a collaboration between Dickerson, Procaccia, and Sandholm [75, 76]. The work in Section 5.10 appeared at EC-15, and is a collaboration between Blum, Dickerson, Haghtalab, Procaccia, Sandholm, and Sharma [44].

5.1 Related work covering uncertainty in kidney exchange

We begin by overviewing related work in managing both short- and long-term uncertainty in kidney exchange. We address long-term dynamism as "dynamic kidney exchange" in Chapter 6, while this chapter and the next address short-term uncertainty.

Short-term uncertainty (to be addressed in this chapter). In the *stochastic matching* problem, we are given an undirected graph $G = (V, E)$, where we do not know which edges in E actually exist. Rather, for each edge $e \in E$, we are given an existence probability p_e. Of interest are algorithms that first query some subset of edges to find the ones that exist, and based on these queries, produce a matching that is as large as possible. The stochastic matching problem is a special case of *stochastic k-set packing*, where each set exists only with some probability, and the problem is to find a packing of maximum size of those sets that do exist. This problem is addressed in Section 5.10.

Prior work has considered multiple variants of stochastic matching. A popular variant is the *query-commit* problem, where the algorithm is *forced* to add any queried edge to the matching if the edge is found to exist. Considering approximation ratios with respect to the omniscient optimum, Goel and Tripathi [101] establish an upper bound of 0.7916 for graphs in which no information is available about the edges, while Costello et al. [64] establish a lower bound of 0.573 and an upper bound of 0.898 for graphs in which each edge e exists with a given probability p_e. Similarly to the work on which Section 5.10 is based, these approximation ratios are with respect to the omniscient optimum, but the informational disadvantage of the algorithm stems purely from the query-commit restriction.

Within the query-commit setting, another thread of work [7, 29, 60] imposes an additional *per-vertex budget constraint* where the algorithm is not allowed to query more than a specified number, b_v, of edges incident to vertex v. With this additional constraint, the benchmark that the algorithm is compared to switches from the omniscient optimum to the constrained optimum, i.e., the performance of the best decision tree that obeys the per-vertex budget constraints and the query-commit restriction. In other words, the algorithm's disadvantage compared to the benchmark is only that it is constrained to run in polynomial time. Here, again, the best known approximation

ratios are constant. A generalization of these results to packing problems has been studied by Gupta and Nagarajan [103].

Blum et al. [43] consider a stochastic matching setting without the query-commit constraint. They set the per-vertex budget to exactly 2, and ask which subset of edges is queried by the optimal collection of queries subject to this constraint. They prove structural results about the optimal solution, which allow them to show that finding the optimal subset of edges to query is NP-hard. In addition, they give a polynomial-time algorithm that finds an almost optimal solution on a class of random graphs (inspired by kidney exchange settings). Crucially, the benchmark of Blum et al. [43] is also constrained to two queries per vertex. Section 5.10 will present recent work with Blum et al. [44] that looks at a generalization of this problem.

This chapter addresses a complementary problem, where we have knowledge of the probability of existence for each edge, but cannot test edges before running the match. Instead, our problem is to select a set of edges that maximizes the expected utility from the subsequent matching (after coins are flipped to determine if edges exist). Subsequent to the initial posting of the work in this chapter, Anderson [14] studied the problem through the lens of two-stage stochastic optimization (relevent work was also published by Anderson et al. [16] in an online appendix to that full publication). That approach is promising but runs into substantial scalability issues. As-yet-unpublished work due to Glorie et al. [98] approaches the problem from a robust optimization point of view, and considers post-match "failure" of both vertices and edges. Similarly, recent work by Klimentova et al. [135] and subsequently by Pedroso and Ikeda [173] approaches the problem in a similar setting, that of stochastic matching with within-structure (i.e., cycles and chains) recourse; critically, they also provide effective search techniques to make the inclusion of recourse more scalable. Finally, in Chapter 8, we will build on techniques from this chapter, Chapter 6, and Chapter 7 to learn using data how to match in a realistic model of dynamic kidney exchange.

Long-term dynamism (to be addressed in Chapter 6). There has been prior work on the dynamics of kidney exchange, but that work has largely focused on the dynamics driven by pairs and altruists arriving into, and departing from, the exchange rather than on the dynamics driven by failures [9, 15, 24, 26, 213]. We make a distinction here between dynamic exchange and work in the (re)design of deceased-donor online allocation matching policies [37, 202]; we overview this work in greater depth in Chapter 6.

On the theory side, Zenios [223] and Ünver [213] provided initial analysis of dynamic kidney exchange; Ünver [213] shows that greedy matching is optimal in a simplified model, which, for example, does not include transplant chains. Ashlagi et al. [24] work in a model with chains and two types of vertices and characterize different matching policies based on *batching* policies, where vertices of a certain type build up in the pool over time and eventually trigger a myopic match. Anderson et al. [15] work in a similar model and show that greedy matching is generally optimal with respect to minimizing average waiting time in the pool and does not marginalize hard-to-match pairs. However, vertices in their model do not disappear for reasons other than getting matched, which is not realistic, as we discuss in Section 5.8. Akbarpour et al. [9] also analyze a dynamic pool—consisting of only 2-cycles and no chains, but with vertices that expire over time—in the context of a utility function that reflects expected waiting time, and find that sometimes it

is better to batch vertices before matching. Recent work by Ashlagi et al. [26] also investigates building thickness in a dynamic random graph model.

On the practical side, Awasthi and Sandholm [27] use trajectory-based online optimization methods that sample potential future states to inform the matching algorithm what the next action should be. They explore different matching algorithms and batch sizes, and find that non-greedy matching results in a long-term lift in the number of matches. Their method does not scale to very long futures or very large pool sizes. In Chapter 6, we will present a more scalable approach that learns the *potentials* of elements of the current input compatibility graph offline (e.g., potentials of pairs based on features such as donor and patient blood types), translates these to edge weights, and performs a computationally feasible weighted batch matching. None of these papers consider post-algorithmic match failures in their theoretical analysis or in the optimization problem, as we do in this chapter and the next.

As discussed in Chapters 1 and 2, analysis of kidney exchange using random graph models is nowadays the typical method for providing theoretical guidance. Indeed, some of the dynamic kidney exchange papers cited above work in dynamic random graph models [9, 15, 17, 24, 213]. In this chapter, we work with the model of Ashlagi et al. [23]; this is a "sparse" model, as opposed to the related "dense" random models like those of Ashlagi and Roth [19] and Toulis and Parkes [209], which we used in Chapter 2. Next, we discuss the model used in this chapter for "failure-aware" kidney exchange, where post-match edge failure is considered directly in the optimization problem—but edges cannot be queried before the algorithmic matching engine is run.

5.2 Modeling expected utility by considering edge, cycle, and (partial or complete) chain failure

In this section, we augment the standard model of kidney exchange to include the probability of edge, cycle, and chain failure. We formalize the *discounted utility* of an edge, cycle, and chain, which represents the expected number of actual transplants (not just potential transplants). This is used to define the discounted utility of an overall matching, which more accurately reflects its value relative to other matchings.

We begin with the same model of deterministic kidney exchange used in the prior chapters, which instantiates a directed compatibility graph $G(n)$ by constructing one vertex for each patient-donor pair. An edge e from v_i to v_j is added if the patient in v_j wants the donor kidney (or item, in general) of v_i. A donor is willing to give her kidney if and only if the patient in her vertex v_i receives a kidney. Edges e can have associated weights w_e. A *matching* M is then a collection of disjoint cycles and chains in the graph G.

In the basic kidney exchange model, the weight w_c of a cycle or chain c is the sum of its edge weights, and the weight of a matching M is the sum of the weights of the cycles and chains in the matching. The clearing problem is then to find a maximum (weighted) matching M; we presented the current fastest method for doing this in Chapter 3. In reality, not all of the recommended matches proceed to transplantation, due to varying levels of sensitization between candidates and donors in the pool (represented by a scalar factor called CPRA), illness, uncertainty in medical

knowledge, or logistical problems. As such, the weight of a cycle or chain as the sum of its constituent parts does not fully characterize its true worth.

Associate with each edge $e = (v_i, v_j)$ in the graph G a value $q_e \in [0, 1]$ representing the probability that, if algorithmically matched, the patient of v_j would successfully receive a kidney from v_i's donor. We will refer to q_e as the *success probability* of the edge, and $1 - q_e$ as the *failure probability* of the edge. Using this notion of failure probability, we can define the expected (failure-discounted) utility of chains and cycles.

Discounted utility of a cycle

For any transplant in a k-cycle to execute, each of the k transplants in that cycle must execute. Put another way, if even one algorithmically matched transplant fails, the entire cycle fails. Thus, for a k-cycle c, define the *discounted utility* $u(c)$ of that cycle to be:

$$u(c) = \left[\sum_{e \in c} w_e\right] \cdot \left[\prod_{e \in c} q_e\right]$$

That is, the utility of a cycle is the product of the sum of its constituent weights and the probability of the cycle executing. The simplicity of this calculation is driven by the required atomicity of cycle execution—a property that is not present when considering chains.

Discounted utility of a chain

While cycles must execute entirely or not at all, chains can execute incrementally. For example, a 3-chain $c = (a, v_1, v_2, v_3)$ starting at altruist a might result in one of four numbers of transplants:

- No transplants, if the edge (a, v_1) fails
- A single transplant, if (a, v_1) succeeds but (v_1, v_2) fails
- Two transplants, (a, v_1) and (v_1, v_2) succeed but (v_2, v_3) fails
- Three transplants, if all edges in the chain represent successful transplants. (In this case, the donor at v_3 typically donates to the deceased donor waiting list, or stays in the pool as a bridge donor. Whether or not this additional transplant is included in the optimization process is decided by each individual kidney exchange program.)

In general, for a k-chain $c = (v_0, v_1, \ldots, v_k)$, where v_0 is an altruist, there are k possible matches (and the final match to, e.g., a deceased donor waiting list candidate). Let q_i be the probability of edge $e_i = (v_i, v_i + 1)$ leading to a successful transplant. Here, we assume $w_e = 1$ for ease of exposition; in Section 5.5, we show that relaxing this assumption does not complicate matters.

Then, the expected utility $u(c)$ of the k-chain c is:

$$u(c) = \left[\sum_{i=1}^{k-1} (1 - q_i) i \prod_{j=0}^{i-1} q_j\right] + \left[k \prod_{i=0}^{k-1} q_i\right]$$

The first portion above calculates the sum of expected utilities for the chain executing exactly $i = \{1, \ldots, k-1\}$ steps and then failing on the $(i+1)^{th}$ step. The second portion is the utility gained if the chain executes completely.

Discounted utility of a matching

The value of an individual cycle or chain hinges on the interdependencies between each specific patient and potential donor, as was formalized above. However, two cycles or chains in the same matching M fail or succeed independently. Thus, define the discounted utility of a matching M to be:

$$u(M) = \sum_{c \in M} u(c)$$

That is, the expected number of transplants resulting from a matching M is the sum of the expected number of transplants from each of its constituent cycles and chains.

For a (possibly weighted) compatibility graph $G = (V, E)$, let $\mathcal{M}$ represent the set of all legal matchings induced by G. Then, given success probabilities $q_e, \forall e \in E$, the *discounted clearing problem* is to find M^* such that:

$$M^* = \arg\max_{M \in \mathcal{M}} u(M)$$

The distinction between M^* and any maximum (non-discounted) weighted matching M' is important, as we show in the rest of this chapter—theoretically in Section 5.3, on real data from the fielded UNOS kidney exchange in Section 5.4, and on simulated data in Sections 5.7 and 5.8.

5.3 Maximum cardinality matching is far from maximizing the expected number of transplants

In this section, we prove a result regarding the (in)efficacy of maximum cardinality matching in kidney exchange, when the probability of a match failure is taken into account. We show that in pools containing equally sensitized patient-donor pairs (and not necessarily equally sensitized altruistic donors), with high probability there exists a discounted matching that has linearly higher utility than all maximum cardinality matchings. This theoretical result motivates the rest of the chapter; since current fielded kidney exchanges perform maximum cardinality or maximum weighted matching, many potential transplants may be left on the table as a consequence of not considering match failures.

5.3.1 Random graph model of sensitization

We work with (a special case of) the model of Ashlagi et al. [23] (§4.2), which is an adaptation of the standard dense theoretical kidney exchange model (on which we built in, e.g., Chapter 2) to pools with highly and non-highly sensitized patient-donor pairs.

The model works with random compatibility graphs with $n + t(n)$ vertices, pertaining to n incompatible patient-donor pairs (denoted by the set P), and $t(n)$ altruistic donors (denoted by the set A) respectively. Edges between vertices represent not just blood-type compatibility, but also immunological compatibility—the *sensitization* of the patient. Given a blood type-compatible donor, let p denote the probability that an edge exists between a patient and that donor.

Given uniform sensitization p for each of the n patients in the pool, random graphs from this model are equivalent to those of Erdős and Rényi (1960) with parameters n and p. In Section 5.3.2, we use techniques from random graph analysis to prove that maximum cardinality matching in highly sensitized pools (with altruists) does not optimize for expected number of actual transplants.

5.3.2 Maximum cardinality matching in highly sensitized pools

Let $G(n, t(n), p)$ be a random graph with n patient-donor pairs, $t(n)$ altruistic donors, and probability $p = \Theta(1/n)$ of incoming edges. Such a p represents highly-sensitized patients. Let q be the probability of transplant success that we introduced, such that q is constant for each edge e. Note that for a chain of length k, the probability that $t < k$ matches execute is $q^t(1-q)$, and the probability that k matches execute is q^k. There is no chain cap (although we could impose one, which depends on q). Given a matching M, let $u_q(M)$ be its expected utility in our model, i.e., expected number of successful transplants. Denote the set of altruistic donors by A, and denote the vertex pairs by P.

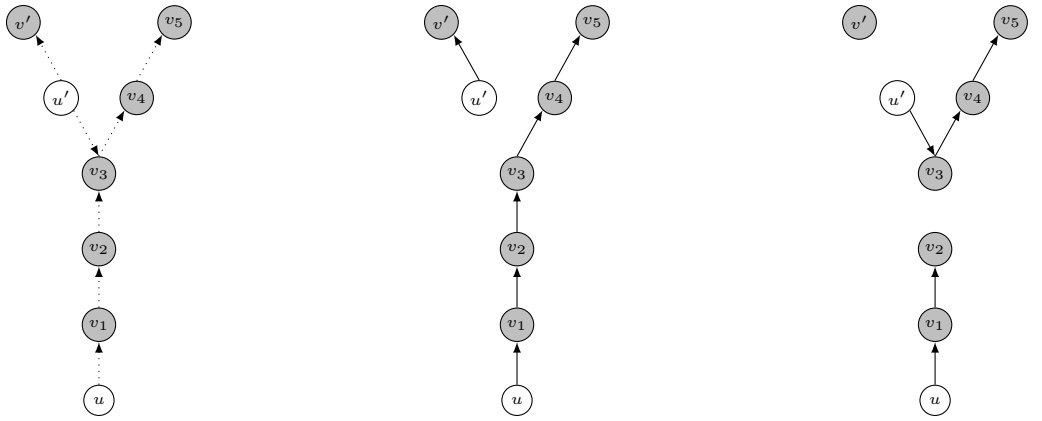
The proof of the following theorem builds on techniques used in the proof of Theorem 5.4 of [23], but also requires several new ideas.

Theorem 13. *For every constants $q \in (0,1)$ and $\alpha, \beta > 0$, given a large random graph $G(n, \alpha n, \beta/n)$, with high probability there exists a matching M' such that for every maximum cardinality matching M,*

$$u_q(M') \geq u_q(M) + \Omega(n).$$

Proof. We consider subgraphs that we call *Y-gadgets*, with the following structure. A Y-gadget contains a path $(u, v_1, \ldots, v_k)$ such that $u \in A$ and $v_i \in P$ for $i = 1, \ldots, k$ for a large enough constant k, to be chosen later. Furthermore, there is another altruistic donor $u' \in A$ with two outgoing edges, (u', v_3) and (u', v') for some $v' \in P$. Finally, the edges described above are the *only* edges incident on the vertices of the Y-gadget. See Figure 5.1a for an illustration.

We first claim that it is sufficient to demonstrate that with high probability the graph $G(n, \alpha n,$ $beta/n)$ contains cn Y-gadgets, for some constant $c > 0$. Indeed, because each Y-gadget is disconnected from the rest of the graph, a maximum cardinality matching M must match all the vertices of the Y-gadget, via a k-chain and a 1-chain. Let M_Y be the restriction of M to the Y-gadget (see



(a) A Y gadget. **(b)** The maximum cardinality matching M_Y. **(c)** The matching M'_Y.

Figure 5.1: Illustration of a Y-gadget with $k = 5$. The vertices of A are white and the vertices of P are gray. Clearly $|M_Y| > |M'_Y|$, but (using Equation 5.1) $u_q(M'_Y) - u_q(M_Y) > q^2 - 6q^4$; this difference is positive for $q < 0.41$, which is a realistic value.

Figure 5.1b). It holds that

$$u_q(M_Y) = (1-q)\sum_{i=1}^{k-1} iq^i + kq^k + q.$$

We next construct a matching M'_Y for the Y-gadget, via two chains: (u, v_1, v_2) and $(u', v_3, \ldots, v_k)$, i.e., vertex v' remains unmatched (see Figure 5.1c). We obtain

$$u_q(M'_Y) = (1-q)\sum_{i=1}^{k-3} iq^i + (k-2)q^{k-2} + q(1-q) + 2q^2.$$

Therefore,

$$u_q(M'_Y) - u_q(M_Y) = q^2 + (k-2)q^{k-1} - (k-1)(1-q)q^{k-1} - kq^k > q^2 - (k+1)q^{k-1}. \quad (5.1)$$

Clearly if k is a sufficiently large constant, $q^2/2 > (k+1)q^{k-1}$, and hence the right hand side of Equation (5.1) is at least $q^2/2$, which is a constant. By applying this argument to each of the cn Y-gadgets we obtain a matching M' such that $u_q(M') - u_q(M) > (q^2/2)cn = \Omega(n)$.

It remains to establish the existence of $\Omega(n)$ Y-gadgets. Consider a random undirected graph with $n+\alpha n$ vertices. The edge probabilities are $p = 2(\beta/n)(1-\beta/n)+(\beta/n)^2$, i.e., the probability of at least one edge existing between a pair of vertices in P. A standard result on random graphs (see, e.g., [120]) states that for every graph X of constant size, with high probability we can find $\Omega(n)$ subgraphs that are *isomorphic* to X and *isolated* from the rest of the graph. In particular,

with high probability our random graph has $\Omega(n)$ subgraphs that are isomorphic to the undirected, unlabeled version of a Y-gadget.

There are two independent issues we need to address. First, these subgraphs are unlabeled, i.e., we do not know which vertices are in A, and which are in P. Second, the graph is undirected, and may have some illegal edges between pairs of vertices in A. We presently address the first issue. We randomly label the vertices as A or P, keeping in mind that ultimately it must hold that $|P| = n$ and $|A| = \alpha n$. Assume without loss of generality that $\alpha \leq 1$. Consider an arbitrary vertex in one of the special subgraphs. This vertex is in P with probability $1/(1+\alpha)$, and in A with probability $\alpha/(1+\alpha)$. The label of the second vertex depends on the first. For example, if the first is in P then the probabilities are $(1-1/n)/(1-1/n+\alpha)$ for P and $\alpha/(1-1/n+\alpha)$ for A.

We sequentially label the vertices of $\min\{cn, (\alpha n)/(10k)\}$ gadgets, where cn is the number of Y-gadgets, taking into account the previous labels we observed. (Note that we are labeling a linear number of Y-gadgets, since k is constant.) Because we observed far fewer than $\alpha n/2$ labels, in each trial the probability of each of the two labels, conditioned on the previous labels, is at least $(\alpha/2)/(1+\alpha/2)$, which is a constant. This lower bound allows us to treat the labels as independent Bernoulli trials. Thus, the probability that a gadget has exactly the right labels (two A labels in the correct places, and P labels everywhere else) is at least $r = ((\alpha/2)/(1+\alpha/2))^{k+3}$, which is a constant. The expected number of correctly labeled gadgets is therefore at least $r \cdot \min\left\{cn, \frac{\alpha n}{10k}\right\}$, i.e., $\Omega(n)$. Using Chernoff's inequality, with high probability we can find $\Omega(n)$ correctly labeled gadgets.

We next address the second issue, that is, the directions of the edges. For each of the $\Omega(n)$ correctly labeled gadgets, each undirected edge corresponds to a directed edge in one of the two direction with probability

$$\frac{\frac{\beta}{n}\left(1-\frac{\beta}{n}\right)}{2\frac{\beta}{n}\left(1-\frac{\beta}{n}\right)+\left(\frac{\beta}{n}\right)^2} \approx \frac{1}{2},$$

and corresponds to edges in both directions with the complement (small) probability. The probability that each edge in a Y-gadget corresponds to a single edge in the correct direction is therefore constant, and using similar arguments as above, with high probability a constant fraction of the correctly labeled gadgets will have correctly oriented edges.

Finally, note that the labels of the vertices and the directions of the edges in each of the initially unlabeled, undirected Y-gadgets are independently assigned. Given one of these initial (linearly many) Y-gadgets, we have shown that the probability of that Y-gadget being labeled correctly is a constant. Similarly, we have shown that the probability of that Y-gadget having all edges in exactly the right orientations is a constant. Thus, since these events are independent, the probability that both events occur is a constant, and we have a constant fraction of the linearly many initial Y-gadgets with the correct orientation of edges and labels of nodes.

A possible concern at this point is that there are no edges between pairs of vertices in A, and the probability of edges (u, v) where $u \in A$ and $v \in P$ is smaller than p. We first note that, since we are looking at a denser graph, isolation is harder to achieve. Moreover, since a Y-gadget has no adjacent vertices in A, we discard any such Y-gadgets, so the increased probability of edges between such pairs does not help us. Finally, for pairs (u, v) where $u \in A$ and $v \in P$, the

probability of an edge (u, v) existing is *equal* to the probability of an edge existing in the undirected graph *and* the edge being in the right direction; Y-gadgets where the edge is in the wrong direction are discarded. □

Importantly, while the proof of Theorem 13 only explicitly discusses chains (in the construction of the Y-gadgets), the optimal matching also contains cycles—they are just not the driving force behind this result. In the next section, we provide experimental validation of this theoretical result using real data from the UNOS nationwide kidney exchange.

5.4 Experiences from, and experiments on, the UNOS kidney exchange

Over the past decade, fielded kidney exchanges have begun appearing in the United States. This section draws on data from the United Network for Organ Sharing (UNOS) exchange through July 2015, except when explicitly written otherwise. At that point, it matched on a biweekly basis, and interacts nationwide with 143 transplant centers. In this section, we present experimental results comparing discounted and non-discounted matching on real data from this exchange, using multiple estimated distributions over edge failure probabilities.

5.4.1 Estimating edge failure probabilities

The UNOS kidney exchange computes a maximum weighted matching at each clearing. The function used to assign weights to edges was determined by a committee of medical professionals, and takes into account such factors as donor and patient location, health, and CPRA score. (We will discuss new approaches to setting these weights in Chapters 6 and 8.) We have access to this data, and use it in our experiments.

However, medical knowledge is incomplete; as such, we cannot determine the exact probability q that a potential transplant will succeed. For our experiments, we use multiple distributions of edge failure probabilities.

First, we draw from all the data from the match runs conducted in the UNOS exchange to date. Figure 5.2 displays success and failure results for recommended matches from the UNOS kidney exchange for matches between October 27, 2010 and November 12, 2012.[1] Approximately 7% of matches resulted in a transplant, while approximately 93% failed. Of the 93% that failed:

- 53% (49% of all matches) were not the reason for failure. The cycle or chain in which the potential transplant was involved failed entirely (in the case of cycles) or before the patient's turn (in the case of chains).
- 47% (44% of all matches) were the reason for failure.

[1]The aggregate match data from which we infer crossmatch failure probabilities is available in a report from the [131] and summarized by [143]. Updated aggregate data is now available in a report from the [132]; this most recent data was not incorporated into our experiments, but is very similar to that which was.

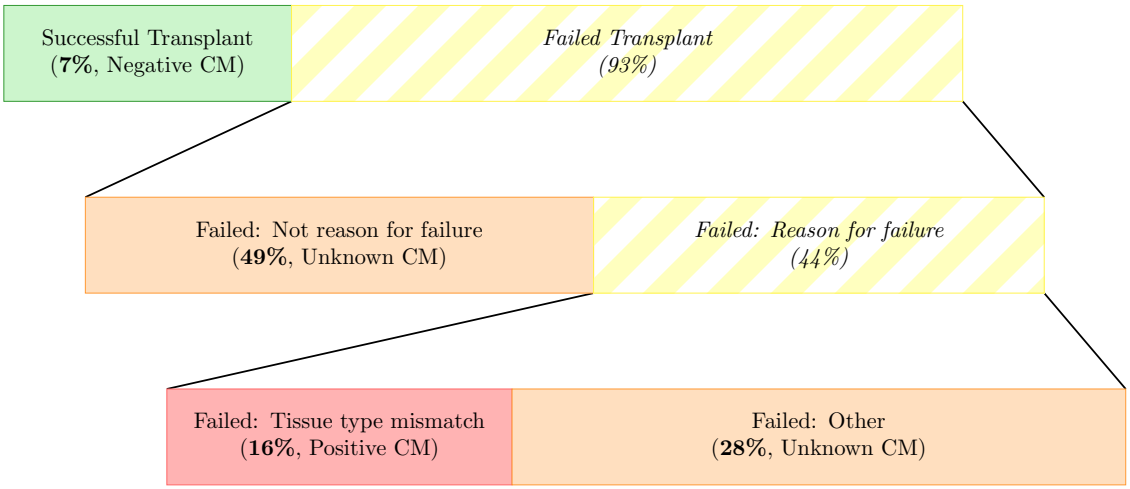


Figure 5.2: Determining the probability of a match failing is difficult because many potential patient-donor pairs are not crossmatched. Of the aggregate UNOS data, we are only sure that the 7% who successfully received a transplant and the 16% who explicitly failed due to a positive crossmatch were tested.

- 36% of these (about 16% of the total) failed due to a *positive crossmatch*, signifying blood-type incompatibility (beyond the ABO model).
- 64% failed due to a variety of other reasons, as discussed below.

Triggering a cycle or chain failure can occur for a variety of reasons, including:

- Receiving a transplant from the deceased donor waiting list
- Receiving a transplant from another exchange
- Patient or donor becoming too ill for surgery or expiring
- An altruistic patient "running out of patience" and donating elsewhere, or not at all
- A donor in a chain reneging (i.e., backing out after his patient received a kidney)
- Pregnancy or sickness changing a patient or donor's antigen incompatibilities

In these cases, a patient and potential donor may or may not have received a crossmatch test. In fact, the only sureties regarding crossmatches to be derived from the data above are that 7% crossmatched negative (those who received transplants) and 16% crossmatched positive. Thus, roughly $7/(16+7) \approx 30\%$ of these crossmatches came back negative. We use this value for our first set of simulations, setting the probability of a crossmatch failing to be a constant 70%. This 70% expected failure is optimistic (i.e., too low) in that it ignores the myriad other reasons for match failures. UNOS currently performs batch myopic matches, so—for these simulations—we only simulated crossmatch failures. We take additional failure reasons into account in Sections 5.8 and 5.9.

Second, in the UNOS exchange and in others (see., e.g., [23]), patients tend to have either very high or very low sensitization, i.e., there is a very low or very high probability that their blood will pass a crossmatch test with a random organ. For *highly-sensitized* patients, finding a kidney

is very difficult. Drawing from this and the 70% failure rate derived above, our second set of experiments samples randomly from a bimodal distribution: 25% of edges have a low failure rate $(1 - q_L) \in U[0.0, 0.2]$, while 75% have a high failure rate $(1 - q_H) \in U[0.8, 1.0]$, such that the overall expected failure rate is 70%. Third, we systematically vary the variance of the underlying failure probability distribution and explore its effect on the behavior of both matching methods.

5.4.2 UNOS results: Discounted matching is better in practice

We now simulate probabilistic matching on real data from UNOS. We performed simulations using both the constant 70% failure rate and the bimodal failure rate. On the former, we can compute an exact expected value for the discounted matching on each real UNOS matching. On the latter, we simulated failure probabilities at least 100 times for each UNOS match run.

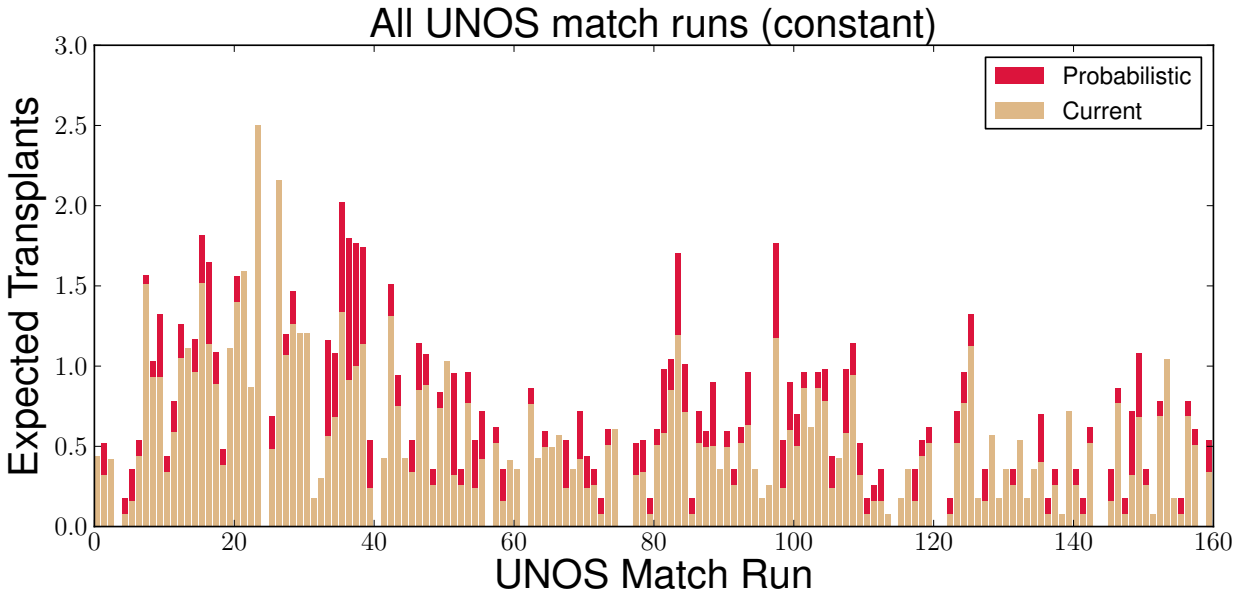


Figure 5.3: Comparison of the expected number of transplants resulting from the maximum weighted matching and discounted weighted matching methods, on 161 UNOS match runs between October 2010 and November 2014, with a constant edge success probability.

Figures 5.3 and 5.4 show that, in both cases, taking failure probabilities into account results in significantly more expected transplants.In the constant probability case, discounted matching yields many more matches than (or in some cases the same number as) the status quo of maximum weighted matching. (In cases where the expected utility of both matching methods was equal, the matchings with equivalent compositions (i.e., same number of 2-cycles, 3-cycles, and k-chains) were returned by both solvers.) The discounted matching performed better when the maximum weighted matching included long chains, a frequent phenomenon in the UNOS pool (and other fielded exchange pools in the US and abroad), as discussed by [74], [23], and [100].

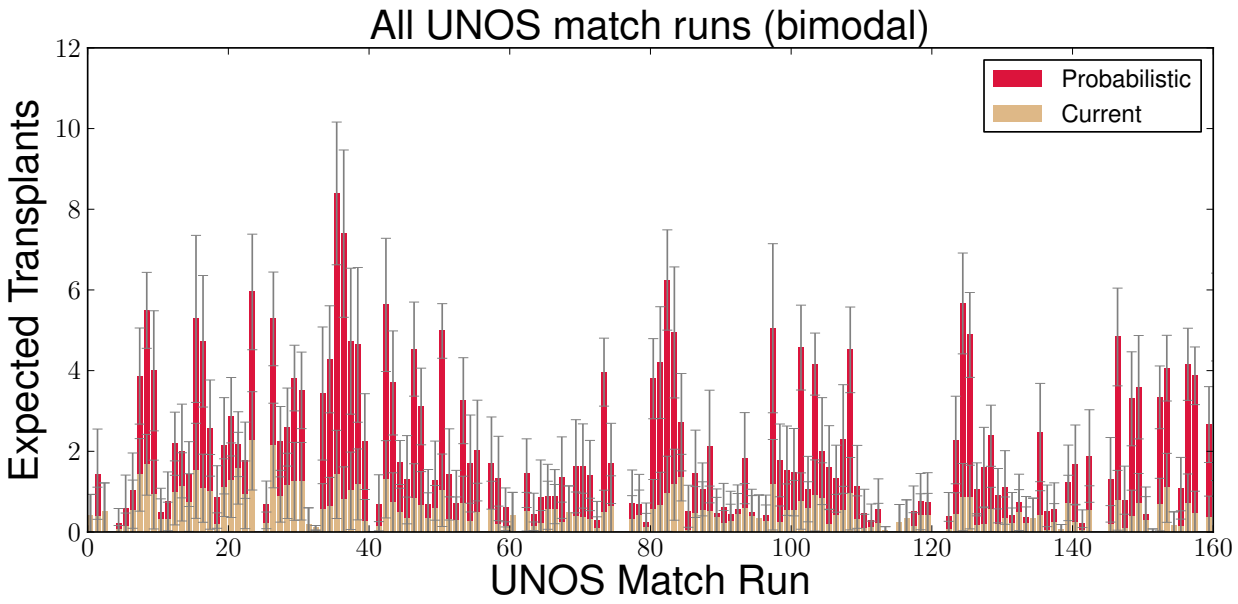


Figure 5.4: Comparison of the expected number of transplants resulting from the maximum weighted matching and discounted weighted matching methods, on 161 UNOS match runs between October 2010 and November 2014, with bimodal edge success probabilities (some very high, some very low).

In the bimodal case, discounted matching shines, often beating the maximum cardinality matching by a factor between 2 and 5, and again never doing worse in expectation. Here, the discounted matching algorithm is able to pick cycles and chains that contain edges with very high probabilities of success over those with very low probabilities of success.

Table 5.1 gives aggregate match data for both the current UNOS solver and our proposed method on both the constant and bimodal underlying failure rate probability distributions. Across all UNOS match runs using a constant edge failure probability of 0.7, the failure-aware method results in an expected 0.15 more transplants per match run over the maximum weighted matching solver. Using the bimodal distribution, the failure-aware method returns an expected additional 1.38 transplants per match run. Table 5.1 gives the results of both a paired t-test and a Wilcoxon signed-rank test (a non-parameteric version of the t-test); we ran both on the expected number of transplants from the 161 paired deterministic and failure-aware optimal matchings for each of the UNOS match runs to test if their population means were different. Clearly, the gains seen under both failure distributions are statistically significant.

5.4.3 Distributional diversity begets greater gains

Section 5.4.2 showed experimentally that (i) a statistically significant gain in expected matches occurs under the consideration of match failure and (ii) a bimodal underlying failure probability distribution resulted in more of a gain than a constant underlying failure probability distribution. We delve deeper into this insight in this section.

	Current		Ours		t-test		Wilcoxon Signed-rank	
Distribution	Average	St. Dev.	Average	St. Dev	t-statistic	p-value	Siegel's T	p-value
Constant	0.52	0.43	0.67	0.50	10.95	$< 10^{-10}$	0	$< 10^{-10}$
Bimodal	0.51	0.43	1.89	1.79	11.85	$< 10^{-10}$	0	$< 10^{-10}$

Table 5.1: Distributional difference between maximum weighted matching and failure-aware matching on real UNOS data.

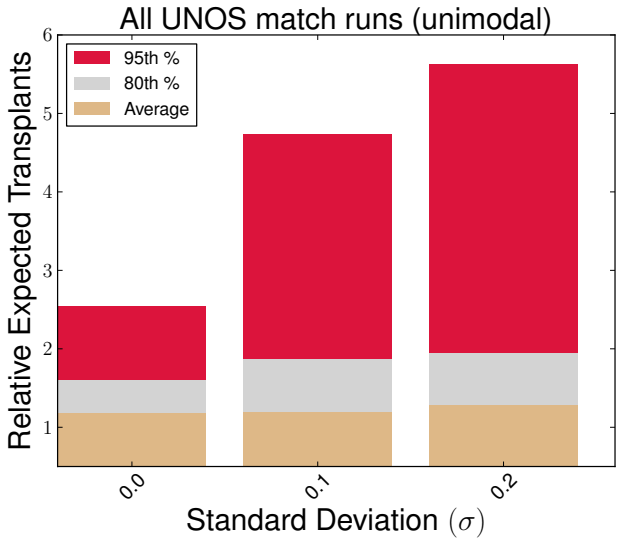


Figure 5.5: Aggregate additional transplants over all UNOS match runs through November 2013, for edge failure probabilities drawn randomly from $\mathcal{N}(\mu = 0.7, \sigma \in \{0.1, 0.2\})$. The leftmost point "$\sigma = 0.0$" represents a constant failure rate of 70%.

We now investigate the effect that higher *variance* in edge failure probabilities has on the overall value of both matching methods. For this section's experiments, we sample from a normal distribution with mean of 0.7 and varying standard deviation. If a sample returns an illegal failure probability p (i.e., $p < 0$ or $p > 1$), we resample from the underlying distribution. In this way, we expand the underlying distribution from a constant 0.7 toward a more uniform randomness.

Figure 5.5 shows the aggregate number of expected transplants (summed over all UNOS match runs through November 2013) for varying levels of variance σ^2, given a standard deviation of σ, in the underlying distribution from which failure probabilities are sampled. For convenience, we label the constant probability of 0.7 case as "$\sigma = 0.0$". Positive crossmatches are simulated based on an edge's sampled probability of failure.

In the constant probability case, failure-aware matching results in an average expected 18.4% increase in expected transplants. As the standard deviation of the underlying distribution increases, so too does this expected boost: from 18.8% to 28.5% respectively, for $\sigma = 0.10$ and $\sigma = 0.20$, respectively. An increase in variance also results in the maximum cardinality matching method frequently missing the highest utility match by a large margin. For instance, the 80^{th} and 95^{th}

percentiles increase from an additional 59.8% and 154.2% in the constant probability case to 94.6% and 462.9% when $\sigma = 0.20$. Higher variance results in more opportunities for the maximum cardinality matching to contain many matches with an extremely low probability of execution (e.g., a 3-cycle with edges that are likely to fail instead of a smaller 2-cycle with more reliable edges).

Next, in Section 5.5, we construct a solver that can optimally solve capable of clearing large exchanges than those currently available at UNOS.

5.5 Building a scalable solver to clear failure-aware exchanges

Current kidney exchange pools are small, containing at most a few hundred patients at a time. However, as kidney exchange gains traction, these pools will grow. As discussed by Abraham et al. [6], the estimated steady-state size of a US nationwide kidney exchange is $10{,}000$ patients; more recent estimates vary, but the prospect of international exchange [10] and multi-organ exchange [72] will result in large pools.

Clearing pools of this size is a computational challenge, as we motived in Chapter 3. Abraham et al. [6] showed that the undiscounted clearing problem is NP-hard. Since the undiscounted clearing problem is a special case of the discounted clearing problem—that is, it is the discounted clearing problem with constant success probability $q = 1.0$—it follows that the discounted clearing problem is also NP-hard.

Proposition 1. *The discounted clearing problem is NP-hard.*

In the current UNOS solver, chains are incorporated by adding from the end of each potential chain a "dummy" edge of weight 0 to every vertex that represents an altruist. Chains are generated in the same fashion as cycles, and look identical to cycles to the optimization algorithm—with one caveat. Recall that chains need not be executed atomically, and thus, in practice, the cycle cap of 3 is not applicable to chains. Due to the removal of this length restriction, this approach does not scale even remotely to the nationwide level—failing in exchanges of sizes as low as 200 in the undiscounted case (as shown in Chapter 2). We showed in Chapter 3 that this scalability issue can be solved via compact models like PIEF and HPIEF, or non-compact-but-amenable-to-faster-pricing models like PICEF, for the deterministic case. Indeed, we also showed how to extend PICEF to a failure-aware model with *uniform* failure probabilities. We relax that assumption to the fully-general case now.

In this section, we augment the current UNOS solver to solve the discounted clearing problem on exchanges with edge failure probabilities. We first show that a powerful tool used in the current solver—the technique used to upper bound the objective value—is no longer useful. We show how to adapt the current solver's lower bounding technique to our model. We then significantly improve the core of the solver, which performs *column generation*, to only consider cycles and chains that are useful to the optimal discounted matching, and provide failure-aware heuristics for speeding up the column generation process.

5.5.1 Why we cannot use the current UNOS solver

In integer programming, a tree search that branches on each integral decision variable is used to search for an optimal solution. At each node, upper and lower bounds are computed to help prune subtrees and speed up the overall search. In practice, these bounding techniques are critical to proving optimality without exhaustively searching the space of all assignments.

Computing a good upper bound is hard

The current kidney exchange solver uses the cycle cover problem with no length cap as a heuristic upper bound. This unrestricted clearing problem is solvable in polynomial time by encoding the pool into a weighted bipartite graph and computing the maximum weighted perfect matching (see reduction by [6]). This is useful in practice because the unrestricted bound often matches the restricted (e.g., $L \leq 3$) optimal objective value. Unfortunately, for the discounted version of this problem, Proposition 2 shows that computing this bound is NP-hard.

Proposition 2. *The unrestricted discounted maximum cycle cover problem is NP-hard.*

Proof. We build on the proof of Theorem 1 from Abraham et al. [6], which shows that deciding if G admits a perfect cycle cover containing cycles of length at most 3 is NP-complete. They reduce from 3D-Matching. All the cycles in the constructed widgets in their proof are of length at least 3. Due to discounting, a perfect cover which uses only 3-cycles has higher utility than any other cover, since each edge in a 3-cycle is worth more than a vertex in a k-cycle for $k > 3$ due to discounting. The reduction of Abraham et al. [6] has the property that there is a perfect cover with only 3-cycles if and only if there is a 3D-Matching. Determining this is NP-complete, and thus the search problem is NP-hard. □

Driven by this hardness result, our new solver can use one of two looser upper bounds. If chains are to be executed in an "all-or-nothing" fashion—that is, either they execute in their entirety or are completely cancelled if any edge fails—our solver uses a looser upper bound, solving the unrestricted clearing problem on a graph $G' = (V, E')$ such that $w'_e = w_e q_e$, for each $e \in E$. If the execution policy is incremental—as it is in practice, and in our experimental results—then an even weaker upper bound ignoring q_e is used.

Computing a good lower bound is not hard

The current UNOS solver uses the 2-cycle maximum matching problem (which is equivalent to the undiscounted clearing problem for $L = 2$) as a primal heuristic, or lower bound. The new solver uses the *discounted* version of the 2-cycle maximum matching problem as a primal heuristic during the branch-and-price search. Solving this problem is still in polynomial time, as stated in Proposition 3.

Proposition 3. *The discounted clearing problem with cycle cap $L = 2$ is solvable in polynomial time.*

Proof. Given a directed compatibility graph $G = (V, E)$, construct an undirected graph $G' = (V, E')$ such that an edge exists between two vertices in G' if and only if there exists a two-cycle between those vertices in G. Then, set the weight of every edge $e' = (v_i, v_j)$ in G' to:

$$w_{e'} = q_{(v_i,v_j)} \cdot q_{(v_j,v_i)}(w_{(v_i,v_j)} + w_{(v_j,v_i)})$$

Now find the maximum weighted matching on G', which can be done in polynomial time by Edmond's maximum-matching algorithm [84]. □

Incremental solving of very large IPs

The number of decision variables in the integer program formulation of the clearing problem grows linearly with the number of cycles and chains in the pool; however, as discussed in Section 3.1.2, the number of cycles of length at most L is $O(|P|^L)$, the number of uncapped chains is exponential in $|P|$ if $A \neq \emptyset$, and the number of capped chains of length (here, defined to be the number of constituent edges) at most K is $O(|A||P|^K)$.

As a recap, the current UNOS solver uses an incremental formulation called column generation to bring only some variables into the search tree at each node. The basic idea behind column generation is to start with a reduced model of the problem, and then incrementally bring in variables (and their constraints) until the solution value of this reduced model is provably the solution value of the full (implicitly represented) model. This is done by solving the pricing problem, which associates with each variable a real-valued price such that, if any constraint in the full model for a variable c is violated, then the price of that variable is positive. In our case, the *price* of a cycle or chain c is just the difference between the discounted utility $u(c)$ and the dual value sum of the vertices in that cycle or chain.

When no positive price cycles exist, we have proved optimality at this node in the search tree. Proving this is hard, since the solver might have to consider each cycle and chain. Because we are pricing *both* cycles and chains, we cannot use the fast pricing scheme from Section 3.2 applied to PICEF in Section 3.3.5. We now present a method for "cutting off" a chain after we know its expected utility is too low to improve the reduced problem's objective value.

5.5.2 Iterative generation of only potentially "useful" chains.

Given a k-chain $c = (v_0, v_1, \ldots, v_k)$, with v_0 an altruist, we show a technique for curtailing the generation of additions to c (while maintaining solution optimality). Consider the $(k+1)$-chain $c' = c + \{v_{k+1}\}$. Then the *additional* utility of this chain over c is just:

$$
\begin{aligned}
u(c') - u(c) &= \left(\sum_{i=1}^{k} (1-q_i) i \prod_{j=0}^{i-1} q_j + (k+1) \prod_{i=0}^{k} q_i \right) - \left(\sum_{i=1}^{k-1} (1-q_i) i \prod_{j=0}^{i-1} q_j + k \prod_{i=0}^{k-1} q_i \right) \\
&= (1-q_k) k \prod_{i=0}^{k-1} q_i - k \prod_{i=0}^{k-1} q_i + (k+1) \prod_{i=0}^{k} q_i \\
&= (k+1) \prod_{i=0}^{k} q_i - q_k k \prod_{i=0}^{k-1} q_i = (k+1) \prod_{i=0}^{k} q_i - k \prod_{i=0}^{k} q_i = \prod_{i=0}^{k} q_i
\end{aligned}
$$

That is, the additional utility is just the probability of c' executing perfectly from start to finish (times the weight of the new edge, if $w_k \neq 1$).

Now, assume we are given some maximum success (minimum failure) probability q_{max} of the edges left in the remaining total pool of patients V' (so for $G = (V, E)$, the remaining pool is $V' = V \setminus c$). Then, an upper bound on the additional utility of extending c to an infinitely long chain c^{∞} is just the geometric series:

$$
u(c^{\infty}) - u(c) < \sum_{j=k}^{\infty} \prod_{i=0}^{k-1} q_i \prod_{i=k}^{j} q_{max} = \prod_{i=0}^{k-1} q_i \left(\sum_{j=k}^{\infty} \prod_{i=k}^{j} q_{max} \right)
$$

Since $q_{max} < 1$, this converges to:

$$
u(c^{\infty}) - u(c) =_{k \to \infty} \frac{q_{max}}{1 - q_{max}} \prod_{i=0}^{k-1} q_i \tag{5.2}
$$

An *upper bound* on the expected utility of a (possibly infinite) chain c', extended from some base k-chain $c = (v_0, v_1, \ldots, v_k)$, is given in Equation (5.2) above. We are interested in using this computed value to stop extending c.

Let the dual value of a vertex v be d_v. Furthermore, let d_{min} be the minimum dual value of any vertex in $V' = V - c$. Then a *lower bound* on the "cost" of using this extended chain c' is given by $d_{min} + \sum_{i=0}^{k} d_i$.

By taking the optimistic upper bound on the utility of an infinite extension c' and the lower bound on the "cost" of using c', a criterion for c' not being useful is:

$$
\left(\frac{q_{max}}{1 - q_{max}} \prod_{i=0}^{k-1} q_i \right) + u(c) + \ell - \left(d_{min} + \sum_{i=0}^{k} d_i \right) \leq 0 \tag{5.3}
$$

Here, ℓ is the utility derived from the final donor in a chain donating his or her kidney to the deceased donor waitlist. This is set by each individual kidney exchange.

Note that the sum of any finite subsequence of the infinite geometric series is less than the sum of the infinite series. Then, the first segment of Equation 5.3 can be only *lower* for any finite extension of c. Thus, if the inequality holds for the infinite extension, it must also hold for the finite extension.

Proposition 4. *Given a k-chain c, if the infinite extension c^∞ is not promising (i.e., Equation 5.3 holds), then no finite extension is promising, either.*

We use Proposition 4 to stop generating extensions of chains during our solver's iterative chain (column) generation routine. We incrementally maintain the expected utility of the chain $u(c)$ and the sum of the dual values of vertices in that chain, and compute the infinite series' convergence of the infinite chain whenever an extension is considered. If Equation 5.3 holds, from Proposition 4, we know no finite (or infinite) extension of c can have positive price, and the solver cuts off generating additions to c.

5.5.3 Heuristics for generating positive price chains and cycles.

During the column generation process, the optimizer iteratively brings positive price cycles and chains into a reduced linear program (LP). Once no cycles or chains outside the reduced LP have positive price, where the price of a cycle/chain c is defined to be $u(c) - \sum_{v \in c} d_v$, we can determine optimality from the reduced LP for the full LP.

In practice, the order in which positive price cycles and chains are brought into the reduced problem drives solution time. One approach is to try to generate those cycles and chains with lowest price. In our solver, we heuristically order the edges from which we start cycle or chain generation toward this end.

Ordering the cycle generation

For cycles, where v is a patient-donor vertex and v' is the vertex in v's outgoing neighbors with maximum discounted edge weight, we sort in descending order of ν:

$$\nu_v = \bar{q}_v^{in} q_{(v,v')} w_{(v,v')} - d_v$$

Here, $\bar{q}_v^{in}$ is the average success probability of all incoming edges to v. Note that, for each vertex v, the $\bar{q}_v^{in} q_{(v,v')} w_{(v,v')}$ term can be computed exactly once (at cost $O(|V|^2)$), since these values do not change. Then, at each iteration of column generation, we perform an $O(|V| \log |V|)$ sort on the difference between this term and the current dual values.

Proposition 5. *For any non-altruist v and next step v', such that $(q_{(v,v')} w_{(v,v')} - d_v) \leq 0$, we need not initiate cycle generation from v (which still guarantees all cycles are generated).*

Proof. A cycle c involves at least two vertices, including v. If v has a non-positive dual-discounted weight, then at least one other vertex v' in the cycle must have positive dual-discounted weight. If not, the cycle will have non-positive price and will not be considered in the column generation. Starting a search from v' will generate c. □

Ordering the chain generation

For chains, where a is an altruist and v is the vertex corresponding to the initial edge from that altruist, we sort in descending order of ν:

	CPLEX		Ours		Ours without chain curtailing	
$\|P\|$	Solved	Time (solved)	Solved	Time (solved)	Solved	Time (solved)
10	127 / 128	0.044	128 / 128	0.027	128 / 128	0.052
25	125 / 128	0.045	128 / 128	0.023	128 / 128	0.049
50	105 / 128	0.123	128 / 128	0.046	125 / 128	0.057
75	91 / 128	0.180	126 / 128	0.072	123 / 128	0.066
100	1 / 128	1.406	121 / 128	0.075	121 / 128	0.071
150	0 / 128	–	114 / 128	0.078	95 / 128	0.098
200	0 / 128	–	113 / 128	0.135	76 / 128	0.096
250	0 / 128	–	94 / 128	0.090	48 / 128	0.133
500	0 / 128	–	107 / 128	0.264	1 / 128	0.632
700	0 / 128	–	115 / 128	1.071	0 / 128	–
900	0 / 128	–	38 / 128	2.789	0 / 128	–
1000	0 / 128	–	0 / 128	–	0 / 128	–

Table 5.2: Scaling results for our method versus CPLEX, timeout of 3600 seconds.

$$\nu_{a,v} = q_{(a,v)} w_{(a,v)} - d_a$$

The intuition here is that chains with a high utility outgoing edge (at low cost, from d_a) are more likely to be included in the final solution than those with low initial utilities. Note that we must consider all first hops out of all altruists, including those such that $\nu_{a,v} \leq 0$. Due to this, each iteration of column generation requires an $O(|A||P|\log(|A||P|))$ sort. With $|A|$ small, as in the UNOS exchange, this is an allowable cost.

5.6 Scalability experiments

In this section, we test the ability of our new solver on kidney exchange compatibility graphs that are larger than current fielded kidney exchange pools, with an eye toward the future where kidney exchanges will be larger. We use data generated by the dense kidney exchange instance generator by Saidman et al. [191], augmented to include altruistic donors. These graphs are significantly denser than current kidney exchange pools, as discussed in Chapter 2, but may more accurately represent future large exchange pools. We test in the static (that is, myopic batch matching) setting here; in the next section, we expand to dynamic matching (albeit with a myopic optimizer, which will be relaxed in Chapter 6). For the experiments in this section, we assume a constant failure probability of 0.7 for each donor-patient edge.

We compare our novel solver against IBM ILOG CPLEX 12.2 [117], a recent version of a state-of-the-art integer linear programming solver. Since CPLEX does not use branch-and-price, it must solve the full integer program (with one decision variable per possible cycle and chain).

Table 5.2 shows runtime and completion results for both solvers on graphs of varying size. Each graph has $|P|$ patient-donor pairs and $|A| = 0.1|P|$ altruistic donors. For example, a row

labeled $|P| = 50$ corresponds to a graph with 50 patient-donor pairs and 5 altruists. We generated 128 such graphs for each value of $|P|$. Each solver was allocated 8GB of RAM and 1 hour of solution time on Blacklight, a large cc-NUMA shared-memory supercomputer at the Pittsburgh Supercomputing Center. (Blacklight was used solely to parallelize multiple runs for experimental results; our solver does not require any specialized hardware. In fact, the current version of our solver that runs the weekly matches at UNOS runs on commodity hardware.) CPLEX was unable to solve instances of size 100 (except once) in under an hour, while our solver was able to solve (at least some) instances of size 900.

To test how much speed was added by each of the improvements in this chapter to the current UNOS solver, we deactivated the cycle and chain generation ordering heuristics (§5.5.3), as well as the solver's ability to cut off chain generation after the initial portion of a chain has been proven not to be in an optimal match (§5.5.2). Interestingly, removing the cycle and chain ordering heuristics did not noticeably affect the runtime or number of instances solved by our solver. Their low impact on performance is caused by the weak upper bounding performed during the IP solve; since the bounding is weak, often optimality must be proved by considering all (discounted, possibly "good") chains and cycles, as opposed to being proved via bounding in the search tree. We believe these ordering heuristics, or ones like them, will hold greater merit when better bounding techniques are developed in the future. However, turning off the solver's ability to reason about the maximum additional discounted utility of a chain did significantly affect overall runtime and number of instances solved; in fact, without this technique, only a single instance with 500 patient-donor pairs finished within the one hour time limit.

Table 5.2 also lists runtime results for those instances that did complete. When a solver was able to solve an instance within an hour, the solution time was typically quite low. This is a function of the upper and lower bounds becoming tight early on in the search tree. Overall, our method of incrementally generating cycles and chains results in dramatically increased completion percentages and lower runtimes than CPLEX.

5.7 Instantaneous rematching in the static model

Fielded kidney exchanges operate in the *static* setting, first performing a batch matching (typically at a defined periodicity), then testing edges in that match, and finally performing successfully tested transplants and placing patient-donor pairs with failed edges back in the pool. In this section, we explore the effect of this policy in the static setting (which leads to Section 5.8 and the formulation of a general dynamic model of kidney exchange).

A myopic optimization (using the failure-aware method described in this chapter) is performed without taking into account the possibility of rematching. The leftover pool—together with the positive crossmatches from the last match—are then instantaneously rematched. We are interested in the additional expected transplants gained from this second (or third, or more) round of matching. Note that this only makes sense in a model that includes match failures, as the maximum weighted matching in a deterministic setting would leave no matchable vertices in the remaining pool.

Figure 5.6 shows the effect of instantaneous rematching on both the size of the discounted

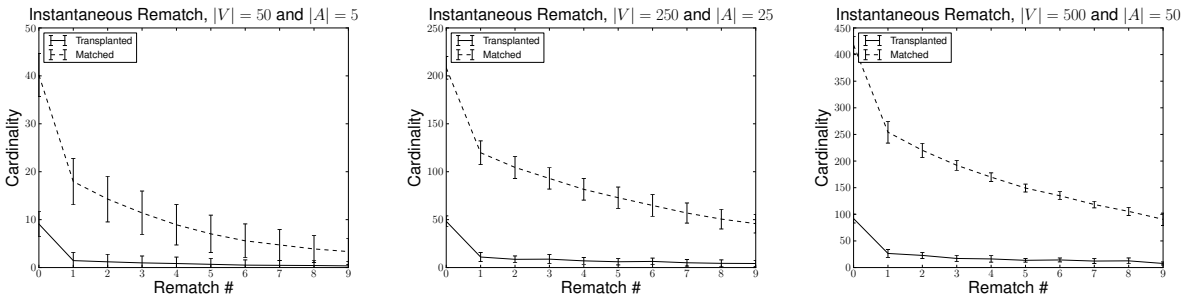


Figure 5.6: Expected cardinality of the algorithmic match (dotted line) and number of transplants (solid line) as the number of simultaneous rematches is increased, for $|P| \in \{50, 250, 500\}$ and $|A| = 0.1|V|$.

$\lvert P\rvert$	1M Avg. (St. Dev.)	10M Avg. (St. Dev.)	Percent Change
50	9.09 (2.56)	15.97 (3.98)	+75.69%
100	18.25 (3.61)	36.95 (6.35)	+102.02%
250	48.23 (5.69)	109.66 (9.84)	+127.27%
500	92.00 (8.28)	235.00 (16.25)	+155.43%

Table 5.3: Expected number of transplants given a single matching ("1M") versus a single matching and nine rematches ("10M").

maximum matching and the expected number of resulting transplants, as the number of allowed rematches is increased from zero (a single batch match) to nine (ten total matches, nine instantaneous rematches). While the size and value of subsequent matches decreases (as expected), the results have a heavy tail; that is, even later rounds of rematching contribute nontrivially to the aggregate expected transplants in large enough pools.

Table 5.3 quantifies the heavy tails shown in Figure 5.6. It compares the expected number of transplants after a single batch matching against the aggregate expected value of a single batch matching and nine instantaneous rematches. Even in a small pool with 50 patient-donor pairs and 5 altruists, multiple rematches result in an expected additional 75.69% transplants. Greater percentage-wise gains are realized for pools of larger size due to the remaining thickness in an exchange, even after multiple vertex and edge removals.

5.8 A model for experimental dynamic kidney exchange

In this section, we explore failure-aware matching in the context of dynamic kidney exchange. Kidney exchange is a naturally dynamic event, with patients, paired donors, and altruists arriving and departing the pool over time. Section 5.4 enumerated some of the reasons we have seen in our experiences with the UNOS nationwide exchange. Formally, a dynamic kidney exchange can be explained by the evolution of its graph—that is, the addition and removal of its vertices and edges.

Table 5.4 formalizes the evolution of a compatibility graph over time. The only vertex and edge

Vertex –	Edge –	Vertex/Edge +
Transplant, this exchange	Matched, positive crossmatch	Normal entrance
Transplant, deceased donor waitlist	Matched, candidate refuses donor	
Transplant, other exchange ("sniped")	Matched, donor refuses candidate	
Death or illness	*Pregnancy, sickness changes HLA**	
Altruist runs out of patience		
Bridge donor reneges		

Table 5.4: Reasons for the arrival and departure of vertices and edges. **We do not consider edge removal due to pregnancy/sickness because there are a variety of ways in which pregnancy and sickness can affect the immune system.*

additions to the graph come in the form of new patients and donors arriving over time. Edges are removed due to, e.g., crossmatch failures or donor refusals. Vertices are removed if the patient or her respective donor must leave the pool, due to reasons ranging from a successful transplantation to patient expiration.

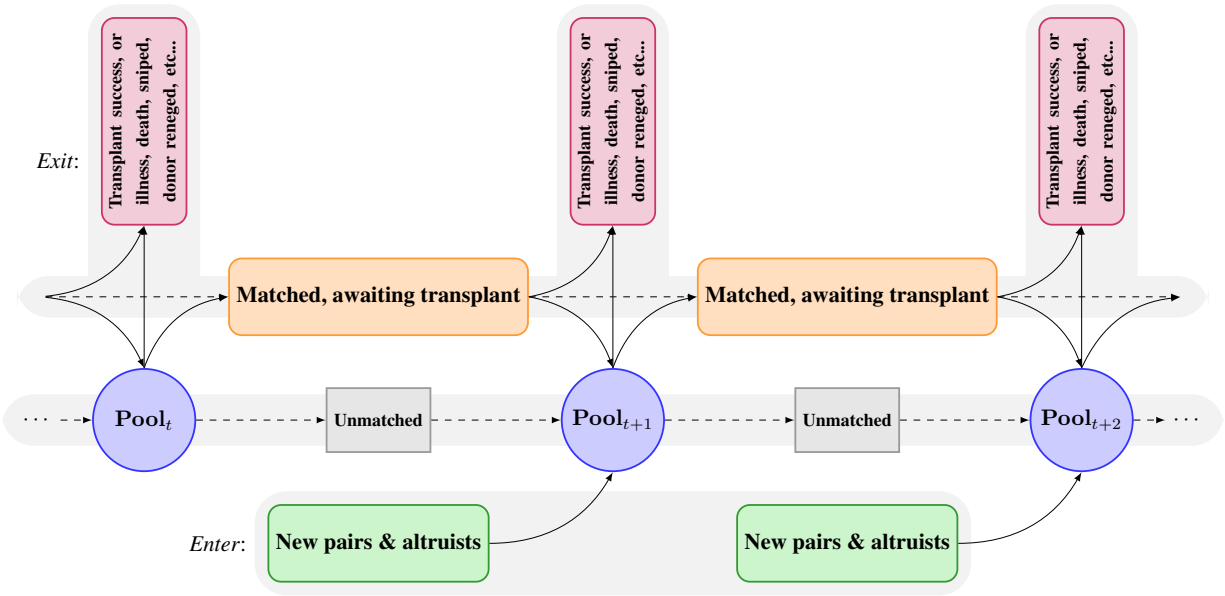


Figure 5.7: The evolution dynamics of a kidney exchange.

Figure 5.7 provides a snapshot of a compatibility graph over three points in time. The pool at time t consists of unmatched patients and donors from time $t-1$, any new pairs and altruists entering the pool, and any vertices who were waiting for a successful match, but whose match failed (due to, e.g., a positive crossmatch). Note that these patients are still formally in the pool, just marked temporarily "inactive" until the status of their pending transplant is known. At each time period t, vertices leave the pool permanently through any of the reasons in the first column of Table 5.4, or are temporarily marked "inactive" through a pending match.

5.8.1 Failure-aware matching in dynamic kidney exchange

We now present experimental results on dynamic kidney exchanges, taking transplant success probabilities into account. We built a simulator that mimics the evolutionary diagram of Figure 5.7, and used parameters learned from our work with UNOS. We vary the number of patient-donor pairs and altruists entering the pool over time, and match on a weekly basis for 24 weeks. We use the bimodal distribution of failure probabilities described in Section 5.4, as it more accurately represents current kidney exchanges. The deceased-donor waitlist donation at the end of a chain is counted in the expected number of transplants.

In our experience with UNOS at the time of this writing, typically the time between a match offer and transplant success or failure is about 8 weeks. Thus, whenever a match is offered in our simulator, involved patients and donors become inactive in the pool, but can still be removed from the match for a variety of reasons ("sniping" by another exchange, patient illness, etc). Of the 610 patients who had ever been listed in the UNOS exchange program when these experiments were run (over a period of 106.7 weeks), 192 left for reasons other than receiving a kidney through UNOS. Thus, for each time period, a vertex has a probability of $1 - e^{(\ln 418/610)/106.7} \approx 0.003536$ chance of leaving (for a non-UNOS transplant reason). As in real kidney exchange, if a cycle fails, or part of a chain fails, then the affected patients and donors are returned to the pool—or is removed permanently, if the reason for failure was that patient-donor pair's exit from the exchange. Results from crossmatches that were done as part of a failed cycle or chain are maintained in the pool; if a crossmatch was negative, then future crossmatches performed on that edge will also be negative. We assume all crossmatches are done simultaneously for cycles and incrementally from the initiating non-directed donor until the first failure for chains.

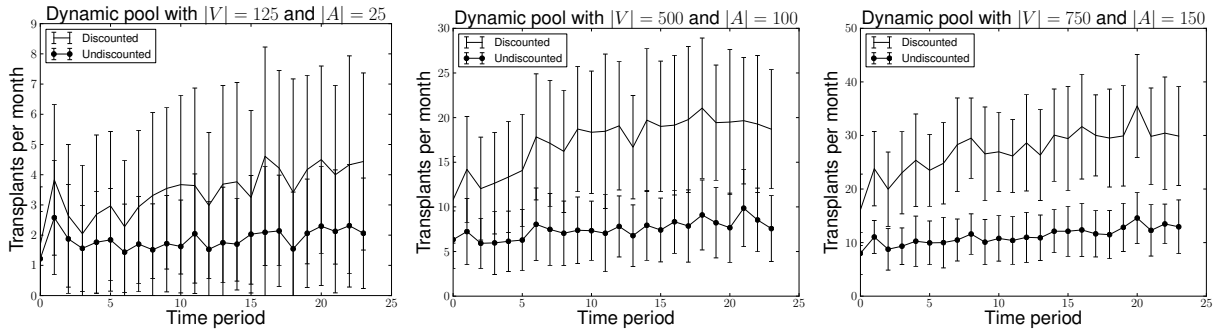


Figure 5.8: Expected number of transplants per week for graphs of different sizes. From left to right, 5 pairs and 1 altruist, 20 pairs and 4 altruists, and 25 pairs and 5 altruists (on expectation) appear every week.

Figure 5.8 shows the number of expected transplants per week on graphs of three different sizes, each generated from the [191] distribution of compatibility graphs. (In the following section, we generate graphs from the UNOS distribution.) In expectation, 5, 20, or 25 pairs and 1, 4, or 5 altruists appear weekly in each of the three graphs. Discounted matching typically results in roughly twice as many expected transplants than maximum cardinality matching. The slight increase in weekly expected matches for both matching techniques is due to the buildup of un-

matched patient-donor pairs and altruists in the pool over time; larger pools typically admit larger matchings.

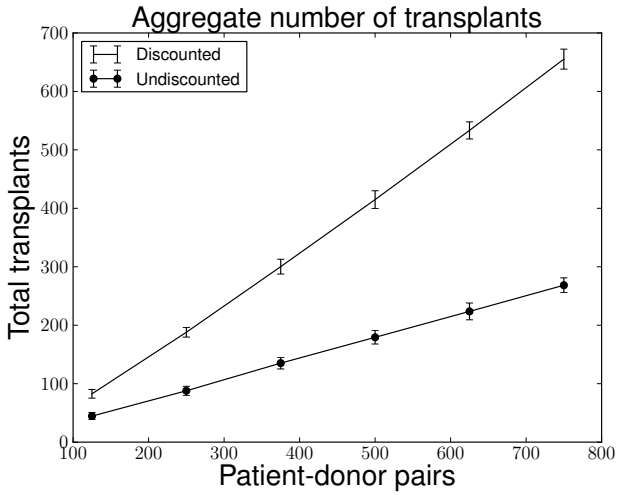


Figure 5.9: Expected aggregate transplants over 24 weeks, for increasing $|P|$ (and $|A| = 0.1|P|$).

Figure 5.9 gives aggregate results for total number of expected transplants over 24 weeks, for graphs of varying size, for both discounted and maximum cardinality matching. Graphs have 10% as many altruists on top of the patient-donor pool. The gap between discounted and non-discounted matching widens as the activity level of the dynamic kidney exchange increases. For our largest graphs, discounted matching improved expected transplants by a factor of three over maximum cardinality matching. In the following section, we will explore how these global efficiency gains change as we prioritize highly-sensitized patients and on graph distributions that more closely mimic presently fielded exchanges.

5.9 Balancing efficiency and fairness in failure-aware kidney exchange

So far, we have motivated a move to discounted kidney exchange optimization from a global efficiency perspective. One might ask how this affects fairness. For example, a proposed transplant to a highly-sensitized patient might intuitively fail with higher probability than one to a patient of low sensitization due to coupled health issues (e.g., chronic illness) in the former, and thus the discounted approach could disfavor highly-sensitized patients. While data from the UNOS kidney exchange [132] does not show a correlation between post-match failure and CPRA, data from other exchanges does show such a correlation [e.g., 22, 99]. Regardless, prioritizing highly-sensitized patients is currently done explicitly or implicitly in fielded kidney exchanges, so we address that here—and then in much greater depth in the next part of the thesis in Chapters 7 and 8.

In general, striking a balance between fairness and efficiency in kidney exchange is an increasingly important line of work combining medical policy, economics, and optimization. Roth et al.

[186] define a fair mechanism to be one that equalizes, to the greatest extent possible, patients' chances of getting a match. While this is almost certainly too strict a fairness criterion to be fielded in practice, the notion of prioritizing some patients—possibly at the cost of overall efficiency in the exchange—is common (and is performed in the current UNOS exchange as well). Recent and parallel work by Bertsimas et al. [35, 36] and by Caragiannis et al. [53] studies the *price of fairness*, a measure of the tradeoff between fairness and efficiency, in general resource allocation problems. We will adapt that in greater depth to kidney exchange in Chapter 7, but overview the instantiation of that adaptation in this chapter. Hooker and Williams [110] provide general Rawlsian equity optimization models that maximize the minimum utility of any one agent or set of agents. Bertsimas et al. [37] design a realistic method for maximizing, given a set of user-defined fairness constraints, some notion of efficiency in the *deceased donor* kidney transplantation problem, where patients on a waiting list are allocated cadaveric kidneys. In general, accurate quantification of the theoretical and empirical advantages and disadvantages of various fairness definitions would be of great value to policymakers in the kidney exchange community.

In this section, we use a recent fairness criterion for kidney exchange, to be discussed in much greater depth in Chapter 7. We show (experimentally) that the "price of fairness" in both static and dynamic failure-aware models is also typically low. More importantly, we show that failure-aware matching under well-chosen fairness criteria results in more expected transplants to *both* the global pool *and* highly-sensitized patients than maximum cardinality matching. We conclude that there is an enormous "price of using the wrong model" that is potentially more harmful to *all* patients—and will present FUTUREMATCH in Chapter 8 as a method to automatically tune toward a "less wrong" model.

5.9.1 Weighted fairness as a prioritization scheme for sensitized patients

One simple method to emphasize a certain class of patient-donor pairs—for us, those in the set of highly-sensitized vertices V_H—is to increase the weight of edges with a sink in V_H. This definition generalizes the policy UNOS currently applies to highly-sensitized patients in the fielded kidney exchange, where incoming edges to patients above a certain CPRA threshold are given a positive constant additive weight increase. We adopt a parameterized form of this rule here.

To implement this rule, in Chapter 7 we will build on the standard kidney exchange integer programming formulation and rewrite the objective as follows:

$$\max \quad \textstyle\sum_c v_\Delta(c) x_c$$

Here, $v_\Delta(c)$ is the value of a cycle or chain c (either the weight in the deterministic model or the discounted utility in our failure-aware model) such that the weight of each edge $e \in c$ is adjusted by some re-weighting function $\Delta : E \to \mathbb{R}$.

A simple example re-weighting function is multiplicative:

$$\Delta^\beta(e) = \begin{cases} (1+\beta)w_e & \text{if } e \text{ ends in } V_H \\ w_e & \text{otherwise} \end{cases}$$

Intuitively, for some $\beta > 0$, this function scales the weight of edges ending in highly-sensitized vertices by $(1+\beta)$. For example, if $\beta = 0.5$, then the optimization algorithm will value edges

that result in a highly-sensitized patient receiving a transplant at 50% above their initial weight (which may then be discounted by other factors like failure probability and chain position, as in our chapter's current model).

For any $M \in \mathcal{M}$, let M' be the matching such that every cycle $c \in M$ has augmented weight $v_\Delta(c)$. Then define the *weighted* fairness rule u_Δ in terms of the utilitarian rule u applied to the augmented matching M', such that $u_\Delta(M) = u(M')$. Thus, the clearing problem is rewritten as finding $M^* = \arg\max_{M \in \mathcal{M}} u_\Delta(M)$. We will discuss this rule and others in greater theoretical and experimental depth in Chapter 7.

In the rest of this section, we explore the effect this weighted fairness rule has on the expected number of transplants performed in the pool as a whole and by highly-sensitized patients in V_H, under a variety of modeling assumptions.

5.9.2 Experiments in the static setting

We begin by studying the weighted fairness rule in the context of static kidney exchange. We do this on both the 161 individual UNOS match runs to date, and on generated graphs that mimic the UNOS graphs. The generator runs by loading all pairs and altruistic donors that have ever been present in the UNOS pool into a set of vertices V, then drawing with replacement vertices from that pool and running the UNOS edge existence algorithm on the sampled vertices to create a compatibility graph. We test these real or sampled graphs under three probability distributions: constant and bimodal as above, as well as a differently-distributed bimodal family that draws failure probabilities in accordance with those rates published by Ashlagi et al. [22]. Critically, this last distribution correlates edge failure rate with patient CPRA; incoming edges to highly-sensitized patients are much more likely to fail than incoming edges to the rest of the pool. Specifically, they state that patients with a CPRA above 75 have a crossmatch failure probability of 0.5, while those with lower CPRA values (reported in ranges [0–24], [25–49], and [50–74]) have much lower probabilities of crossmatch failure (0.05, 0.2, and 0.35, respectively). They also experiment with an additional additive exogenous failure rate varied between 0 and 0.16; we use 0.08 in our experiments.

Constant failure rate

We begin by assuming that every edge fails with the same constant probability, as in previous sections. This assumption, while not likely to hold in practice, is easily parameterized and allows us to explore the differences in models as matchings become less reliable. Different exchanges have different failure rates, and this exploratory analysis might serve as a useful tool to quantify the marginal gains of decreasing edge failure rates.

Figure 5.10 compares the weighted fairness rule u_Δ applied to the failure-aware model against the utilitarian rule applied to the deterministic model, which computes a maximum cardinality disjoint cycle cover without regard for edge failure. Figure 5.10(left) shows that the efficient failure-aware matching always results in at least as many (typically more) expected transplants as the efficient deterministic matching. However, interestingly, even matchings under the fair rule u_Δ in the failure-aware model often result in significant overall gains when compared to the

utilitarian deterministic matching. Figure 5.10(right) shows that even the *fully efficient* matching rarely results in a loss of highly-sensitized transplants, and that even slightly prioritizing sensitized patients results in large gains (at low cost to global efficiency).

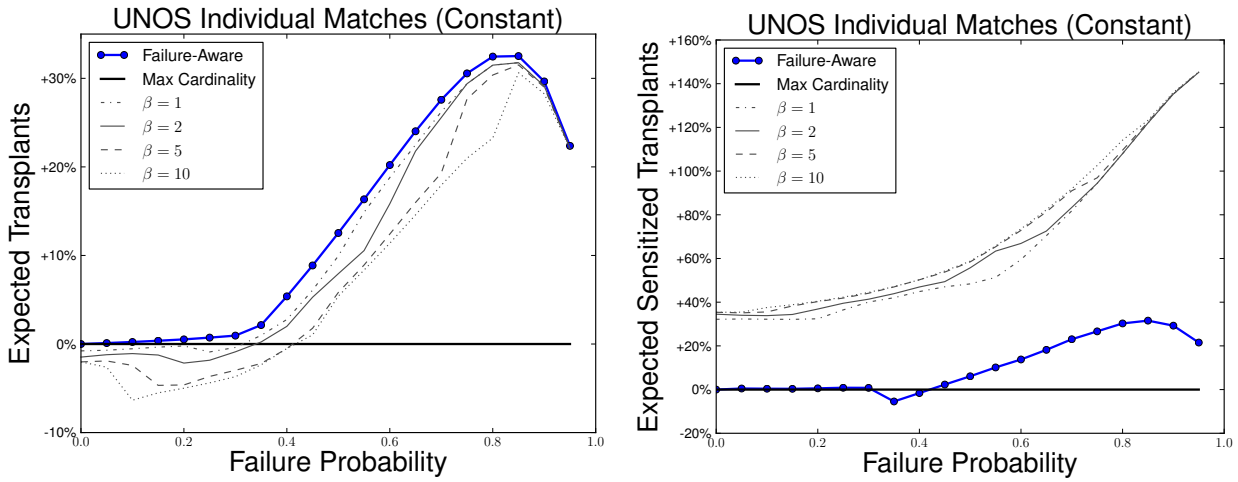


Figure 5.10: Percentage change in expected number of transplants (left) and sensitized transplants (right) for *actual* UNOS match runs using failure-aware matching—possibly with fairness constraints—instead of maximum cardinality matching. The x-axis varies constant edge failure probability from 0 to near 1.

For example, for $\beta = 1.0$—that is, when highly-sensitized patients are valued twice as much as lowly-sensitized patients—we see a drop of only a couple of percentage points of expected transplants when there is *no* probability of edge failure. This is countered by a very large (over 30%) gain in the expected number of highly-sensitized transplants. In fact, when the probability of edge failure is at least 45%, valuing highly-sensitized transplants at 11x ($\beta = 10.0$) that of a lowly-sensitized patient results in more expected total transplants than deterministic matching that does not consider fairness.

Also, we see that efficient failure-aware matching almost always results in more expected sensitized transplants than deterministic matching, with the exception of a small relative drop at failure rates around 35–45%. This can be explained by comparing, given a failure probability p, the relative discounted utilities of a 2-cycle c_2 ($u(c_2) = 2(1-p)^2$) and 3-cycle c_3 ($u(c_3) = 3(1-p)^3$). When $p < \frac{1}{3}$, $u(c_2) < u(c_3)$, so the optimizer favors 3-cycles over 2-cyles. When $p > \frac{1}{3}$, $u(c_2) > u(c_3)$, so the optimizer favors 2-cycles. Highly-sensitized patients are often matched in 3-cycles; intuitively, if a highly-sensitized pair's donor can donate to another pair, it is more likely that this pair will not be able to connect back to the highly-sensitized pair directly (by virtue of that initial pair being highly-sensitized and thus having low in-degree) via a 2-cycle but will rather connect back through a lowly-sensitized pair via a 3-cycle). So, for $p < \frac{1}{3}$, failure-aware gains are only realized by rearranging the low-probability tails of chains into 2- and 3-cycles, while for $p > \frac{1}{3}$, failure-aware optimization may start to cannibalize 3-cycles (that likely contain highly-sensitized pairs). Empirically, this is only an issue for $p \in (\frac{1}{3}, 0.45]$; once $p > 0.45$, the efficient objective's gains

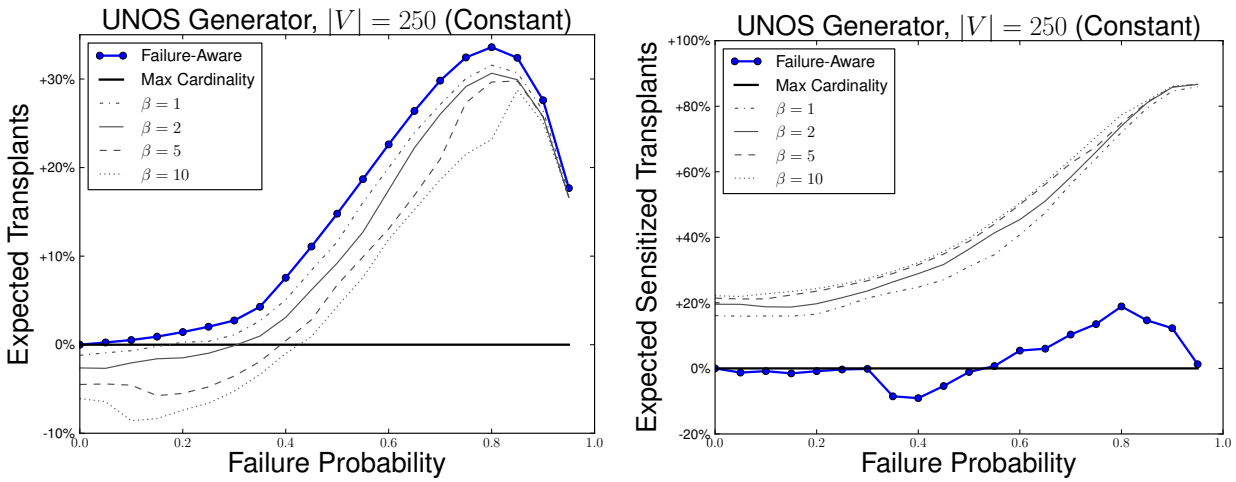


Figure 5.11: Percentage change in expected number of transplants (left) and sensitized transplants (right) for *generated* UNOS match runs using failure-aware matching—possibly with fairness constraints—instead of maximum cardinality matching. The x-axis varies constant edge failure probability from 0 to near 1.

outweigh these losses. Furthermore, we see that a small prioritization (even $\beta = 1$) results in both global and sensitized gains even for $p \in (\frac{1}{3}, 0.45]$ (and for other values of p).

This general behavior is supported in Figure 5.11, which shows the same experiments on generated data that mimics the UNOS distribution, for pools of size 250—roughly the size of the current UNOS pool. We include these results because, in Section 5.9.3, we run dynamic experiments on data that mimics the UNOS pool (unlike the results in Section 5.8, which used the generator due to Saidman et al. [191]). The similarity of Figures 5.10 and 5.11 serves as validation of the simulator.

It may be difficult to accurately estimate failure probabilities on edges in practice. Indeed, in extreme cases, it may even be deemed unethical to allow vastly different failure probabilities to be included in the optimization process, as the probabilities act as a prioritization tool. As these experiments show, one could simply set all the probabilities in the optimization to be equal in order to not disfavor patients with high failure probabilities. Even with this extreme approach, the discounted framework strikes good endogenous tradeoffs between short chains, long chains, short cycles, and long cycles—unlike the current undiscounted approach.

Bimodal failure rate

We now consider the weighted fairness rule in the static setting with bimodal failure probabilities. We will refer to the prior bimodal failure distribution derived in Section 5.4, where edge failure rates are not correlated with patient CPRA, as the "UNOS Bimodal" distribution. We also perform experiments on a distribution derived from published failure rates from a different exchange, the Alliance for Paired Donation (APD), where edge failures *are* correlated with patient CPRA [22]. We refer to this distribution as "APD Bimodal." This difference in correlations could be due to

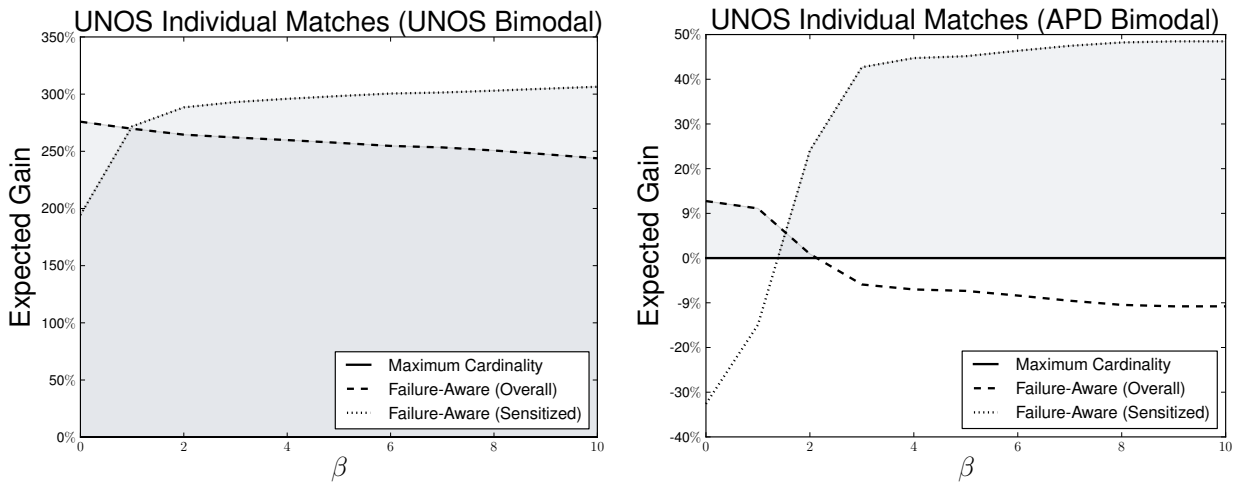


Figure 5.12: Change in the expected number of transplants on average for *actual* UNOS match runs when using failure-aware matching instead of maximum cardinality matching, assuming bimodal edge failure rates derived from UNOS (left) and APD (right). The x-axis varies the β fairness factor applied to the failure-aware matching algorithm.

highly-sensitized patients being less likely to find a match outside of the exchange (e.g., on the deceased donor wait list or another exchange) but more likely to have a match fail due to medical reasons such as crossmatch incompatibility—whereas an easy-to-match patient might quickly find a living donor elsewhere, but be less likely to have a match fail for medical reasons. UNOS has a slower matching cadence than some other exchanges like the National Kidney Registry (NKR), which matches whenever the underlying compatibility graph changes, so easily-matched patients may be "sniped" by such faster-moving exchanges. By lowering these non-medical reasons for failure (e.g., by merging all exchanges into a single program to reduce inter-exchange competition), the overall failure rate for highly-sensitized patients would probably become higher than that of other patients.

Figure 5.12 shows expected gains in both the number of overall transplants (dashed line) and sensitized transplants (dotted line) relative to a baseline of deterministic matching (solid line). The expected number of failure-aware overall and highly-sensitized transplants are compared against the expected number of deterministic overall and highly-sensitized transplants, respectively, as the fairness factor β is increased from 0 (fully efficient matching) to 10 (highly biased matching).

Immediately visible is that, when failure rates are not correlated to CPRA, the gains seen by failure-aware matching are quite large across the board. This aligns with our Saidman-generated results from Section 5.8, as well. However, when failure rates are highly correlated with patient CPRA, the situation becomes more delicate. Failure-aware matching without fairness considerations does result in a large gain in overall expected transplants, but harms highly-sensitized patients. We can identify a "sweet spot" that balances these conflicting objectives; empirically, this is approximately when $\beta \in [2, 4]$ (depending on the underlying family of graphs; see Figures 5.12 and 5.13). When β is toward the lower end of this interval, the loss in marginalized transplants is zero while the gain in global expected transplants is positive (approximately 10%). When β

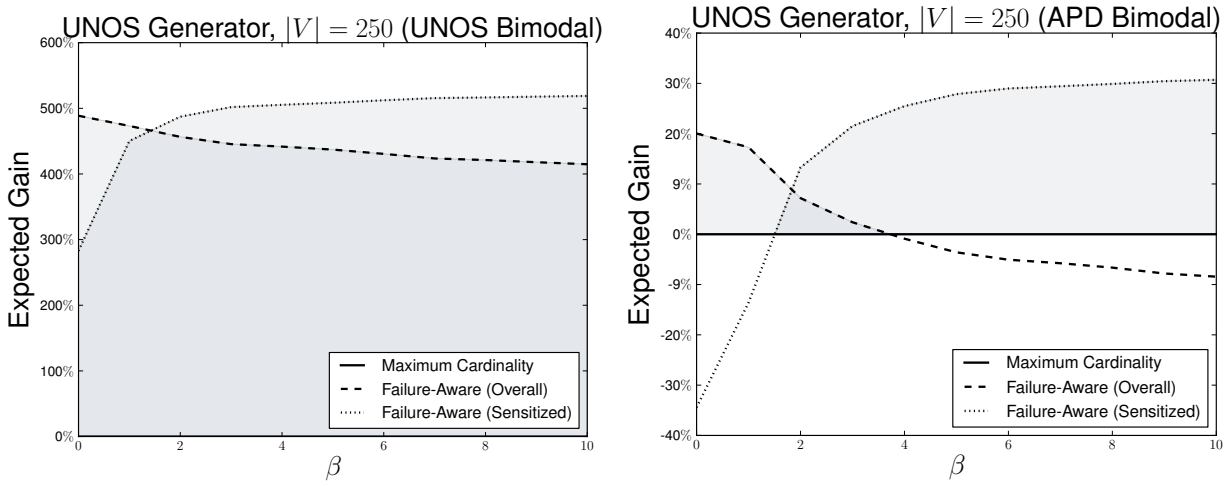


Figure 5.13: Change in the expected number of transplants on average for *generated* UNOS match runs when using failure-aware matching instead of maximum cardinality matching, assuming bimodal edge failure rates derived from UNOS (left column) and APD (right column). The x-axis varies the β fairness factor applied to the failure-aware matching algorithm.

is toward the higher end of this range, the global gain in transplants is zero while the gain in marginalized transplants is positive (approximately 25%). Within the interval, we realize gains in both objectives—a clear win.

As in the constant failure probability case, Figure 5.13 shows similar results on generated UNOS compatibility graphs, under both failure rate distributions, for $|V| = 250$. This provides validation for our simulator. In the rest of the section, we further explore the correlated failure rate setting in the realistic dynamic kidney exchange simulator presented in Section 5.8 using these equally realistic compatibility graphs, and show that this same balance of fairness and efficiency can be struck so that *both* global efficiency and the expected number of transplants to highly-sensitized patients increases.

5.9.3 Experiments in the dynamic setting

We now continue our exploration of the correlated failure probability case into a dynamic model. This is important because, although we showed that a balance can be struck between efficiency and fairness in the static case such that failure-aware matching results in gains in both objectives, it is possible that this balance comes at the cost of matching "easier" hard-to-match pairs in the now and leaving the "hardest" hard-to-match pairs for later. We show that this is not the case. Specifically, the same winning balance can be struck in the dynamic setting. (In the interest of space, we do not include experiments in the non-correlated bimodal failure case, because even failure-aware matching without fairness considerations results in large increases in both global and marginalized transplants over time. In this sense, the experiments in this section on the correlated APD distribution give a conservative estimate of the gains seen by failure-aware matching in dynamic kidney

exchange.)

We perform experiments in the same dynamic model as Section 5.8, only this time using the realistic UNOS graph generator validated above. We vary arrival rates over 24 time periods with $\{12, 16 \ldots, 32\}$ pairs or altruistic donors arriving per time period, as sampled from the real pairs and altruists. Tables 5.5 and 5.6 show the median overall absolute and percentage gains and losses in number of transplants and number of sensitized transplants, respectively, aggregated over all time periods by failure-aware matching for $\beta \in \{0, 1, \ldots, 5\}$ compared against deterministic matching.

	$\|V\| = 300$		$\|V\| = 400$		$\|V\| = 500$		$\|V\| = 600$		$\|V\| = 700$		$\|V\| = 800$	
	Gain	(%)	*Gain*	(%)	*Gain*	(%)	*Gain*	(%)	*Gain*	(%)	*Gain*	(%)
EFFICIENT	+0	(0.0%)	+5	(5.9%)	+1	(1.9%)	+2	(2.5%)	+9	(7.1%)	+5	(3.6%)
FAIR, $\beta = 1$	+2	(4.2%)	+5	(6.7%)	+1	(1.0%)	+8	(8.1%)	+8	(6.2%)	+11	(7.3%)
FAIR, $\beta = 2$	+0	(0.0%)	+3	(4.1%)	+0	(-1.3%)	+3	(2.4%)	+2	(1.8%)	+5	(3.4%)
FAIR, $\beta = 3$	+2	(4.3%)	-1	(-2.1%)	-1	(-1.1%)	-1	(-1.3%)	+3	(2.8%)	+2	(1.5%)
FAIR, $\beta = 4$	+2	(4.3%)	+2	(2.5%)	+2	(2.5%)	-1	(-1.3%)	+1	(0.9%)	+3	(2.3%)
FAIR, $\beta = 5$	+0	(-0.1%)	+1	(2.0%)	+3	(4.0%)	+0	(-0.5%)	-1	(-0.8%)	-2	(-1.7%)

Table 5.5: Gains in expected number of transplants overall, for increasing values of fairness β and for different arrival rates.

	$\|V\| = 300$		$\|V\| = 400$		$\|V\| = 500$		$\|V\| = 600$		$\|V\| = 700$		$\|V\| = 800$	
	Gain	(%)	*Gain*	(%)	*Gain*	(%)	*Gain*	(%)	*Gain*	(%)	*Gain*	(%)
EFFICIENT	-4	(-40.0%)	-2	(-21.4%)	-3	(-15.4%)	-4	(-21.4%)	-5	(-23.4%)	-6	(-19.1%)
FAIR, $\beta = 1$	-2	(-26.1%)	+0	(0.0%)	-1	(-10.0%)	+0	(0.0%)	+0	(-1.3%)	-1	(-4.4%)
FAIR, $\beta = 2$	+1	(9.5%)	+3	(18.8%)	+0	(1.2%)	+2	(9.9%)	+2	(11.2%)	+5	(15.5%)
FAIR, $\beta = 3$	+0	(5.6%)	+1	(10.8%)	+1	(11.7%)	+7	(35.1%)	+8	(33.2%)	+6	(20.3%)
FAIR, $\beta = 4$	+0	(5.6%)	+3	(29.0%)	+2	(11.0%)	+8	(46.2%)	+6	(23.9%)	+8	(29.3%)
FAIR, $\beta = 5$	+0	(0.0%)	+2	(22.6%)	+2	(12.1%)	+8	(43.7%)	+6	(24.0%)	+8	(23.9%)

Table 5.6: Gains in expected number of highly-sensitized transplants, for increasing values of β and for different arrival rates.

Mirroring the static experiments above, we see that for low values of β, failure-aware matching results in global gains and marginalized losses. However, as above, for $\beta \approx 2$, a winning balance is struck, with nonnegative gains in expected overall transplants and significant gains in number of highly-sensitized transplants. Perhaps most excitingly, for higher values of β, the number of highly-sensitized transplants increases markedly (reaching +20%–+40% over deterministic matching for higher arrival rates), while the overall effect on global efficiency is negligible. In reality, kidney exchanges are often seen as a "last hope" for highly-sensitized patients; even with a higher likelihood of pre-transplant match failure, we have shown that failure-aware matching can increase successful match rates for these highly-prioritized patients at no cost to the global system efficiency.

5.10 Pre-match edge testing complements failure-aware matching

In this section, we present an approach to dealing with uncertainty over the (non)existence of edges in a compatibility graph by querying for existence a small set of edges *before* running the matching algorithm. The matching algorithm that is run after the testing phase can be one that takes edge failure into account (or not); in this sense, the approach of this chapter is a complement to the failure-aware matching model presented earlier in this chapter. Results from this section, as in the rest of the chapter, are stated in the language of a *stochastic matching* problem, which is a special case of *stochastic k-set packing*, where each set exists only with some probability, and the problem is to find a packing of maximum size of those sets that do exist.

Without any constraints, one can simply query all edges or sets, and then output the maximum matching or packing over those that exist—but this level of freedom may not always be available, as is the case in kidney exchange. This chapter is interested in the real-world tradeoff between the number of queries and the fraction of the omnisicient optimal solution achieved. Toward that end, it works in settings where a clearinghouse can test edges *adaptively in rounds*, where test results from one round can inform the succeeding rounds, and *non-adaptively*, where only a single batch of edges can be tested prior to running the final matching algorithm.

In this section, we support the theoretical results of Blum et al. [44] with empirical simulations from two kidney exchange compatibility graph distributions. We operate in the following model. For any graph $G = (V, E)$, let $M(E)$ denote its maximum (cardinality) matching. (The notation $M(E)$ intentionally suppresses the dependence on the vertex set V, since we care about the maximum matchings of different subsets of edges for a fixed vertex set.) Given a set of edges X, define $X_{\mathbf{p}}$ to be the random subset formed by including each edge x_i of X independently with probability $p_i \in \mathbf{p}$, the vector of edge failure probabilities for each edge in X.

Given a graph $G = (V, E)$, define $\overline{M}(E)$ to be $\mathbb{E}[|M(E_{\mathbf{p}})|]$, where the expectation is taken over the random draw $E_{\mathbf{p}}$. In addition, given the results of queries on some set of edges T, define $\overline{M}(E|T)$ to be $\mathbb{E}[|M(X_{\mathbf{p}} \cup T')|]$, where $T' \subseteq T$ is the subset of edges of T that are known to exist based on the queries, and $X = E \setminus T$.

In the *non-adaptive* version of the problem, the goal is to design an algorithm that, given

a graph $G = (V, E)$ with $|V| = n$, queries a subset X of edges in parallel such that $|X| = O(n)$—in our algorithms, a constant number of incident edges per vertex—and maximizes the ratio $\overline{M}(X)/\overline{M}(E)$.

In contrast, an *adaptive* algorithm proceeds in rounds, and in each round queries a subset of edges in parallel. Based on the results of the queries up to the current round, it can choose the subset of edges to test in the next round. Formally, an R-*round adaptive* stochastic matching algorithm selects, in each round r, a subset of edges $X_r \subseteq E$, where X_r can be a function of the results of the queries on $\bigcup_{i<r} X_i$. The objective is to maximize the ratio $\mathbb{E}[|M(\bigcup_{1\leq i\leq R} X_i)|]/\overline{M}(E)$, where the expectation in the numerator is taken over the outcome of the query results and the sets X_i chosen by the algorithm.

As above, we test on the dense distribution due to Saidman et al. [191], as well as on *real* kidney exchange compatibility graphs drawn from the first 169 match runs of the UNOS nationwide kidney exchange. While these two families of graphs differ substantially, we find that even a small number R of non-adaptive rounds, followed by a single period during which only those edges selected during the R rounds are queried, results in large gains relative to the omniscient matching.

This section does not directly test the algorithms we presented in Blum et al. [44], which focuses solely on polynomial-time algorithms. For the 2-cycles-only case, we do directly implement Algorithm 1 from the full paper with Blum et al. [44]. However, for the cases involving longer cycles and/or chains, we do not restrict ourselves to polynomial time algorithms (unlike in the theory part of this chapter), instead choosing to optimally solve matching problems using integer programming during each round, as well as for the final matching and for the omniscient benchmark matching. This decision is informed by the current practice in kidney exchange, where computational resources are much less of a problem than human or monetary resources (of which the latter two are necessary for querying edges).

In our experiments, the planning of which edges to query proceeds in rounds as follows. Each round of matching calls as a subsolver the failure-aware matching algorithm presented earlier in this chapter, which includes edge failure probabilities in the optimization objective to provide a maximum discounted utility matching. The set of cycles and chains present in a round's discounted matching are added to a set of edges to query, and then those cycles and chains are constrained from appearing in future rounds. After all rounds are completed, this set of edges is queried, and a final maximum discounted utility matching is compared against an omniscient matching that knows the set of non-failing edges up front.

Appendix C.2 presents an algorithm that relaxes the assumption that all edges in a round must be tested at once, and provides initial experimental results comparing against the round-based algorithms in this chapter. That appendix provides an implementation of a basic "expected improvement"-style algorithm (see, e.g., work by Jones et al. [122] for foundational work on general expected improvement algorithms) for adaptive edge testing of barter exchange graphs. There, we work under the assumption that testing can be performed and that information digested on an edge-by-edge basis, as opposed to in larger batches as in this chapter; our experimental results show that by relaxing the batch assumption, we can achieve the same expected match size as the batch algorithms in this chapter by testing even fewer edges.

5.10.1 Experiments on dense generated graphs due to Saidman et al. [191]

We begin by looking at graphs drawn from a distribution due to Saidman et al. [191]. Figure 5.14 presents the fraction of the omniscient objective achieved by $R \in \{0, 1, \ldots, 5\}$ non-adaptive rounds of edge testing for generated graphs with 250 patient-donor pairs and no altruistic donors, constrained to 2-cycles only (left) and both 2- and 3-cycles (right). Note that the case $R = 0$ corresponds to no edge testing, where a maximum discounted utility matching is determined by the optimizer and then compared directly to the omniscient matching. The x-axis varies the uniform edge failure rate f from 0.0, where edges do not fail, to 0.9, where edges only succeed with a 10% probability. Given an edge failure rate of f in the figures below, we can translate to the p used in the theoretical section of the paper as follows: 2-cycles exists with probability $p_{\text{2-cycle}} = (1 - f)^2$, while a 3-cycle exists with $p_{\text{3-cycle}} = (1 - f)^3$. For example, in the case of $f = 0.9$, a 3-cycle exists with very low probability $p = 0.001$.

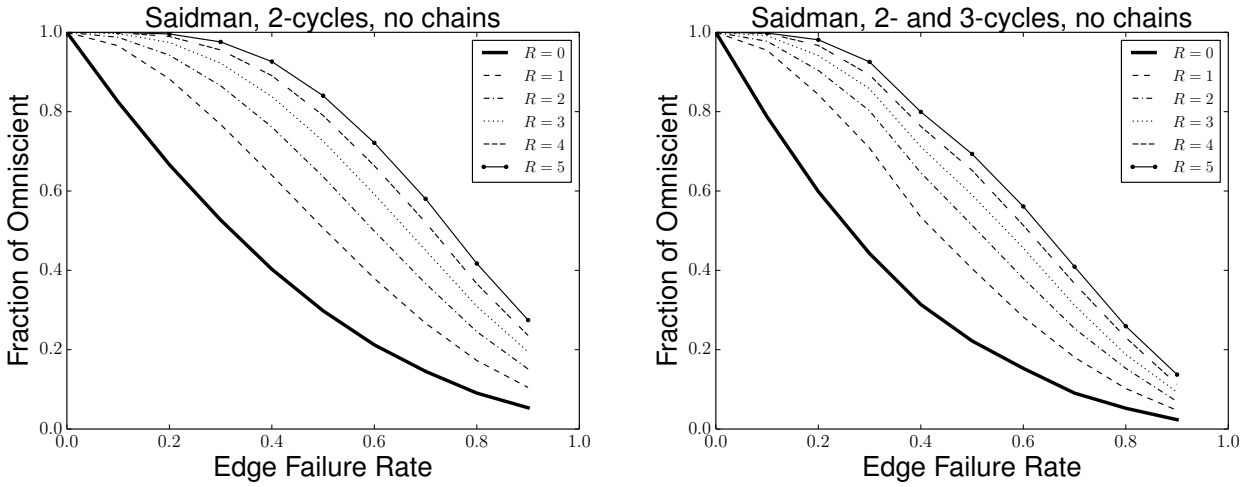


Figure 5.14: Saidman generator graphs constrained to 2-cycles only (left) and both 2- and 3-cycles (right).

The utility of even a small number of edge queries is evident in Figure 5.14. Just a single round of testing ($R = 1$) results in 50.6% of omniscient—compared to just 29.8% with no edge testing—for edge failure probability $f = 0.5$ in the 2-cycle case, and there are similar gains in the 2- and 3-cycle case. For the same failure rate, setting $R = 5$ captures 84.0% of the omnisicient 2-cycle matching and 69.3% in the 2- and 3-cycle case—compared to just 22.2% when no edges are queried. Interestingly, we found no statistical difference between non-adaptive and adaptive matching on these graphs.

5.10.2 Experiments on real match runs from the UNOS kidney exchange

We now analyze the effect of querying a small number of edges per vertex on graphs drawn from the real world. Specifically, we use the first 169 match runs of the UNOS nationwide kidney exchange. These graphs, as discussed in Chapter 3, are substantially less dense than those produced

by the Saidman generator. As shown in, e.g., Chapter 2 and later in Chapter 7, this disparity between generated and real graphs has led to different theoretical results (e.g., efficient matching does not require long chains in a deterministic dense model [19, 74] but does in a sparse model [23]) and empirical results (both in terms of match composition and experimental tractability [16, 63, 100]) in the past—a trend that continues here.

Figure 5.15 shows the fraction of the omniscient 2-cycle and 2-cycle with chains match size achieved by using only 2-cycles or both 2-cycles and chains and some small number of non-adaptive edge query rounds $R \in \{0, 1, \ldots, 5\}$. For each of the 169 pre-test compatibility graphs and each of edge failure rates, 50 different ground truth compatibility graphs were generated. Chains can partially execute; that is, if the third edge in a chain of length 3 fails, then we include all successful edges (in this case, 2 edges) until that point in the final matching. More of the omniscient matching is achieved (even for the $R = 0$ case) on these real-world graphs than on those from the Saidman generator presented in Section 5.10.1. Still, the gain realized even by a small number of edge query rounds is stark, with $R = 5$ achieving over 90% of the omniscient objective for every failure rate in the 2-cycles-only case, and over 75% of the omniscient objective when chains are included (and typically much more).

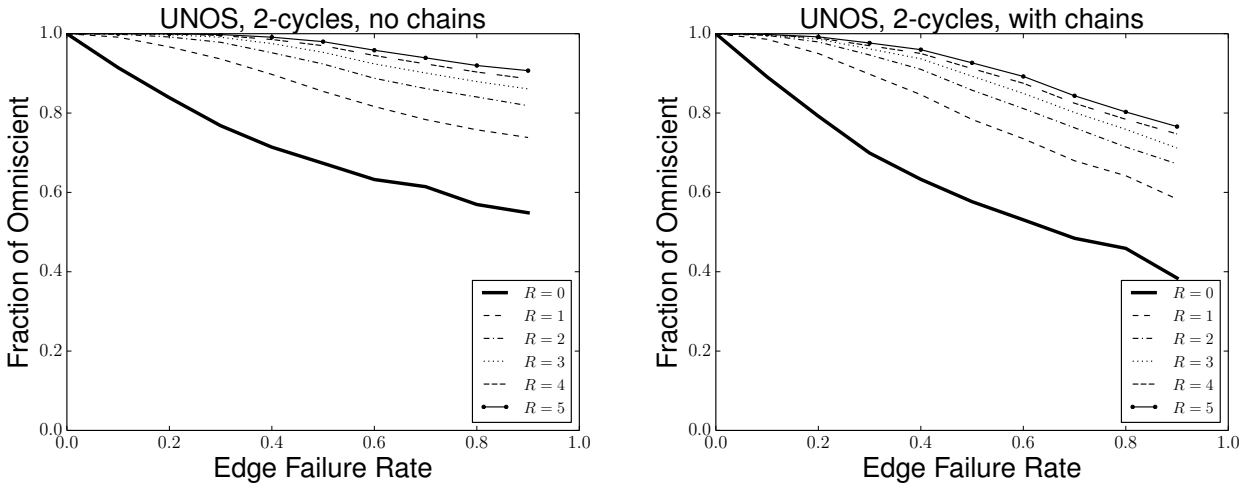


Figure 5.15: Real UNOS match runs constrained to 2-cycles (left) and both 2-cycles and chains (right).

Figure 5.16 expands these results to the case with 2- and 3-cycles, both without and with chains. Slightly less of the omniscient matching objective is achieved across the board, but the overall increases due to $R \in \{1, \ldots, 5\}$ non-adaptive rounds of testing is once again prominent. Interestingly, we did not see a significant difference in results for adaptive and non-adaptive edge testing on the UNOS family of graphs, either.

We provide additional experimental results in Appendix C. Code to replicate all experiments is available at `https://github.com/JohnDickerson/KidneyExchange`; this codebase includes graph generators but, due to privacy concerns, does not include the real match runs from the UNOS exchange.

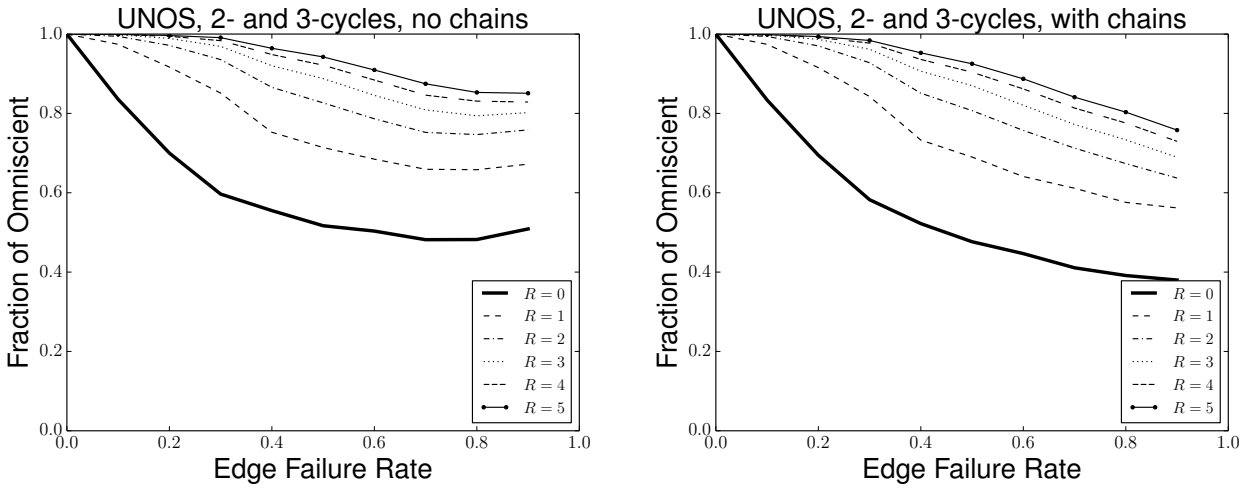


Figure 5.16: Real UNOS match runs with 2- and 3-cycles and no chains (left) and with chains (right).

5.11 Conclusions, future work, & implications

In this chapter, we addressed the problem of edges in a matching (e.g., recommended transplants in a kidney exchange) failing after a matching algorithm has committed to them. This is a timely problem; in the UNOS nationwide kidney exchange, very few of the algorithmically matched patients actually receive a transplanted kidney through the exchange, and similar rates apply to other kidney exchanges.

Failure-aware kidney exchange. We first introduced a failure probability to each edge in a compatibility graph, and defined an expected utility of edges, cycles, chains, and matches. This model drives our main theoretical result, that (with high probability, in a random graph model) there exists a non-maximum cardinality matching that provides linearly more utility than any maximum cardinality matching. We then ran simulations on real data from all UNOS match runs between 2010 and late 2014, and found that our failure-aware matching increases the number of expected transplants dramatically.

Armed with this new model, we showed that the current state-of-the-art kidney exchange solver (used in the UNOS kidney exchange) cannot be used for this problem because now each edge has both a weight and a failure probability, and simply multiplying them to get a revised weight would make the algorithm incorrect. We designed a branch-and-price-based optimal clearing algorithm specifically for the probabilistic exchange clearing problem. It has many enhancements over the prior best kidney exchange clearing algorithm. For one, we designed a failure-aware column generator that incrementally brings only "possibly good" chains into consideration. We showed experimentally that this new solver scales well on large simulated data. We also explored the idea of immediately reentering failed cycles and chain segments from an initial matching back into the waiting pool and subsequently rerunning the matching algorithm again; this instantaneous rematching

results in significant extra transplants and can be performed multiple times with relative ease. We then developed a realistic model of dynamic kidney exchange based on our experiences with, and data from, UNOS, and showed that failure-aware matching in dynamic graphs increases expected transplants significantly. Finally, we explored the effect of failure-aware matching on marginalized patients; it is possible to strike a balance between fair and efficient failure-aware matching that results in more expected transplants both globally and to marginalized patients specifically, in both the static and dynamic cases, in a variety of graph distributions.

Experimentally, our solver would benefit from a better (i.e., tighter) upper bound on the discounted clearing problem—the current bound is especially loose when failure probabilities are high and when bounding the utility of long cycles and long chains. Tightening the upper bound would decrease the size of the search tree and, in turn, reduce column generation and overall runtime. The accuracy of our and others' experimental results relative to real kidney exchange will continue to improve as we work with more exchanges. The community's understanding of the underlying failure probabilities—especially on a patient-by-patient basis—will improve as more data becomes available; some initial work has already been done by [99] using simple models driven by individual patients' data to predict crossmatch failures. Further quantitative analysis of both international living donor and kidney exchange data would be of great help to both the science and practice of kidney exchange.

Theoretically, exploring the efficacy of failure-awareness in a fully dynamic model of kidney exchange (like that presented in Section 5.8, where vertices and edges arrive and depart over time for a variety of reasons) would be of practical interest in designing dynamic matching algorithms and in making high-level policy recommendations. Following seminal work by [213], recent research [e.g., 9, 15, 24] explores different analytical models of dynamic deterministic kidney exchange and finds different optimal (or near-optimal) policies in each model. Extending any of these models to include post-algorithmic match failure and then observing the effect this has on, e.g., optimal matching cadence would be of interest. Also, as we showed empirically in Section 5.9, matching to blindly maximize efficiency (in this case, maximizing long-term number of matches or transplants, or minimizing average waiting time) can further marginalize already hard-to-match patients. Formalizing the tradeoff that we showed can be struck empirically between efficient and fair short- and long-term matching in an analytical dynamic model with edge failures would be extremely informative. In Chapter 7, we will go into greater depth on the so-called "price of fairness" in kidney exchange; this work is a batch setting, and perhaps could be extended.

Finally, we note that balancing efficiency and fairness was done by hand in this section. Chapter 8 will present FUTUREMATCH, a framework that learns to strike that balance automatically.

Adaptive pre-match edge testing. More accurate models are often not adopted quickly, if at all, by exchanges. One reason for this is complexity—and not in the computational sense. Humans—doctors, lawyers, and other policymakers who are not necessarily versed in optimization or theoretical economics and computer science—and the organizations they represent rightfully wish to understand the workings of an exchange's matching policy. The techniques described in this chapter are particularly exciting in that they are quite easy to explain in accessible language and they involve only mild changes to the status quo. At a high level, we are proposing to test some

small number of promising potential matches for some subset of patient-donor pairs in a pool. As Section 5.10.2 shows, even a *single* extra edge test per pair will produce substantially better results.

Any new policy for kidney exchange has to address three practical restrictions in this space: (i) the monetary cost of crossmatches, (ii) the number of crossmatches that can be performed per person, as there is an inherent limit on the amount of blood that can be drawn from a person, and (iii) the time it takes to find the matches, as time plays a major role in the health of patients and crossmatches become less accurate as time passes and the results get old. For both our non-adaptive and adaptive algorithms, even a very small number of rounds ($R \leq 5$) results in a very large gain in the objective. This is easily within the limits of considerations (i) and (ii) above. Our non-adaptive algorithm performs all chosen crossmatches in parallel, so the time taken by this method is similar to the current approach. Our adaptive algorithm, in practice, can be implemented by a one-time retrieval of R rounds worth of blood from each donor-patient pair, then sending that blood to a central wet laboratory. Most crossmatches are performed via an "immediate spin", where the bloods are mixed together and either coagulate (which is bad) or do not (which is good). These tests are very fast, so a small number of rounds could be performed in a single day (assuming that tests in the same round are performed in parallel). Therefore, the timing constraint (iii) is not an issue for small R (such as that used in our experiments) for the adaptive algorithm.

Clearly, more extensive studies would need to be undertaken before an exact policy recommendation could be made. These studies could take factors like the monetary cost of an extra crossmatch test or variability in testing prowess across different medical laboratories into account explicitly during the optimization process. Furthermore, various prioritization schemes could be implemented to help, for example, hard-to-match pairs find a feasible match by assigning them a higher edge query budget than easier-to-match pairs. The positive theoretical results presented in our paper with Blum et al. [44], combined with the promising experimental results on real data presented in this chapter, provide a firm basis and motivation for this type of policy analysis.

Life can only be understood backwards; but it must be lived forwards.

– Søren Kierkegaard

6 Dynamic kidney exchange

Chapter 5 dealt with "last-minute" failures in barter exchanges, where the optimizer is interested in finding a good batch match within a single time period, but with no eye toward future match runs. This is how fielded kidney exchange algorithms typically match patients to donors in a myopic fashion, maximizing the number of candidates who get kidneys (usually on a weekly, monthly, or bimonthly basis) in an offline fashion. However, this is sub-optimal, since patients and donors arrive and leave the pool over time. Recent work shows that more candidates can be matched over time by considering the future [9, 17, 24, 27, 213, 223]; those approaches are either overly simplified or do not scale.

In this chapter, we introduce a method for informing myopic optimization (e.g., matching) about the future in dynamic applications. It automatically learns *potentials* of elements of the problem (e.g., of vertices or edges in a graph) offline, and then uses these potentials to guide myopic matching at run time. The potential represents an estimate of how much that element can contribute to the objective in the future. The potentials can be viewed as policy parameters to be optimized using a black box program; we are able to leverage state-of-the-art automated algorithm configuration tools to learn their values [113]. Then, at run time, we simply run an offline matching algorithm at each time period, but subtracting out in the objective the potentials of the elements used up in the matching. This causes the batch optimizer—which is traditionally myopic—to take the future into account without suffering a run-time cost.

We apply these techniques to kidney exchange. We first prove bounds on the power of learning potentials on vertices, edges, cycles, and the entire graph (which is optimal). Then, we learn potentials for vertices using a state-of-the-art parameter learning package and kidney exchange instance generator. We show that these learned potentials beget a greater number of matches than simple myopic matching at nearly no run time cost across all the settings tested. The work in this chapter will then be included in a general framework for dynamic matching, FUTUREMATCH, to be presented in Chapter 8.

Related Publications

A preliminary version of the work in this chapter appeared at AAAI-12; it is a collaboration between Dickerson, Procaccia, and Sandholm [74].

6.1 Dynamic matching

In dynamic problems, the problem changes over time. In kidney exchange, patient-donor pairs and altruists enter and expire. While fielded kidney exchanges currently operate under the static paradigm described above, recent work in the kidney exchange community has shown that optimizing dynamically leads to higher cardinality matching overall. We now overview some related work in dynamic matching and barter exchange, extending on the brief overview given in Section 5.1.

From a theoretical standpoint, Ünver [213] derives efficient dynamic mechanisms for general barter exchanges such that the total exchange surplus is maximized. Those results are a generalization of an abstracted kidney exchange. He derives dynamic dispatch policies for kidney exchange in that abstract model; however, the model itself does not accurately reflect real-world kidney exchange. The results hinge on the assumption that one pair's candidate will be compatible with another pair's donor if they are blood type (aka ABO) compatible, ignoring other aspects of the potential match, most critically tissue type. His model also does not have chains. This is a key difference: the setting he studies is sufficiently simple that optimal policies can be analytically derived while in our setting even the batch problem is NP-complete, as explained above. [24] work in a theoretical model with two classes of patients (based on tissue type but not blood type) and batch matching. They characterize waiting times between batch match runs in this reduced model for cases involving only 2-cycles, 2- and 3-cycles, and 2-cycles with a single chain. Anderson et al. [15] and Ashlagi et al. [26] extend the results of that paper using a similar model, while Akbarpour et al. [9] investigates the effect of match cadence in a stylized dynamic model of traditional matching.

Very few empirical results on non-myopic matching in unsimplified dynamic kidney exchange are known. Most notable, prior to our work, was a paper by Awasthi and Sandholm [27] that uses trajectory-based online stochastic optimization algorithms to inform the matching algorithm of possible futures, thus potentially holding off matching some candidates and donors in an effort to increase overall matches later. Their results are promising, but the algorithm does not scale beyond fairly small exchanges due to the empirical complexity of sampling a large number of future world states, and the memory requirements associated with storing those trajectories and optimizing what to do in the present in light of them. Anshelevich et al. [17] perform simulations in a reduced model of kidney exchange including only cycles of length two; they look at batch matching in this model where pairs lose utility from waiting in the pool via a fixed discount parameter.

Many papers include experimental results on dynamic kidney exchange using myopic clearing [22, 23, 65, 74, 75, 96, 97]; indeed, we gave such experimental results in Chapters 2 and 5. This is useful due to the ubiquity of myopic matching in fielded kidney exchanges, but, as we show in this chapter, ultimately the strategy is flawed. Myopic matching that is *informed about the future*

results in more matches.

Finally, while we focus on kidney exchange in this work, the dynamic matching problem in *deceased* organ allocation has been studied before. That problem is fundamentally different than kidney exchange, but we briefly review that work here for completeness. Su and Zenios [202] study the online allocation problem from a theoretical point of view, where patients and kidneys have types, kidneys arrive over time, kidney types are not known ahead of time, and patients much choose to accept or decline kidney offers. Bertsimas et al. [37] propose a data-driven allocation policy for the deceased-donor kidney wait list; they work within today's point-based allocation framework, and balance efficiency over time with various notions of fairness. Alagoz et al. [11] formulate the organ acceptance or rejection problem for the deceased-donor liver wait list as a Markov decision process and study its properties using real data, while Akan et al. [8] derive optimal policies for the liver wait list under a variety of settings.

Our Contributions. Our main contributions in this chapter are twofold:

- We introduce a method for informing myopic optimization about the future in dynamic problems. It automatically learns *potentials* of elements of the problem (e.g., of vertices or edges in a graph) offline, and then uses these potentials to guide myopic optimization at run time. The potential represents an estimate of how much that element can contribute to the objective in the future. The potentials can be viewed as policy parameters, so they are a natural, fairly general way of parameterizing policies. They can be optimized using a black box program; we are able to leverage state-of-the-art automated algorithm configuration tools to learn their values. Then, at run time, we simply run an offline optimization algorithm at each time period, but subtracting out in the objective the potentials of the elements used up in the solution. This causes the batch optimizer—which is traditionally myopic—to take the future into account without suffering a run-time cost.
- We successfully apply the approach to a recognized difficult dynamic matching problem, kidney exchange. It is a significantly richer matching problem than traditional matching (where the task is to pair vertices). We address the unsimplified problem. We first prove bounds on the power of learning potentials on vertices, edges, cycles, and the entire graph (which is optimal). Then, we learn potentials for vertices using a state-of-the-art parameter learning package and kidney exchange instance generator. We show that these learned potentials beget a greater number of matches than simple myopic matching at nearly no run time cost across all the settings tested.

Estimating the potential—roughly, some quantification of the ability to "help" in the future—of a structural element of a problem can be viewed through the lens of the reinforcement learning/-Markov decision processes (MDP) as a policy search process (e.g., [47, 141]). For some problems, however, estimating the value of a state is incredibly difficult. For example, in kidney exchange, determining the static value (i.e., with no lookahead) of even a reasonably small pool involves solving an integer program that cannot fit in memory [6], while estimating the value with even limited lookahead requires searching over an infinite state space and an exponential (in the number of edges) action space, with highly-variable transition probabilities between states (due to the probabilistic nature of the evolution of the pool (e.g., [75])).

For this reason, we turn to tools that have been purposefully designed to search over a large state space where the objective value is highly instance-specific and non-deterministic: automated algorithm parameter tuners [113]. Automated algorithm configuration tools are used primarily to find the set of parameters to a black box algorithm that minimizes runtime (e.g., determining the "best" set of command line parameters to pass to CPLEX or Gurobi). By viewing potentials as parameters that determine the performance of a black box optimization algorithm—the total proportion of patients matched over time by our myopic but potential-aware matching algorithm—we are able to translate our problem to one more well-studied in a different space. To our knowledge, this is the first such use of parameter tuning in dynamic optimization.

We experimentally support our technique on simulated kidney exchange problems. We show that while the theoretical worst-case performance of potentials can be quite bad, in practice the learning algorithm works well—even on relatively inexpressive structural elements (vertices). Our technique shows statistically significant increases in the number of patients matched over a wide variety of pool compositions, and scales to pools of size at least an order of magnitude larger than the previous state-of-the-art in dynamic kidney exchange.

6.2 Using *potentials* to inform myopia

We now introduce the idea of *potentials* to capture the future into myopic matching. Given a structural element (e.g., vertex, edge, cycle type, etc.) of the problem, a potential $P \in \mathbb{R}$ quantifies the future expected usefulness to the exchange of that element.

For example, consider potentials on vertices. In terms of ABO compatibility, an O-type donor can give to O-, A-, B-, and AB-type patients, an A-donor can give to A or AB, a B-donor can give to B or AB, and an AB-donor can only give to AB. Therefore, an O-type altruistic donor typically leads to more matched patients (i.e., has higher potential) than other types of altruist. Similarly, an altruist with a given blood type tends to have higher potential than a patient-donor vertex with that same donor type because the pair is harder to match, since it expects a kidney in return and that kidney needs to be compatible with the pair's patient. Intuitively, then, it might make sense to not match an O-altruist until he or she can trigger a long chain of patient-donor pairs in the pool. Similarly, if a feasible match for a hard-to-match patient-donor pair exists at the present time, it might make sense to immediately match this pair since saving the pair for later would likely yield no benefit (and would risk that pair never being matchable in the future). So, the altruist would be given a high potential: by saving the O-type altruist until she triggered a long chain, more lives would be saved overall. Similarly, the hard-to-match patient-donor pair would be given a low potential to incentivize immediate matching.

As described in the introduction, the exchange clearing problem finds a maximum-weight exchange of disjoint cycles. Given the potential for a vertex, edge, or cycle type (where type is defined by the ABO blood types of the donors and the patients in the vertices), we can easily translate this information into a language the matching algorithm understands. For example, with vertex potentials, the translation works as follows. Given vertex potentials P_X and P_Y representing the potentials of patient-donor pairs of ABO type X and Y, respectively, any edge e between vertices of type X and Y receives weight $w_e = f(P_X, P_Y)$, for some function $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$. Cycles

in the exchange are then assigned weights as usual, as the sum of their edge weights. In this way, the potentials assigned to specific elements affect the final maximum-weight exchange of disjoint cycles.

A small example of potentials. We now provide an example of setting *vertex potentials* in a reduced model of kidney exchange; note that, as described in our paper, we use a much richer set of features in our experiments. We must first select a set Θ of features representing different element types in the pool. Then, for each element type θ, assign some value $P_\theta \in \mathbb{R}$ that represents the expected potential usefulness of that kind of element to the pool over time. As an overly simplified pedagogical example, let $\Theta_{\text{ALT}} = \{\text{ALT}, \text{PAIR}\}$; potentials are assigned based on whether or not a vertex is an altruist or a patient-donor pair. Intuitively, $P_{\text{ALT}} \geq P_{\text{PAIR}}$; altruistic donors tend to be (much) easier to match because no returning edge is required to "close the cycle" at the end of a chain. These potentials are then subtracted out for each element in the objective.

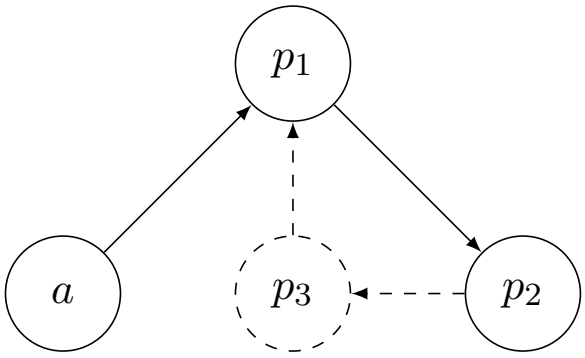


Figure 6.1: Example of potentials Θ_{ALT}. Pair p_3 appears in the second period. Myopic matching uses a to match two pairs; assigning a positive potential results in all three pairs matched without using a.

Figure 6.1 shows a two time period example under Θ_{ALT}. Vertices a, p_1, and p_2 arrive in the first time period, while vertex p_3 arrives in the second time period. Assigning a (large enough) potential P_{ALT} results in the chains $\langle a, p_1 \rangle$ and $\langle a, p_1, p_2 \rangle$ having negative weight and thus not being matched in the first time period. However, when p_3 arrives, the chain $\langle a, p_1, p_2, p_3 \rangle$ may now have positive value (i.e., the utility of matching three pairs outweighs the learned potential of holding a back for another round) and can be matched, or the 3-cycle $\langle p_1, p_2, p_3 \rangle$ has higher positive value and is matched instead, continuing to save the altruistic donor a for a longer chain in the future.

A more concrete example of ABO potentials. Returning to our initial example of learning potentials over patient and donor blood types, say we have learned that $P_{-O} = 2.1$ and $P_{O-AB} = 0.1$, representing the vertex potentials of an O-type altruist and an O-AB type patient-donor pair, respectively. Furthermore, define $f(P_X, P_Y) = 1 - \frac{1}{2}(P_X + P_Y)$. (This formula assumes that edge weights before taking potential into account are 1, as is the case in the generator due to Saidman

et al. [191], which we use in this chapter. Some fielded kidney exchanges also set all edge weights to 1, and others set them roughly equally. The methodology in this chapter applies to unequal edge weights as well—which we discuss in the more realistic experiments using potentials via FUTUREMATCH in Chapter 8.) Then, any edge e between an O-type altruist and an O-AB type patient-donor pair will receive weight $1 - 0.5(2.1 + 0.1) = -0.1$. Informally, this is telling the matching algorithm that any chain c including edge e—triggered by the extremely valuable O-type altruist—will need to be long (i.e., high weight) enough to offset the negative weight of e.

In the ABO model of kidney exchange, there are 20 possible vertex types: 4 types of altruists (O-, A-, B-, and AB-type), and 16 types of patient-donor pairs (O-O, O-A, ..., AB-AB-type). If we consider edge types, this number jumps to 244: 208 possible edges originating from patient-donor pairs, and 36 originating from altruists. Allowing potentials to be learned for variable-length cycles increases this number dramatically. Intuitively, there is a tradeoff between the expressive power of the potentials (allowing potentials for larger structural elements such as edges, or cycles, or even beyond, having more expressiveness) and the computational power needed to learn the potentials (the hypothesis space being larger the more variables there are). To lend insight to this, in the next section, we prove bounds that compare associating potentials with vertices, edges, cycles, or even higher-level elements such as the entire graph.

6.3 How much can associating potentials to larger elements help?

In this section, we prove bounds on the best-case benefit from associating potentials on larger elements compared to associating them with smaller elements. (However, as we will experimentally show later in the paper, even properly setting vertex potentials works very well in practice.)

First, in Theorem 14 we compare the application of potentials to vertices and to edges and show that edge potentials can do notably better. Second, Theorem 15 considers allowing potentials to be applied to edges and cycles; again, we show that the finer-grained resolution of cycles can allow better overall results than just edges. Finally, Theorem 16 shows that, in certain pathological cases, even applying potentials to cycles can perform poorly compared to potentials on unlimited graph elements (which is equivalent to comparing against an omniscient algorithm with perfect foresight).

The constructions in the proofs are in the two- and three-stage setting and work even if the clearing engine is omniscient about the future. To evaluate the quality of a choice of potentials, we compute the number of matched pairs as follows: in the first stages the number of matched pairs plus the potentials of leftover elements is maximized, and in the final stage the number of matched pairs is maximized.

6.3.1 Vertex potentials versus edge potentials

We first consider associating potentials only to the vertices, in the two-stage model.

Theorem 14 (Vertex potentials vs edge/graph potentials)**.** *We give results first for the cycles-only case, and second for the case with cycles and chains.*

1. *For every $k \in \mathbb{N}$ there exists an input with cycles of length at most $2k+4$ and no chains such that for any choice of vertex potentials the number of matched patients is at most a $(4k+4)/(6k+4)$-fraction of the optimum.*
2. *For every $k \in \mathbb{N}$ there exists an input with 2-cycles and chains of length at most $2k+5$ such that for any choice of vertex potentials the number of matched patients is at most a $(4k+5)/(6k+5)$-fraction of the optimum.*[1]

In each of the three cases, the construction is such that the optimum is achievable using edge potentials.

Proof. For case 1, given $k \in \mathbb{N}$, consider the input illustrated in Figure 6.2. In stage 1 there is a 2-cycle between AB-O and AB-A, a 2-cycle between A-A and A-O, k 2-cycles between AB-O and A-O, and k 2-cycles between AB-A and A-A. In stage 2 the gray vertices disappear, and $2k$ AB-AB vertices appear, so that a $(2k+4)$-cycle is formed between the white and dashed vertices. Note that an edge exists between two donor-patient pairs only if they are blood type compatible.

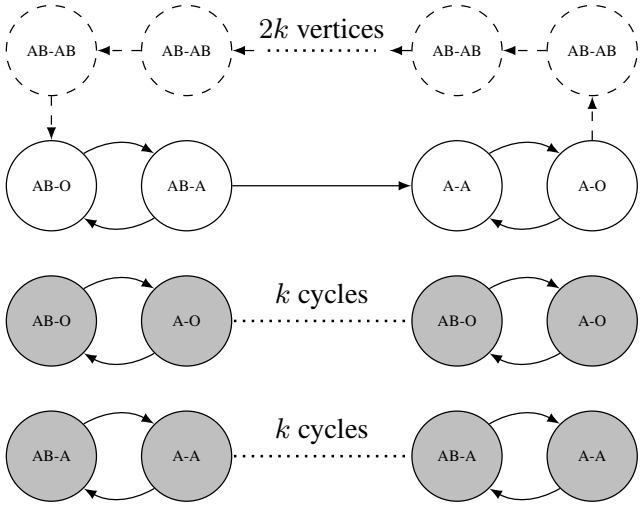


Figure 6.2: Example of Theorem 14. Vertices present in both stages are white. Vertices present only in stage 2 are dashed. Vertices present only in stage 1 are gray.

The optimal solution matches all gray cycles in the first stage, and the long cycle in the second stage, for a total of $6k+4$ pairs matched. This can easily be accomplished using edge potentials but, as we will show, not vertex potentials.

To analyze the quality of the solution when using vertex potentials, first consider the case where the 2-cycle between AB-O and AB-A and the 2-cycle between A-A and A-O both remain unmatched at the end of the first stage. Because these cycles are disjoint from other cycles in the

[1] As is common practice, we say that the last donor in a chain donates to the deceased-donor wait list (not shown in our illustrations). That is included in all our analyses and experiments. For example, if an altruist donates to a pair that donates to the wait list, the chain length is 2 and the number of patients saved is 2.

graph, they can remain unmatched only if the potential of their vertices is greater than their length; formally, $P_{\text{ABO}} + P_{\text{AB-A}} > 2$ and $P + P_{\text{A-O}} > 2$. By summing these two inequalities we obtain the inequality

$$P_{\text{ABO}} + P_{\text{AB-A}} + P + P_{\text{A-O}} > 4. \tag{6.1}$$

It follows that $P_{\text{ABO}} + P_{\text{A-O}} > 2$ or $P_{\text{AB-A}} + P > 2$. Indeed, otherwise by summing we would obtain that $P_{\text{ABO}} + P_{\text{AB-A}} + P + P_{\text{A-O}} \leq 4$, in contradiction to Equation (6.1). Assume without loss of generality that $P_{\text{ABO}} + P_{\text{A-O}} > 2$. Therefore, in stage 1 the k cycles between AB-O and A-O pairs are not matched. It follows that the number of pairs that are matched can be at most $4k + 4$.

We now consider the case where the first-round matching includes either the 2-cycle between AB-O and AB-A or the 2-cycle between A-A and A-O. In stage 2, the $2k$ AB-AB vertices in the long cycle cannot be matched. Thus, the number of matched pairs is at most $4k + 4$. We see that the ratio between the number of patients matched under optimal vertex potentials and the optimum is at most $(4k + 4)/(6k + 4)$.

The input for case 2 is almost identical, and is obtained by removing the edge from AB-AB to AB-O, and adding an altruistic donor (say with blood type O), who appears in stage 2, with an edge to the white AB-O pair. □

As k grows in Theorem 14, the ratio of vertex to edge (and optimal) potentials in parts 1 and 2 tend toward $2/3$. This is a negative result in the high-stakes world of kidney exchange, where losing $1/3$ of the possible matches is highly undesirable. However, the results below shows that worst-case performance is poor even if we allow potentials on edges.

6.3.2 Edge potentials versus cycle potentials

We now show that edge potentials can have poor performance compared to cycle potentials, tending toward matching 1/2 of the matchable patients in a two-stage model. We allow the cycle potential to be a function of all the ABO types of the vertices of the cycle.

Theorem 15 (Edge potentials vs cycle/graph potentials)**.** *As with Theorem 14, we give results first for the cycles-only case, and second for the case with cycles and chains.*

1. *For every $k \in \mathbb{N}$ there exists an input with cycles of length at most $3k + 2$ and no chains such that for any choice of edge potentials the number of matched patients is at most a $(3k + 2)/(6k + 2)$-fraction of the optimum.*
2. *For every $k \in \mathbb{N}$ there exists an input with 2-cycles and chains of length at most $3k + 3$ such that for any choice of edge potentials the number of matched patients is at most a $(3k + 3)/(6k + 3)$-fraction of the optimum.*

In both of these cases, the construction is such that the optimum is achievable using cycle potentials.

Proof. The construction in Figure 6.3 proves case 1.

Cycle potentials match the gray nodes and the long cycle, while edge potentials can match either the gray and solid white vertices or only the long cycle. In this construction, every vertex has the same ABO-type, so each edge must have the same potential P. Assume that the edge weights before potential is taken into account are all 1. Now, if $P > 1$, the algorithm must match

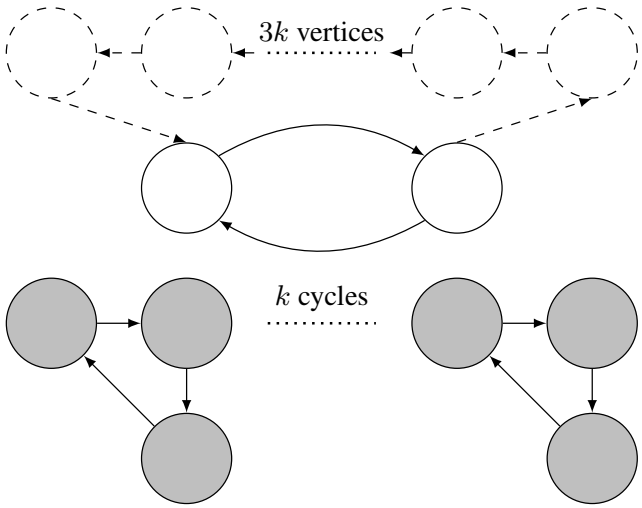


Figure 6.3: Example of Theorem 15. Vertices present in both stages are white. Vertices present only in stage 2 are dashed. Vertices present only in stage 1 are gray.

nothing in stage 1 while if $P < 1$, it must match the 2-cycle in stage 1 (along with the 3-cycles), which precludes it from matching the long cycle in stage 2. The adjustment for case 2 is the same as in Theorem 14. □

6.3.3 Cycle potentials versus graph potentials

We now show that we cannot get optimal matching in the worst case even if we associate potentials with entire cycles, in a three-stage model (where all vertices expire after the second stage).

Theorem 16 (Cycle potentials vs graph potentials)**.** *Denote by L the cap on cycle length. There exists an input with cycles of length at most L (even with no altruistic donors) such that for any choice of cycle potentials the number of matched patients is at most a $1/L$-fraction of the optimum.*

Proof. In Figure 6.4, the optimal solution is to pass on the L-cycle in stage 1, but to accept all the L-cycles except the central one in stage 2. The cycle-potential algorithm cannot accomplish that because it has to wait on all L-cycles or to take them all (since this construction assumes that all edges have the same pre-potential weight, and all L-cycles have the same potential because all the vertices have the same ABO type). It must either take the L-cycle in stage 1 or nothing in either stage; the former choice is better. □

While the theoretical results in this section show that there can be a significant benefit to associating potentials with larger elements, in the experiments that follow, we will use potentials on vertices. This is motivated by there being fewer weights to learn. As we will show, even learning that number of weights is challenging, but as we also show, the approach works well in practice.

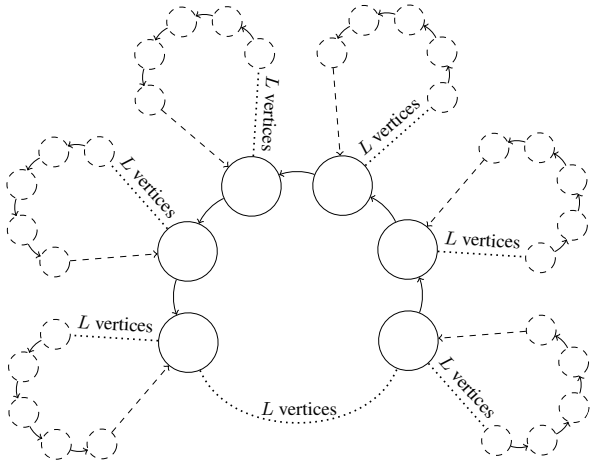


Figure 6.4: Example of Theorem 16. Vertices present in both stages are white. Vertices present only in stage 2 are dashed.

6.4 Learning the values of potentials

In this section, we describe our technique for learning potentials for elements of kidney exchange. We use ParamILS [113], which was at the time of this initial work a state-of-the-art algorithm configuration package, to intelligently search through the parameter space for an optimal (or near-optimal) instantiation of the potential variables. (In the more realistic and up-to-date experiments via FUTUREMATCH in Chapter 8, we use SMAC [114], a successor to ParamILS.) The method given is general enough to handle potentials on vertices, edges, cycles, and so on. However, due to the computational complexity of learning potentials, we focus on learning vertex potentials only.

Given an objective function and a parameter space over which to search, ParamILS takes an initial vector of parameter values and, using iterated local search, tries to optimize the objective by adjusting the parameters. In our case, the parameters are 20 real-valued vertex potentials (16 patient-donor pair ABO types plus 4 altruist ABO types). Our objective function in ParamILS was to maximize the number of patients matched as a fraction of the optimal number of patients that could be matched in a full information model. This ratio is measured by running the myopic kidney exchange solver using the vertex potentials as parameters.

ParamILS requires discretization of the parameters. We let each potential live in the space $\{0, 0.2, \ldots, 3.0\}$. (The highest value that ParamILS ended up with was 1.4.)

To learn the parameter values, ParamILS uses a training set and an internal test set. Our sets

contained 1500 and 300 kidney exchange graphs, respectively. Each graph had 95 patient-donor pairs and 5 altruists total arriving over 25 months, and was generated using the standard kidney exchange generator [191]. We amended the generator to generate altruists to correspond to fielded kidney exchanges. The expiration rate of vertices was set according to the reality that 12% of kidney patients survive 10 years [215]. The expiration rate for altruists was set to be the same. The other ParamILS settings were: 1000 runs per random seed per graph, and 1000 random graphs per parameter vector. The authors of ParamILS recommend running ParamILS multiple times with different settings of the `numRuns` parameter; we ran 24 times with different values and ran for three days on each. Finally, we chose the parameter vector with highest score across all runs.

None of the ParamILS iterated local search runs terminated. This is due to (a) the high variability in running dynamic match runs on the same compatibility graphs, (b) the high variability between different compatibility graphs, and (c) the long run time required to solve each instance (the clearing problem is NP-hard). However, as our results in the next section show, the learned weights did, in fact, improve myopic matching significantly. Also, the relative sizes of the learned parameter values made sense, with easier-to-match vertices receiving greater potentials.

6.5 Weighted myopic matching at run time

We now present extensive experimental results using the potentials learned on our training and ParamILS-internal test data sets. We test on a new test set starting with the problem sizes used for training and then testing on much larger instances. We conclusively show that the learned vertex potentials increase the total number of matches made by the myopic matching algorithm. Furthermore, once these potentials are learned, they do not seriously affect match run time. As such, the scalability results of, e.g., the HPIEF and PICEF models of Chapter 3 remain applicable.

6.5.1 Comparing to optimal matching

We begin by comparing our weighted myopic algorithm and standard myopic algorithm to an optimal matching. The optimal matching is computed in a full-information model where the optimization algorithm is given access up front to exactly which patient-donor pairs and altruists will be in the pool at each period. It is impossible for any algorithm to match more candidates—and unlikely that any would be able to match equally many due to lack of full information about the future.

As in prior experiments on dynamic kidney exchange, each simulation was conducted over 51 time periods, representing 4 years and 3 months of actual time. We vary the number of patient-donor pairs from 110 to 710 and add 5% as many altruists as there are patient-donor pairs. For example, with 510 pairs, there are 25 altruists in the pool.[2] This number is motivated by the UNOS pilot nationwide kidney exchange that we have been working to establish. Altruistic donors were incorporated into that exchange in April 2011. At the time of these experiments, their number had been roughly 5% of the number of patient-donor pairs—that percentage has dropped a bit in recent

[2]At each time period, the number of pairs and number of altruists are drawn from a normal distribution; this is retried until the total number of each of the two across all 51 time periods is correct.

years due to the increase in match cadence using up altruistic donors quickly. The absolute number of altruists entering has increased.

Figure 6.5 shows the improvement gained by our weighted myopic matching algorithm, relative to the difference between plain myopic matching and the optimal match size. For example, if the standard myopic algorithm matched 300 candidates, the optimal matched 360, and the weighted myopic algorithm matched 315, we report $(315 - 300)/(360 - 300) = 25\%$. Clearly, vertex potentials help. Interestingly, as the number of patient-donor pairs and altruists increases, the weighted myopic algorithm tends toward the optimal solution more quickly than standard myopic matching.

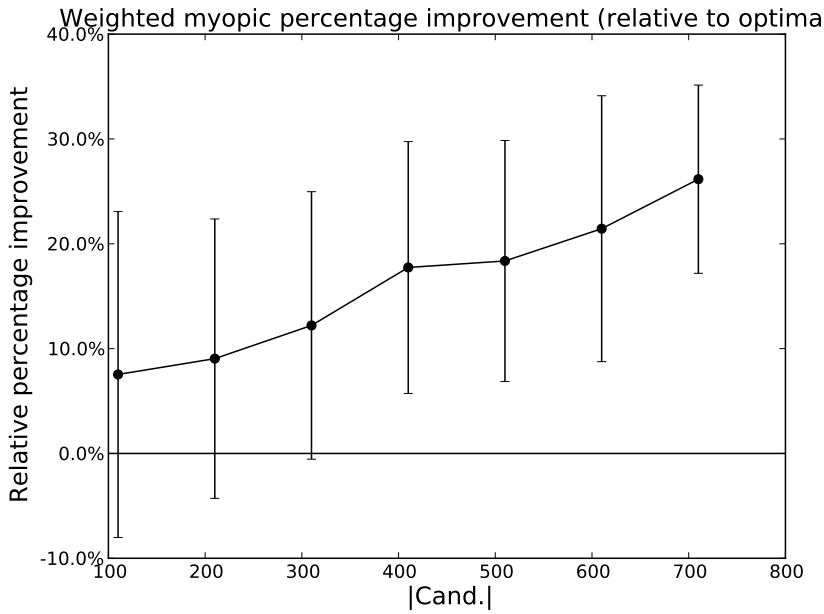


Figure 6.5: Percentage gain of weighted over unweighted myopic matching relative to the optimal match cardinality.

Table 6.1 gives the statistical significance of our results. Statistical significance testing was done using a Wilcoxon signed-rank test, a more conservative, nonparametric version of a t-test. The p-values clearly show the significance of the weighted matching algorithm's gains.

6.5.2 Scaling to larger graphs

We now study how the approach scales to larger problems. Because the parameter learning was complex already at a smaller problem size, we do not attempt to re-learn the parameters for the larger problems.

Furthermore, calculating the optimal match size quickly becomes intractable, since the clearing algorithm must consider an unrealistically large pool over all 51 time periods at once. Therefore,

patient-donor pairs	%improvement	samples	p-value
110	**7.54**	240	2.970e-204
210	**9.04**	240	6.240e-40
310	**12.21**	240	4.515e-37
410	**17.74**	240	5.994e-41
510	**18.36**	239	1.468e-41
610	**21.44**	66	1.400e-12
710	**26.16**	15	5.320e-04

Table 6.1: Statistical significance tests of our results for graphs with a variable number of patient-donor pairs and 5% as many altruistic donors. The %improvement over unweighted is relative to the optimal matching.

we only evaluate the efficacy of our algorithm against the myopic matching algorithm. We maintain the test setup from the last section, with 5% as many altruists as there are patient-donor pairs.

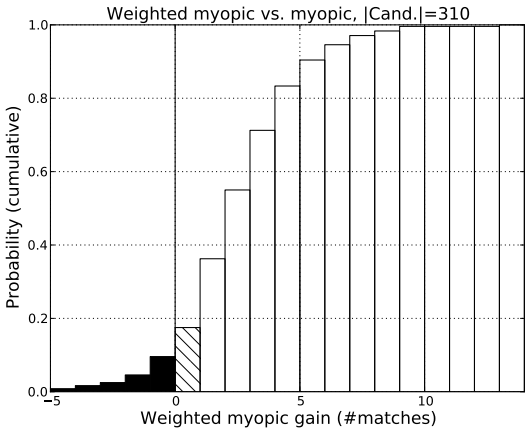
Figure 6.6 gives sample cumulative distribution functions of the weighted myopic gains over unweighted myopic in terms of total match size. White bars correspond to the weighted algorithm matching more than unweighted. Black bars represent when our method was beaten by unweighted myopic. Losses such as these are unavoidable in any non-full-information model; however, in our experiments losses were rare and significantly smaller than respective gains.

Figure 6.7 shows the percentage gain of the weighted myopic algorithm relative to the match cardinality of the unweighted myopic matching algorithm. While these percentages are relatively small, it is important to note that (1) the full-information upper bound on the amount of room for improvement is rather low and our algorithm captures a large percentage of that (Figure 6.5), (2) these are improvements in real lives saved, so even small improvements are important, and (3) the improvements are statistically significant as discussed above. Furthermore, some of the decline in absolute percentage gain may be due to our learning vertex potentials on significantly smaller graphs.

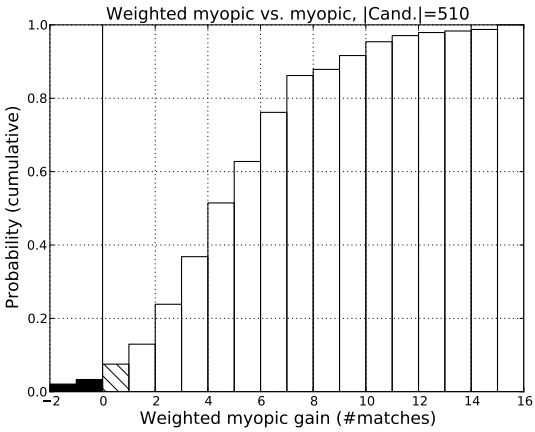
6.6 The role of altruistic donors

One significant unknown in kidney exchange is the ratio of altruistic donors to patient-donor pairs. More altruists in the pool drastically increases the fraction of patients matched (see, for example, Figure 2.6 of Chapter 2). In this section, we vary the percentage of altruists in the pool relative to the number of patient-donor pairs. It turns out that our method works well when there are no altruists, few altruists, or many altruists. This is despite the fact that we do not re-learn potentials for a specific density of altruists.

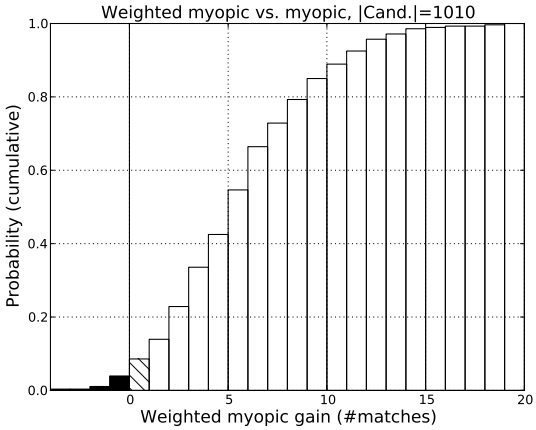
For these experiments, we held the number of patient-donor pairs in the pool constant at 510, and varied the number of altruists from 0 to 120. Figure 6.8a shows the absolute gain in the number of candidates matched by our weighted myopic algorithm over the standard myopic algorithm. As above, the absolute gain is 1–1.5%. Interestingly, the absolute gain percentage decreases as the



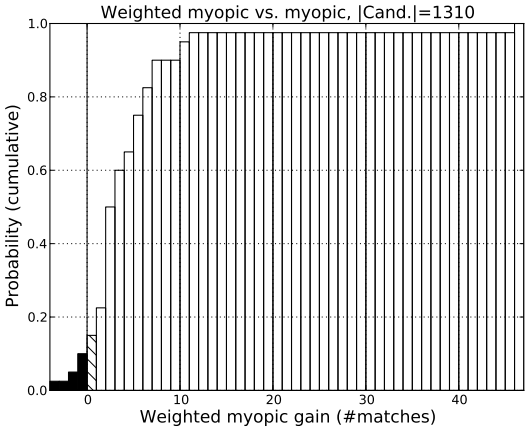
(a) #Candidates=310



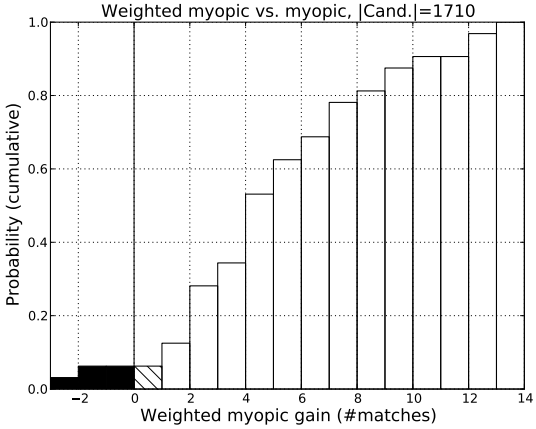
(b) #Candidates=510



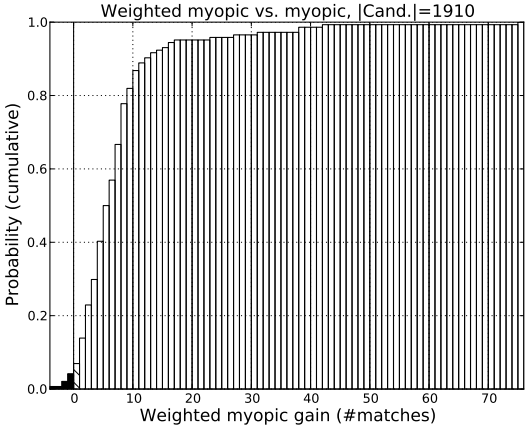
(c) #Candidates=1010



(d) #Candidates=1310



(e) #Candidates=1710



(f) #Candidates=1910

Figure 6.6: CDFs of absolute gains of weighted versus unweighted myopic matching. White bars correspond to the weighted approach outperforming the unweighted.

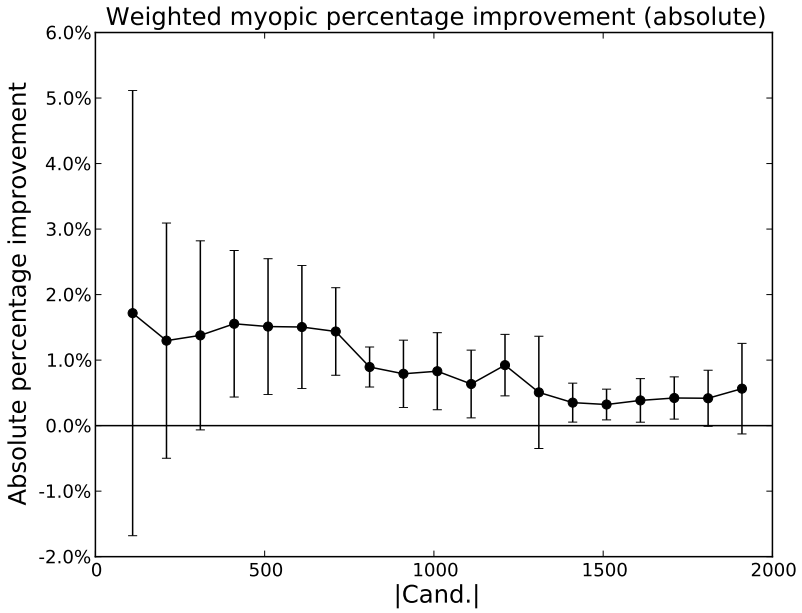
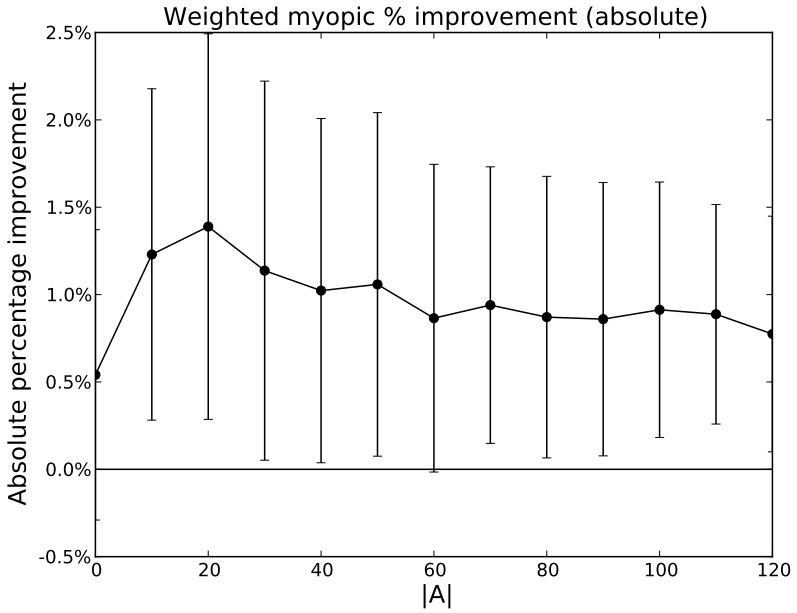


Figure 6.7: Percentage gain of weighted over unweighted myopic matching relative to the myopic match cardinality, for graphs with a variable number of patient-donor pairs and 5% as many altruistic donors.

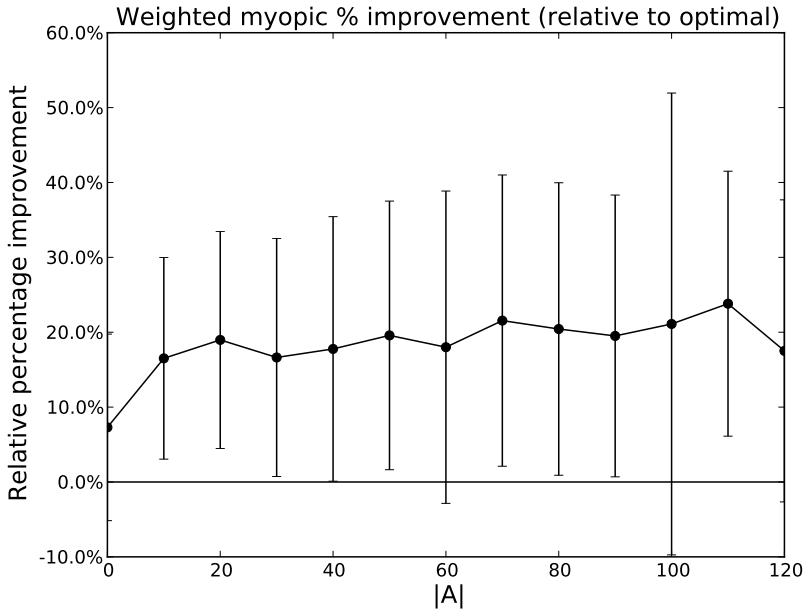
number of altruists increases; this can be explained by myopic matching being closer to optimal when the number of altruists is large. Figure 6.8b shows the percentage gain of weighted myopic matching relative to the gap between optimal matching and traditional myopic matching. We see that using the vertex potentials learned earlier results in matching 10–25% of the candidates left unmatched by unweighted myopic matching, *regardless of the number of altruists in the pool*.

Table 6.2 shows that the vertex potentials improve the matching with extremely high statistical significance. Again, Wilcoxon signed-rank test methodology was used.

Potentials on altruists only. We also ran experiments where we learned and tested potentials on altruists only. This significantly reduces the search space size to 4 parameters. ParamILS learned the following potentials for O-, A-, B-, and AB-altruists, respectively: 0.8, 0.6, 0.4, and 0.2. This makes qualitative sense because O-altruists are easiest to match and AB-altruists the hardest. Interestingly, the potential is less than 1 even for O-altruists. Perhaps surprisingly, this approach led only to a tiny improvement over unweighted myopic matching across graph sizes. For example, averaged over 1500 runs with 410 vertices each, the relative gain over unweighted myopic was 0.06% while it was 16.3% when learning all 20 potentials (over patient-donor pairs and altruists).



(a) Absolute improvement.



(b) Improvement relative to optimal.

Figure 6.8: Percentage gain of weighted myopic over unweighted myopic matching relative to the myopic match cardinality (top) and optimal match cardinality (bottom), for graphs with 510 pairs and a varying number of altruists.

altruists	%improvement	samples	p-value
0	**7.29**	369	2.507e-43
10	**16.51**	280	4.759e-45
20	**18.97**	280	1.304e-44
30	**16.63**	280	1.572e-40
40	**17.77**	280	1.863e-38
50	**19.58**	280	8.005e-42
60	**18.00**	280	1.906e-37
70	**21.55**	280	4.829e-44
80	**20.43**	280	2.407e-41
90	**19.50**	270	1.763e-41
100	**21.10**	160	5.974e-26
110	**23.81**	65	3.552e-12
120	**17.51**	49	1.769e-09

Table 6.2: Statistical significance tests for graphs with 510 patient-donor pairs and varying number of altruistic donors. The %improvement over unweighted is relative to the optimal matching.

6.7 Conclusions & future research

In this chapter, we introduced an automated, scalable method for informing myopic optimization algorithms about the future in dynamic problems. It learns *potentials* of elements of the problem offline and then uses the potentials to guide myopic optimization at run time. The potential represents an estimate of how much that element can contribute to the objective in the future. The potentials can be viewed as policy parameters, so they are a natural, fairly general way of parameterizing policies. They can be optimized using a black box program; we were able to leverage state-of-the-art automated algorithm configuration tools to learn their values. Then, at run time, we simply run an offline optimization algorithm at each time period, but subtracting out in the objective the potentials of the elements used up in the solution. This causes the batch optimizer—which is traditionally myopic—to take the future into account without suffering a run-time cost.

We applied these techniques to kidney exchange. We theoretically compared the power of using potentials on increasingly large elements: vertices, edges, cycles, and the entire graph. Then, experiments showed that by learning vertex potentials, our algorithm matches more patients than the current practice of clearing myopically—at nearly no run-time cost. We experimented with a variety of graph types; weighted myopic matching helped on them all. Furthermore, applying potentials did not significantly affect the runtime of the clearing algorithm, so scalability results from the static kidney exchange problem carry over to potentials-weighted myopic matching.

A clear direction for future research would be an empirical comparison of potentials learned on vertices to larger elements. There is a tradeoff: on the one hand, as we proved, associating potentials with larger elements has more power, but on the other hand, there are more parameters to learn. ParamILS did not converge even on vertex potentials, and neither did its successor

SMAC, which we will use in Chapter 8 for more realistic simulation experiments on UNOS-style data via the FUTUREMATCH framework; we conjecture that a new learning method is required to move beyond vertex potentials. This could involve using domain knowledge of partial ordering of potentials. Furthermore, better automated methods for feature selection and classifier selection could be employed; this is an active area of research in knowledge discovery and data mining [208]. Indeed, recent scalability advances in reinforcement learning may be applicable to this problem [142, 161, 217], and are worth exploring in future work.

PART III:

Balancing Equity & Efficiency in Dynamic Matching Environments

The dilemma over whether to pursue policies that emphasize equity (sometimes regarded as "fairness") or utilitarianism ("total good") faces all societies. Such policies are often in conflict ...

– Hooker & Williams

The primary purposes of the OPTN are to operate and monitor an equitable system for allocating organs donated for transplantation ...

– OPTN Charter

7

The price of fairness in kidney exchange

In Parts I and II, we address the clearing of exchanges under utilitarian matching rules. This approach marginalizes certain classes of participants (in our case, patients) in an exchange. In this chapter, we focus on improving access to kidneys for hard-to-match agents; in the kidney exchange context, these are typically *highly-sensitized* patients. Toward this end, we formally adapt a recently introduced measure of the tradeoff between fairness and efficiency—the *price of fairness*—to the standard kidney exchange model. We show that the price of fairness in the standard theoretical model is small. We then introduce two natural definitions of fairness and empirically explore the tradeoff between matching more hard-to-match patients and the overall utility of a utilitarian matching, on real data from the UNOS nationwide kidney exchange and simulated data from each of the standard kidney exchange distributions.

As discussed in the motivation of failure-aware matching in Chapter 5, successful transplantation of a kidney relies on tissue-type compatibility between the donor organ and patient, among other noisy factors. Compatibility is determined through a tissue-type *crossmatch* between a potential donor and patient's blood; if the two types differ substantially, the patient's body will reject the donor organ. Some patients are *highly-sensitized*; there is a very low probability that their blood will pass a crossmatch test with a random organ. For these patients, finding a kidney is quite difficult (and median time on the waiting list jumps by a factor of three over less sensitized patients [212]).

Roughly 17% of the adult patients on the waiting list for *deceased* donor kidneys are highly-sensitized [107]. Yet, the percentage of highly-sensitized patients in fielded kidney exchanges is quite high; roughly 60% of the UNOS nationwide kidney exchange is highly-sensitized, as shown in Figure 7.1. This is due to kidney exchanges often being seen as a "last hope" for patients who have not been matched through more traditional means.

Roth et al. [186] and Li et al. [146] investigate egalitarianism in the context of 2-cycle-only exchange, with extensions to longer cycles by Yılmaz [221], but so far implementable fairness

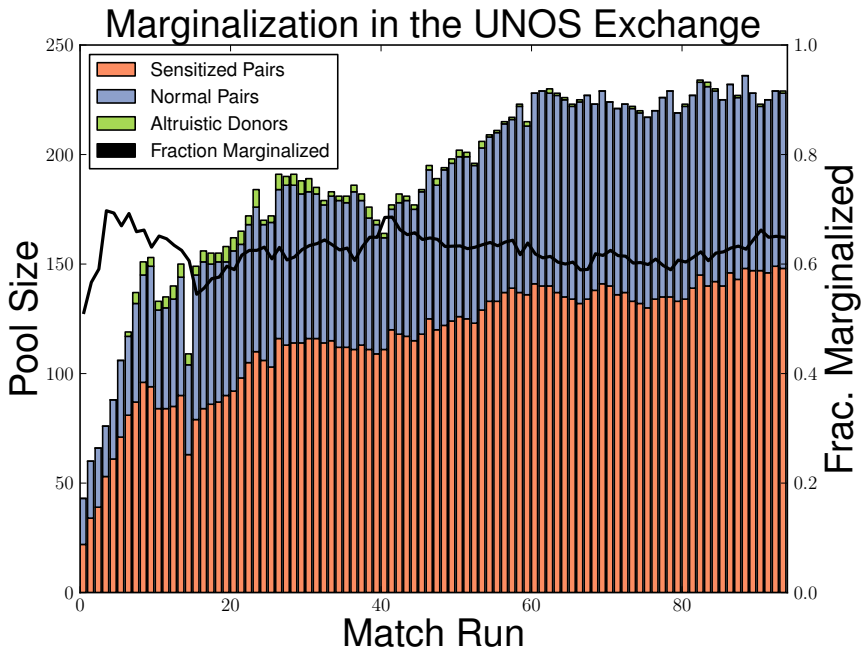


Figure 7.1: Composition of the UNOS national kidney exchange over time. For each of 96 match runs (x-axis), the raw number of highly-sensitized patients, non-highly-sensitized patients, and altruists are plotted (left y-axis), as well as the percentage of patients who are highly-sensitized as a percentage of the pool size (right y-axis).

rules in the context of realistic exchanges with longer cycles and chains has not been rigorously discussed. In this chapter, we explore the *price of fairness* in kidney exchange—the relative loss in total welfare from using a "fair" matching rule, instead of an overall utility-maximizing one [35]. Theoretically, we show that the price of fairness is small in the standard theoretical kidney exchange model. We then define two natural definitions of fairness in kidney exchange and empirically quantify the tradeoff between efficiency and fairness on real and simulated data. We find that, on real data, prioritizing hard-to-match patients results in a price of fairness that is (often quite far from) zero. While this chapter focuses on highly-sensitized patients, its techniques and results are easily adaptable to other notions of fairness in kidney exchange.

Related Publications

Some of the work in this chapter appeared at AAMAS-14 and its workshops; it was performed as a collaboration between Dickerson, Procaccia, and Sandholm [77, 78].

7.1 Fairness in kidney exchange

In this section, we extend the standard graph-based model of kidney exchange to include a notion of sensitization, and formally define the price of fairness in the context of kidney exchange.

A model of kidney exchange with sensitization. Given the set of all legal matchings $\mathcal{M}$, the general clearing problem in kidney exchange is to find a matching M^* that maximizes some utility function $u : \mathcal{M} \to \mathbb{R}$. Formally:

$$M^* = \arg\max_{M \in \mathcal{M}} u(M)$$

As discussed in Chapter 3, in fielded kidney exchanges, one typically finds the maximum weighted cycle cover (i.e., $u(M) = \sum_{c \in M} \sum_{e \in c} w_e$). This *utilitarian* objective can favor certain classes of patient-donor pairs while marginalizing others. We formalize this notion next.

In practice, the sensitization level of a patient is presented as a Calculated Panel Reactive Antibody (CPRA) score that varies from 0 to 100. The CPRA score is an estimate of the percentage of donors that are tissue-type incompatible with the patient (i.e., the percentage of donors with whom a patient would have a positive, or failing, crossmatch).

Assume each non-altruistic vertex v has a sensitization level $v_s \in [0, 100]$, representing the CPRA level of v's patient. Altruistic vertices need no CPRA specification because they have no associated patient. Let $\tau \in [0, 100]$ represent a threshold delimiting low from high sensitization; in practice, $\tau \geq 80$. Partition V into $\{V_L \cup V_H \cup A\}$, such that:

- V_L is not highly-sensitized: $\{v \mid v \in V \setminus A \wedge v_s < \tau\}$
- V_H is highly-sensitized: $\{v \mid v \in V \setminus A \wedge v_s \geq \tau\}$
- A are altruistic donors (with no patients)

Since the highly-sensitized patients will, by definition of CPRA, have fewer incoming edges on average than lowly-sensitized patients, one worries that a mechanism maximizing overall efficiency might favor easier-to-match vertices in V_L to the detriment of those in V_H. Similarly, if a mechanism prioritizes harder-to-match vertices in V_H, one worries that the overall efficiency of the matching might drop.

The price of fairness. Bertsimas, Farias, and Trichakis recently defined the *price of fairness* to be the "relative system efficiency loss under a fair allocation assuming that a fully efficient allocation is one that maximizes the sum of [participant] utilities [35]." Caragiannis et al. defined an essentially identical concept in parallel [53]. We adopt that notion here.

Let $u_f : \mathcal{M} \to \mathbb{R}$ be a *fair* utility function. Formally, a utility function is fair when its corresponding optimal match M_f^* is viewed as fair, where M_f^* is defined as:

$$M_f^* = \arg\max_{M \in \mathcal{M}} u_f(M)$$

Given a fair utility function u_f and the utilitarian utility function u, the price of fairness is defined to be:

$$\mathsf{POF}(\mathcal{M}, u_f) = \frac{u\left(M^*\right) - u\left(M_f^*\right)}{u\left(M^*\right)}$$

That is, $\mathsf{POF}(\mathcal{M}, u_f)$ is the relative loss in match efficiency (from the utilitarian point of view u) due to the maximization of a fair utility function u_f over some family of matchings $\mathcal{M}$.

In the next section, we show that the price of fairness in the standard dense *theoretical* model for kidney exchange is quite small, for any reasonable fair utility function u_f. Then, in Sections 7.3 and 7.4, we present two families of fair utility functions motivated by our experiences with the UNOS national kidney exchange; on *real* data, the price of fairness is frequently far from zero.

7.2 The (theoretical) price of fairness is low

In this section, we give bounds for the price of fairness under the standard model of kidney exchange. The price of fairness is upper-bounded by a small number of vertices (under reasonable assumptions, with high probability).

7.2.1 Upper bound over all fair utility functions

Different notions of fairness may result in more (or less) of an effect on overall system efficiency. As an extreme example, forcing a matching to include at least one highly-sensitized patient (if possible) intuitively restricts the solution space less than forcing a matching to include as many highly-sensitized patients as possible. We must consider this when stating theoretical bounds on the price of fairness.

We derive our bound under the fair utility function $u_{H \succ L}$ that lexicographically ranks any highly-sensitized vertex over any lowly-sensitized vertex. For any matching $M \in \mathcal{M}$, let $M_H = M \cap V_H$ be the subset of highly-sensitized vertices matched by M. Formally:

$$u_{H \succ L}(M) = \begin{cases} u(M) & \text{if } |M_H| = \max_{M' \in \mathcal{M}} |M'_H| \\ 0 & \text{otherwise} \end{cases}$$

This utility function gives nonzero weight only to those matches that include the maximum possible number of highly-sensitized patients. We informally argue that price of fairness guarantees on $u_{H \succ L}$ are upper bounds to the price of fairness of any "reasonable" fair utility function. Indeed, any utility function that does not first maximize the number of highly-sensitized pairs matched will leave a thicker remaining market in which non-highly-sensitized pairs have more options for matching—and thus the resulting match will see less of an efficiency loss.

7.2.2 Model with ABO-blood types and two levels of sensitization

As in Chapter 2, and later in parts of Chapter 9, we work in a dense theoretical model that considers blood types—but augment it now to include a notion of sensitization as well.

We draw random graphs in accordance with the canonical method introduced initially by Ashlagi and Roth [19]. As in Chapter 2, partition the n incompatible patient-donor pairs of some large, directed compatibility graph $G(n)$ into $V^{X\text{-}Y}$ of type X-Y, for each combination of blood types X and Y of the patient and donor respectively. When required, we will further partition each set $V^{X\text{-}Y}$ into $V_L^{X\text{-}Y}$ and $V_H^{X\text{-}Y}$, the lowly- (highly-)sensitized pairs of type X-Y. The frequency of each blood type X is denoted by μ_X. Note that vertices with blood type-compatible patient-donor pairs may still enter the pool due to tissue-type incompatibility, as in Ashlagi and Roth [19]. We

assume that a donor and a lowly-sensitized (highly-sensitized) patient who are blood type compatible are tissue-type incompatible with constant probability p_L (p_H). Let $\bar{p}$ represent the average level of sensitization in the pool; that is, $\bar{p} = \lambda p_L + (1-\lambda)p_H$, where λ is the fraction of the pool that is lowly-sensitized.

Proposition 6 gives a bound on the price of fairness in random graphs in the ABO-model parameterized in a realistic way for a steady-state exchange (i.e., $\bar{p}$ mirrors that of a dense kidney exchange pool, and the blood type distribution mimics that of the US population). The proposition and proof sketch build on the efficiency result presented in §5 of Ashlagi and Roth [19]. Like their work, Proposition 6 considers a model without chains.

Proposition 6. *Assume that $\bar{p} < 2/5$, $\mu_O < 3\mu_A/2$, and $\mu_O > \mu_A > \mu_B > \mu_{AB}$, and $\lambda \geq (1-\bar{p})$. Denote by $\mathcal{M}$ the set of matchings in $G(n)$. Then, almost surely as $n \to \infty$,*

$$\mathit{POF}(\mathcal{M}, u_{H \succ L}) \leq \frac{2}{33}.$$

(And this is achieved using only cycles of length at most 3.)

Proof. Let pairs of type A-B and B-A be called *reciprocal* pairs. Call any non-reciprocal pair X-Y whose donor is not ABO-compatible with its patient an *under-demanded* pair, and any pair X-Y such that $X \neq Y$ ($X = Y$) and whose donor is ABO-compatible with its patient an *over-demanded* (*self-demanded*) pair.

From the results of Ashlagi and Roth [19], with high probability there exists an efficient matching M^* in $G(n)$ that matches all over- and self-demanded pairs, as well as all reciprocal pairs. Trivially, all highly-sensitized pairs are also matched in these specific subgroups. Thus, we need only consider highly-sensitized pairs in under-demanded pools. We do this exhaustively.

- In M^*, all pairs in $V^{\text{AB-B}}$ are matched to as many pairs in $V^{\text{B-AB}}$ as possible. If $|V^{\text{AB-B}}| \geq |V_H^{\text{B-AB}}|$, then all highly-sensitized under-demanded B-AB pairs will be matched. Note $|V^{\text{AB-B}}| \propto \bar{p}\mu_{\text{B}}\mu_{\text{AB}}$ and $|V^{\text{B-AB}}| \propto (1-\lambda)\mu_{\text{B}}\mu_{\text{AB}}$. Then by the assumption on λ, $\bar{p}\mu_{\text{B}}\mu_{\text{AB}} \geq (1-\lambda)\mu_{\text{B}}\mu_{\text{AB}}$, and the above cardinality inequality holds almost surely.[1]
- We now match all highly-sensitized pairs in $V_H^{\text{O-A}}$ and $V_H^{\text{O-B}}$ to pairs in $V^{\text{A-O}}$ and $V^{\text{B-O}}$, respectively, using 2-cycles. This can be done by the same argument as above.
  WLOG, assume $|V^{\text{B-A}}| \leq |V^{\text{A-B}}|$. Then, in M^*, all reciprocal pairs in $V^{\text{B-A}}$ are matched to as many pairs as possible in $V^{\text{A-B}}$, leaving $V_r^{\text{A-B}}$ leftover reciprocal pairs. In M^*, these leftover pairs in $V_r^{\text{A-B}}$ are fully matched through $V^{\text{B-O}}$ and $V^{\text{O-A}}$ in 3-cycles. By executing the 2-cycles above, we may prematurely exhaust $V^{\text{B-O}}$ and prevent up to $|V_r^{\text{A-B}}|$ 3-cycles from executing. (Note that if $V^{\text{O-A}}$ is exhausted, the $V^{\text{A-B}}$ through $V^{\text{B-O}}$ cycle can be closed at length 2 for no efficiency loss.)
  By Lemma 5.1 of Ashlagi and Roth [19], the absolute difference in reciprocal pairs is $o(n)$; that is, $|V_r^{\text{A-B}}|$ is less than any other subgroup in the pool (since the size of any other subgroup is linear in n). Therefore, the *number* of 3-cycles lost is sub-linear in n; thus, this term (relative to price of fairness) is insignificant as n grows.

[1] As $n \to \infty$, the size of a set will be very close to its expectation.

- Next, we match all pairs in $V_H^{\text{A-AB}}$ to over-demanded pairs in $V^{\text{AB-A}}$; this leaves $|V_L^{\text{A-AB}}| \propto \lambda\mu_A\mu_{\text{AB}}$ remaining A-AB pairs.
  In the efficient matching M^*, all over-demanded pairs in $V^{\text{AB-O}}$ are matched through 3-cycles with pairs in $V^{\text{A-AB}}$ and $V^{\text{O-A}}$. If the remaining $|V_L^{\text{A-AB}}|$ vertices in $V^{\text{A-AB}}$ are exhausted before $V^{\text{AB-O}}$, then the remaining vertices in $V^{\text{AB-O}}$ can still be exhausted through 2-cycles to, e.g., $V^{\text{O-A}}$ at no efficiency loss. (Note $|V_L^{\text{O-A}}| \propto \lambda\mu_{\text{O}}\mu_{\text{A}} > \lambda\mu_{\text{O}}\mu_{\text{AB}} \geq (1-\bar{p})\mu_{\text{O}}\mu_{\text{AB}} > \bar{p}\mu_{\text{O}}\mu_{\text{AB}} \propto |V^{\text{AB-O}}|$, since $\bar{p} < \frac{2}{5} < \frac{1}{2}$.) This occurs unless the fair matching accrues efficiency loss through highly-sensitized, under-demanded O-AB pairs, as explained next.
- Finally, we are left with highly-sensitized pairs in $V^{\text{O-AB}}$, the hardest to match group. In the efficient allocation, $V^{\text{O-AB}}$ is unmatched entirely. In a matching under $u_{H \succ L}$, at most $|V_H^{\text{O-AB}}| \propto (1-\lambda)\mu_{\text{O}}\mu_{\text{AB}}$ could find a matching elsewhere, possibly cannibalizing a 3-cycle through $V^{\text{AB-O}}$ to form a 2-cycle, or through $V^{\text{AB-B}}$ and $V^{\text{B-O}}$ or $V^{\text{AB-A}}$ and $V^{\text{A-O}}$ to form a 3-cycle at a cost of two 2-cycles. In any of these three (exhaustive) cases, every match of a highly-sensitized O-AB pair results in an efficiency loss of one pair, or an overall absolute efficiency loss proportional to $(1-\lambda)\mu_{\text{AB}}\mu_{\text{O}}$ pairs.

This exhausts all highly-sensitized pairs in the under-demanded subgroups not fully matched by the efficient allocation M^*. All highly-sensitized pairs in self- and over-demanded subgroups are still matched, as in the efficient allocation. Any newly unmatched reciprocal pairs are lowly-sensitized. Thus, we have matched all highly-sensitized pairs in the pool. This came at an absolute loss proportional to at most $(1-\lambda)\mu_{\text{AB}}\mu_{\text{O}} \leq \bar{p}\mu_{\text{AB}}\mu_{\text{O}}$ pairs. Figure 7.2 visualizes this new matching (and shows where losses occur with respect to the efficient matching M^*).

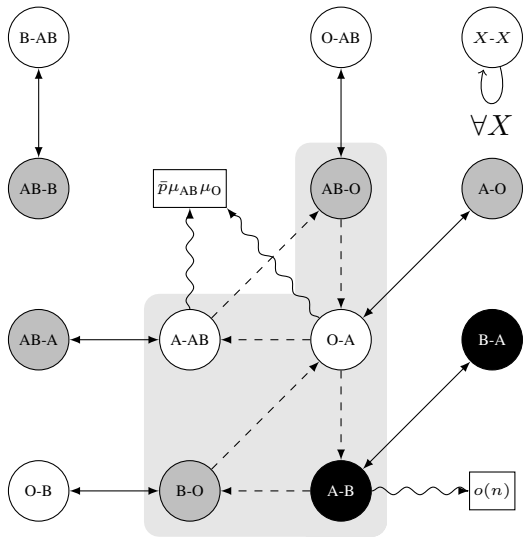


Figure 7.2: An example matching used in Proposition 6. Patient-donor pairs are ovals: under- and self-demanded pairs are white, over-demanded pairs are gray, and reciprocal pairs are black. Regular edges appear in the efficient matching, while dashed edges represent 3-cycles from the efficient matching that may be disturbed via fair matching. Efficiency loss is denoted with rectangular nodes.

To determine a bound on the price of fairness, we must first determine the actual expected loss in number of vertices (through normalization), and then determine the relative loss with respect to the expected size of the efficient matching M^*.

The expected size of M^* is at least E, such that:

$$E \propto \bar{p}\,[2\mu_{\text{AB}}\mu_{\text{B}} + 2\mu_{\text{AB}}\mu_{\text{A}} + 3\mu_{\text{AB}}\mu_{\text{O}} + 2\mu_{\text{A}}\mu_{\text{O}} + 2\mu_{\text{B}}\mu_{\text{O}} + \mu_{\text{O}}^2 + \mu_{\text{A}}^2 + \mu_{\text{B}}^2 + \mu_{\text{AB}}^2] + 2\mu_{\text{A}}\mu_{\text{B}}$$

This value is computed by counting only 2- and 3-cycles that are almost surely guaranteed to execute in the efficient matching. Then, the expected size of a fair matching under $u_{H\succ L}$ is at least $F \propto E - \bar{p}\mu_{\text{AB}}\mu_{\text{O}}$, as calculated above.

An upper bound on the expected price of fairness is then $\frac{E-F}{E}$:

$$\begin{aligned}\mathsf{POF}(\mathcal{M}, u_{H\succ L}) \leq (\bar{p}\mu_{\text{AB}}\mu_{\text{O}}) \,/\, \\ (\bar{p}\,[2\mu_{\text{AB}}\mu_{\text{B}} + 2\mu_{\text{AB}}\mu_{\text{A}} + 3\mu_{\text{AB}}\mu_{\text{O}} + 2\mu_{\text{A}}\mu_{\text{O}} \\ + 2\mu_{\text{B}}\mu_{\text{O}} + \mu_{\text{O}}^2 + \mu_{\text{A}}^2 + \mu_{\text{B}}^2 + \mu_{\text{AB}}^2] + 2\mu_{\text{A}}\mu_{\text{B}})\end{aligned}$$

Note that the notion of proportionality can be dropped, since both the numerator (the size of the fair matching, written in terms of the size of the efficient matching) and the denominator (the size of the efficient matching) are normalized by the same constant.

We prove the statement by giving an upper bound on the right-hand side of the inequality above. Considering the distribution constraint $\sum_{X\in\{\text{O,A,B,AB}\}} \mu_X = 1$, and the ordering constraints on each blood type, this is upper-bounded when $\bar{p} = \frac{2}{5}$, $\mu_{\text{O}} = \frac{1}{3}$, and $\mu_{\text{AB}} = \frac{2}{9}$ (which forces $\mu_{\text{A}} = \mu_{\text{B}} = \frac{2}{9}$). Using these values yields an upper-bound of $\mathsf{POF}(\mathcal{M}, u_{H\succ L}) \leq \frac{2}{33}$ (with high probability). □

Different countries and regions within countries have different blood type distributions. From Proposition 6, a more realistic bound for this model can be drawn from the United States distribution of blood types ($\mu_{\text{O}} \approx 0.44$, $\mu_{\text{A}} \approx 0.42$, $\mu_{\text{B}} \approx 0.10$, $\mu_{\text{AB}} \approx 0.04$); this yields a bound of $\mathsf{POF}(\mathcal{M}, u_{H\succ L}) \lesssim 1.5\%$.

A symmetric result follows if $\mu_{\text{B}} > \mu_{\text{A}}$, instead of $\mu_{\text{A}} > \mu_{\text{B}}$ as assumed. We note that, from the reasoning in [19], similar results can be derived for $\bar{p} > 2/5$—so long as p_L, p_H, and $\bar{p}$ remain constant.

7.3 How should we define fairness?

In this section, we present two definitions of fairness in kidney exchange—one using strict lexicographic preferences and the other using a sliding scale weighting function. Critically, we feel both definitions fit within the scope and practice of current policy in fielded exchange, a necessary consideration when fielding new technology in medicine (as noted by those who designed the recent deceased donor allocation scheme [37] and supported by our experience with the UNOS exchange). We briefly discuss challenges in implementing either fairness rule, then experimentally validate both rules in Section 7.4 on simulated and real exchange data.

7.3.1 Lexicographic fairness

First, we consider a fairness criterion that assumes lexicographic preferences over classes of vertices. This is a generalization of the $u_{H \succ L}$ utility function used in Section 7.2. Informally, this fairness rule allows policymakers to state a hard preference for matchings containing a baseline percentage of highly-sensitized patients and, if and only if this constraint is fulfilled, express a secondary utilitarian preference. We present this rule below in the context of both deterministic and probabilistic kidney exchange.

Deterministic model

Let α^* be the maximum fraction of vertices in V_H that could be matched over $\mathcal{M}$, the set of possible matchings on some compatibility graph G. That is,

$$\alpha^* = \left(\max_{M' \in \mathcal{M}} |M'_H| \right) / |V_H|$$

For a given graph G, α^* can be computed using a simple modification of the standard kidney exchange cycle formulation integer program, discussed in Part I. (We note that the PIEF, HPIEF, and PICEF models of Chapter 3 can be used in place of the cycle formulation via a similar tweak to just their objective functions; for expositional ease, we present this section in the context of the basic cycle formulation.)

$$\begin{array}{rcccll} \alpha^* = & 1/|V_H| & \cdot & \max & \sum_c h(c) x_c & \\ & & & \text{s.t.} & \sum_{c|v \in c} x_c \leq 1 & \forall v \\ & & & & x_c \in \{0,1\} & \forall c \end{array}$$

Unless otherwise specified, vertices v range over the set V, while cycles and chains c range over the set of all legal cycles and chains $\mathcal{C}(L,K)$ (for caps L and K, which may grow with $|V|$). Here, x_c is a binary variable that is set to 1 if cycle c is included in the final matching; otherwise, it is 0. The constant $h(c)$ is the number of highly-sensitized patients in a cycle or chain c. Formally,

$$h(c) = |\{v \mid v \in c \wedge v \in V_H\}|$$

Note that $h(c) = 0$ if a cycle or chain does not contain any highly-sensitized patients, so only those binary variables x_c corresponding to cycles or chains with at least one highly-sensitized patient need be included in the objective. However, all *vertices* belonging to at least one cycle or chain that contains at least one highly-sensitized vertex must be included in the vertex-disjointness constraints (to maintain the feasibility of the final disjoint cycle cover).

Given this setup, a match could be considered equitable if it satisfies some nonnegative, user-defined parameter $\alpha \leq \alpha^*$, such that the matching algorithm includes $\alpha \cdot |V_H|$ highly-sensitized patients in the "optimal" match. E.g., if $\alpha = 0.5$, any returned match would include at least 50% of the number of highly-sensitized patients available. Formally, define the *deterministic lexicographic* fairness rule $u^{\alpha}_{H \succ L}$ over any $M \in \mathcal{M}$ as follows:

$$u^{\alpha}_{H \succ L}(M) = \begin{cases} u(M) & \text{if } |M_H| \geq \alpha \cdot |V_H| \\ 0 & \text{otherwise} \end{cases}$$

We implement this utility function by adding a single constraint to the standard integer program, yielding:

$$\begin{array}{lll} \max & \sum_c |c|\, x_c & \\ \text{s.t.} & \sum_{c|v\in c} x_c \leq 1 & \forall v \\ & \sum_c h(c) x_c \geq \alpha \cdot |V_H| & \\ & x_c \in \{0,1\} & \forall c \end{array}$$

In the next section, we generalize this rule for probabilistic kidney exchange.

Probabilistic (failure-aware) model

As motivated in Chapter 5, most algorithmic matches in fielded kidney exchanges do not result in an actual transplant, even if a patient and donor are seen as ABO- and tissue-type compatible by the optimization software. A variety of issues arise, including sudden illness or death, improper medical testing, a patient finding a deceased or alternate living donor outside of the kidney exchange, or match failure due to more intensive (and expensive) post-match medical testing.

Recall that the failure-aware model maximize the expected number of transplants subject to each edge e in the graph having some probability of success q_e. In the lexicographic fairness model we consider in this chapter, this is further constrained by ensuring some fraction of these expected transplants are to highly-sensitized patients. Toward this end, define $v^h(c)$, the expected number of transplants to highly-sensitized patients of a cycle or chain c, as follows:

Cycles. For cycles, the discounted utility is the same as in Chapter 5, but counting only those vertices in the cycle that are highly-sensitized. Formally,

$$v^h(c) = h(c) \prod_{e\in c} q_e$$

Chains. While cycles necessarily execute atomically, chains can execute partially and then fail—thus making the calculation of v^h a bit trickier. For a chain of length k, let c^i represent the initial i vertices in the chain (including the altruistic donor). Formally,

$$v^h(c) = \left[\sum_{i=1}^{k-1}(1-q_i)h(c^{i-1})\prod_{j=0}^{i-1} q_j\right] + \left[h(c)\prod_{i=0}^{k-1} q_i\right]$$

We now let N^* be the maximum *expected* number of vertices in V_H that could receive transplants over all possible matchings $\mathcal{M}$ on G. That is,

$$N^* = \max_{M\in\mathcal{M}} v^h(M)$$

where for any matching M, $v^h(M)$ is defined as $\sum_{c\in M} v^h(c)$.

Then for a given graph G, this failure-aware N^* can be computed using a similar modification of the failure-aware kidney exchange integer program as follows:

$$\begin{array}{rll} N^* = & \max & \sum_c v^h(c) x_c \\ & \text{s.t.} & \sum_{c|v \in c} x_c \leq 1 \quad \forall v \\ & & x_c \in \{0,1\} \quad \forall c \end{array}$$

Now, as above, the user can supply some parameter $\alpha \in [0,1]$ that guarantees α fraction of the maximum possible expected number of transplants for highly sensitized-patients. This preference is enforced by adding a single constraint to the standard failure-aware kidney exchange IP as follows:

$$\begin{array}{lll} \max & \sum_c v(c) x_c & \\ \text{s.t.} & \sum_{c|v \in c} x_c \leq 1 & \forall v \\ & \sum_c v^h(c) x_c \geq \alpha \cdot N^* & \\ & x_c \in \{0,1\} & \forall c \end{array}$$

Here, $v(c)$ is defined similarly to $v^h(c)$, only including lowly-sensitized transplants in the expectation calculation as well.

A note on implementing this rule

Unfortunately, adding lexicographic preferences into the optimization model breaks the specific branch-and-price structure on which the basic cycle formulation solvers rely (both in the deterministic model [6] and in the probabilistic model of Chapter 5), as well as the branch-and-price version of PICEF presented in Chapter 3. While a solver could still use column generation or general branch-and-price to solve this new problem, the addition of a matching-wide constraint—that a matching must contain cycles containing some fraction of a set of marked vertices—makes solving the pricing problem (see [30] for details) much more difficult than in the utilitarian formulation, where determining the price of a cycle not included in the current subproblem is relatively simple. Indeed, with such an allocation-wide constraint, finding a positive price cycle at a node in the search tree requires solving an integer program, whereas current solvers can use a depth-first search—or, in the case of PICEF, a polynomial-time algorithm—to find a positive price cycle in the standard kidney exchange model.

With these computational constraints in mind, in Section 7.3.2 we define a different fair utility function that respects the constraints of current solvers. This utility function circumvents a matching-wide fairness constraint. We will then compare both fair utility functions against the utilitarian one on real and simulated data in Section 7.4.

7.3.2 Weighted fairness

We now present a different formalization of fairness that relaxes the strict lexicographic preferences from the previous section. This definition generalizes the policy UNOS currently applies to highly-sensitized patients in their fielded kidney exchange, is what was used in the preliminary experiments of Section 5.9, and is what will be used in our most realistic experiments in Chapter 8.

Building on the standard (deterministic or probabilistic) kidney exchange integer programming formulation, we rewrite the objective as follows:

$$\max \quad \sum_c v_\Delta(c) x_c$$

Here, $v_\Delta(c)$ is the value of a cycle or chain c (either in the deterministic or probabilistic model) such that the weight of each edge $e \in c$ is adjusted by some re-weighting function $\Delta : E \to \mathbb{R}$.

A simple example re-weighting function is multiplicative:

$$\Delta^\beta(e) = \begin{cases} (1+\beta)w_e & \text{if } e \text{ ends in } V_H \\ w_e & \text{otherwise} \end{cases}$$

Intuitively, for some $\beta > 0$, this function scales the weight of edges ending in highly-sensitized vertices by $(1+\beta)$. For example, if $\beta = 0.5$, then the optimization algorithm will value edges that result in a highly-sensitized patient receiving a transplant at 50% above their initial weight (possibly scaled by other factors like failure probability and chain position, as in the probabilistic model). We will use this multiplicative re-weighting in our experiments in Section 7.4 and, later, in Chapter 8 with the FUTUREMATCH framework.

For any $M \in \mathcal{M}$, let M' be the matching such that every cycle $c \in M$ has augmented weight $v_\Delta(c)$. Then define the *weighted* fairness rule u_Δ in terms of the utilitarian rule u applied to the augmented matching M', such that $u_\Delta(M) = u(M')$.

A note on implementing this rule

Note that, unlike implementing the lexicographic fairness rule $u^\alpha_{H \succ L}$, this definition of fairness does not break the branch-and-price structure on which current scalable kidney exchange solvers rely. Indeed, the u_Δ rule, for simple re-weighting functions like the multiplicative example above, can be implemented by first preprocessing a compatibility graph using Δ to determine edge weights, and then solving the maximization problem using any standard kidney exchange solver (branch-and-price-based or otherwise).

7.4 Experimental validation

In this section, we compare the behavior of both the lexicographic fair rule $u^\alpha_{H \succ L}$ and weighted fair rule u_Δ (relative to the utilitarian rule u) on (a) real data from the first 73 UNOS kidney exchange match runs, from October 2010 to August 2013, and (b) simulated data from the standard kidney exchange compatibility graph models.

7.4.1 Results from the fielded UNOS exchange

We present fairness experiments on the first 73 match runs (through August of 2013) in this section. The UNOS algorithm currently performs a utilitarian maximum weighted matching, where

edge weights are set through a point-based system determined by a committee of doctors and policymakers. Highly-sensitized patients in the UNOS exchange (at the time these experiments were run) are those with CPRA of at least 80.

Results are presented for both the deterministic (denoted NO-FAIL) and probabilistic versions of the kidney exchange clearing problem. Motivated by Dickerson, Procaccia, and Sandholm [75], we test the probabilistic model on two different edge failure probability distributions: one where each edge has a constant 70% rate of failure (denoted CONSTANT), and the other with a bimodal failure rate such that 25% of edge failures are drawn from $U[0.0, 0.2]$ and the rest from $U[0.8, 1.0]$ (denoted BIMODAL). These distributions are derived from real data (see Section 5.4.1 for details). We use a cycle cap of $L = 3$, as does UNOS, and include chains.

Lexicographic rule. We begin with the strictest version of the lexicographic fair rule: $u^{1.0}_{H \succ L}$. With $\alpha = 1.0$, this rule maximizes the number of highly-sensitized pairs in a match (which aligns with the theory of Section 7.2).

Metric	Minimum	Average	Maximum	St. Dev.
Loss (Objective) %	0.00%	2.76%	19.04%	4.84%
Loss (Cardinality) %	0.00%	4.09%	33.33%	8.18%
Loss (Cardinality)	0	0.55	4	1.10

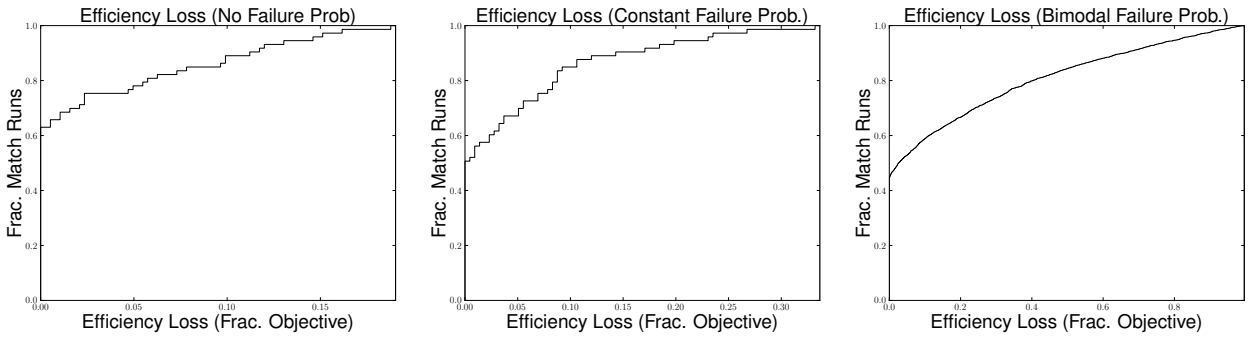
Table 7.1: Minimum, average, and maximum loss in objective value and match size due to $u^{1.0}_{H \succ L}$, across the first 73 UNOS match runs, in the deterministic model.

Table 7.1 presents results in the deterministic model. Under $u^{1.0}_{H \succ L}$, the price of fairness in this deterministic model is, on average, quite small; however, there are outlier cases in which large relative losses in the objective (of 19%) and overall match size (of 33%) are observed. We now explore this phenomenon in depth, in both the deterministic and probabilistic model.

Figure 7.3 presents cumulative distribution functions in efficiency loss with respective to the UNOS-weighted objective value, for the NO-FAIL, CONSTANT, and BIMODAL models. In each of the models, roughly half of all UNOS match runs see no efficiency loss when prioritizing highly-sensitized candidates. However, (a) in each model, there exist a nontrivial number of matches with a nontrivial loss in efficiency, and (b) increasing the variability in failure rates increases the price of fairness. Indeed, in the BIMODAL model, some runs have a nearly 100% loss in efficiency! Intuitively, this is due to the optimizer being "forced" into including edges—possibly with a very low chance of successful execution—that result in a potential highly-sensitized transplant.

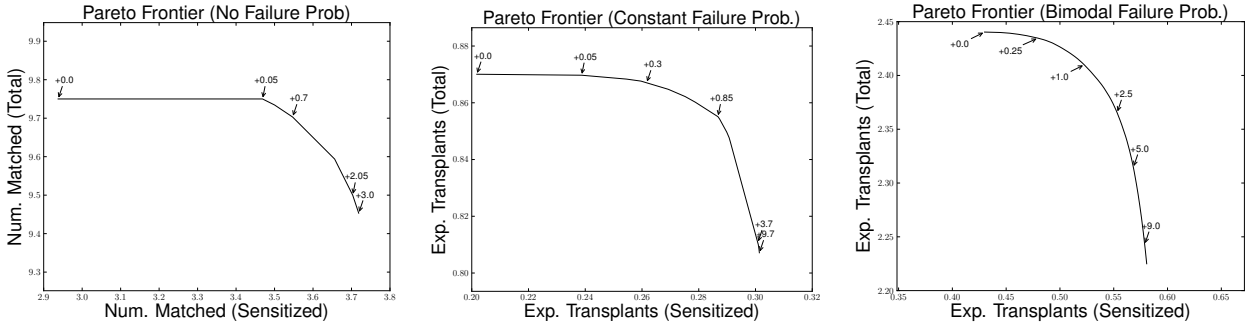
Weighted rule. We now give results for the weighted fair rule u_Δ, where edges are re-weighted under the multiplicative rule Δ^β defined in Section 7.3.2. Recall that $\Delta^{0.0}$ values highly-sensitized transplants at the same rate as lowly-sensitized ones, $\Delta^{0.5}$ values them at 50% over their base value, $\Delta^{1.0}$ at 100% over their base value, etc.

The Pareto frontiers shown in Figure 7.4 represent the set of Pareto efficient matchings constrained by u_Δ as β increases from 0, for each of the NO-FAIL, CONSTANT, and BIMODAL



(a) NO-FAIL **(b)** CONSTANT **(c)** BIMODAL

Figure 7.3: Cumulative distribution functions of the price of fairness under the lexicographic fairness rule $u^{1.0}_{H \succ L}$ according to UNOS' weighting policy, on 73 UNOS match runs since the inception of the exchange.



(a) NO-FAIL **(b)** CONSTANT **(c)** BIMODAL

Figure 7.4: Pareto frontiers for u_Δ under different failure probability distributions, for $\beta \in \{0.0, 0.05, \ldots, 10.0\}$.

models. Intuitively, these plots visualize the relationship between favoring a subset of vertices versus the overall match efficiency. As with the lexicographic rule, higher diversity in the underlying failure probability distribution begets a greater price of fairness.

7.4.2 Simulated results from random graphs

Fielded kidney exchanges are still young and have relatively small pools, containing at most a couple of hundred pairs at a time. To explore fairness-aware matching behavior in larger pool sizes, and to validate the theory developed in Section 7.2, we now turn to generated data. We look at two models:

1. The most well-known dense model of kidney exchange is due to Saidman et al. [191]. It gen-

eralizes the theoretical model in Section 7.2, and includes blood types, three tiers of CPRA (low, medium, and high), and various other medical aspects that affect pool composition (see Chapter 2 for details). We refer to the base Saidman model, which is parameterized with US national data, as SAIDMAN (US). We also parameterize the model using blood type and sensitization statistics from the 73 UNOS match runs (aggregate data is available in [132]), and denote this SAIDMAN (UNOS).

2. Some recent theoretical work looks at kidney exchange graphs without blood types but with increasing sparsity (in the size of the pool) for highly-sensitized candidates [23, 24]. For these experiments, the probability of an incoming edge to a highly-sensitized pair is $O(1/n)$. (See Section 5.3.1 for details.) The probability of an incoming edge to lowly-sensitized pairs is held constant, as before. We denote by HETEROGENEOUS this family of random graphs.

Size	SAIDMAN (US)	SAIDMAN (UNOS)	HETEROGENEOUS
10	0.24% (1.98%)	0.00% (0.00%)	0.98% (5.27%)
25	0.58% (1.90%)	0.19% (1.75%)	0.00% (0.00%)
50	1.18% (2.34%)	1.96% (6.69%)	0.00% (0.00%)
100	1.46% (1.80%)	1.66% (3.64%)	0.00% (0.00%)
150	1.20% (1.86%)	2.04% (2.51%)	0.00% (0.00%)
200	1.43% (2.08%)	1.55% (1.79%)	0.00% (0.00%)
250	0.80% (1.24%)	1.86% (1.63%)	0.00% (0.00%)
500	0.72% (0.74%)	1.67% (0.82%)	0.00% (0.00%)

Table 7.2: Average (St. Dev.) percentage loss in efficiency for three families of random graphs, under the strict $u^{1,0}_{H \succ L}$ rule.

Table 7.2 gives the average loss in efficiency for each of these models over multiple generated pool sizes, with 40 runs per pool size per model, under the strict lexicographic rule $u^{1,0}_{H \succ L}$. For all but the smallest pools, HETEROGENEOUS graphs see no loss in efficiency at all. Efficiency loss in the SAIDMAN (US) and SAIDMAN (UNOS) families of graphs are low (and statistically indistinguishable), aligning with our earlier theoretical results (applied to the distribution of blood types in the US).

7.5 Conclusions & future research

Fielded kidney exchanges use utilitarian or near-utilitarian matching rules, at the cost of marginalizing certain classes of patient-donor pairs. We addressed the utilitarian setting in depth in Parts I and II. In this chapter, we focused on balancing overall exchange efficiency while improving access to kidneys for highly-sensitized patients. We defined the price of fairness in the standard kidney exchange model, and provided theoretical bounds in the major kidney exchange model. We introduced two natural definitions of fairness—lexicographic and weighted—and empirically explored the tradeoff between prioritizing hard-to-match patients and the overall efficiency of a utilitarian

system, on real data from the UNOS nationwide kidney exchange and on simulated data from each of the standard kidney exchange distributions.

How to prioritize highly-sensitized patients—if they should be prioritized at all—is currently the most contentious issue regarding fairness in kidney exchange, which motivates this chapter (and, indeed, Part III of this thesis in general). We note that the price of fairness concept applies to valuing *any* subset of vertices—not just highly-sensitized ones—in the compatibility graph, possibly under different prioritization rules. A clear next step would be developing analogous theoretical results and empirical techniques applicable to fielded kidney exchange that generalize the equity versus efficiency tradeoff presented here to other notions of fairness (while mimicking present-day parameters of the compatibility pool, legal climate, and limits of medical knowledge). Recent work in finding Lorenz-dominant matchings is promising [146, 186], but not yet applicable to fielded kidney exchange due to its simple theoretical model (e.g., only 2-cycles, no chains). Hooker and Williams [110] give a mathematical-programming-based approach to balancing equity—in the Rawlsian maximin sense—and efficiency in a general model; similar results with different definitions of equity would be of theoretical and practical interest.

Finally, as discussed in Chapter 6, kidney exchange is naturally dynamic, where patients and donors arrive to and depart from the pool over time. Developing accurate models and scalable algorithms that consider the price of fairness in the dynamic setting will be of increasing importance as fielded kidney exchanges move from static to dynamic matching. We present one such approach in the succeeding chapter.

When someone speaks of a good strategy or a bad plan, they are making a prudential judgment about the efficacy of the plan or strategy in question, that is, whether it will achieve certain ends.

– Caroline Whitbeck

A lot of times, people don't know what they want until you show it to them.

– Steve Jobs

8

FUTUREMATCH: Learning to match in dynamic environments

In this chapter, we combine a variety of dimensions discussed in previous chapters—scalable clearing of deterministic and probabilistic exchanges (Chapters 3, 4, and 5), considering short-term uncertainty explicitly in the optimization problem (Chapter 5), taking distributional information about the future of the exchange when matching in the now (Chapter 6), and considerations regarding fairness (Chapter 7)—into a single, unified framework for learning to matching "well" under a general objective function.

As a proof of concept for this thesis, we learn a novel transplant quality predictor from a dataset consisting of all living-donor kidney transplant events in the US since Oct 1, 1987, and discover that some present-day features of living-donor transplants do not align with older results from deceased donation in the medical literature [168]. This is noteworthy since today's exchange priority policies have largely been inherited from the United Network for Organ Sharing (UNOS) deceased-donor waiting list policies.

Motivated by our experience running the computational side of the UNOS nationwide kidney exchange, we present FUTUREMATCH, a general framework for learning how to match in dynamic environments. FUTUREMATCH separates the "means" from the "ends" of kidney exchange; it takes as input from human experts an overarching objective, and automatically learns a matching strategy to achieve this goal. We validate the framework on three example objective functions on real data drawn from the UNOS exchange. We find that using FUTUREMATCH *even with economically inefficient objectives*—like maximizing the match size subject to equity constraints, as in Chapter 7—results in significantly higher efficiency than myopic matching with the explicit objective of efficiency.

Related Publications

Some of the work in this chapter appeared at AAAI-15; it was performed as a collaboration between Dickerson and Sandholm [70].

8.1 Proposed method

We now present the FUTUREMATCH framework for learning to match in dynamic environments. We begin by motivating and describing the framework at a high level in the first subsection. In the following subsections we discuss the different parts of the framework in detail and how we instantiated them for kidney exchange.

8.1.1 The FUTUREMATCH framework

We are interested in learning from demographically accurate data how to match *in the present* such that some overarching objective function is maximized over time. Scalability is important: heavy offline statistics can be computed and periodically updated, but the fielded clearing algorithm must run quickly (within minutes or at most hours).

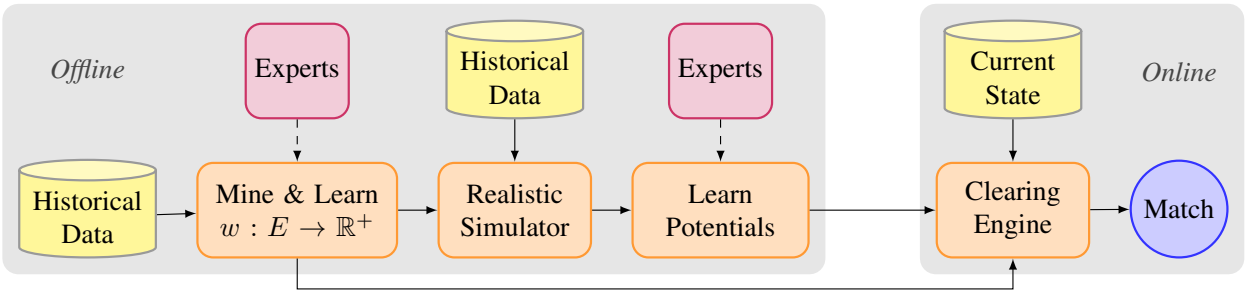


Figure 8.1: The FUTUREMATCH framework.

Figure 8.1 graphically depicts the FUTUREMATCH framework. A domain expert (e.g., a committee of medical and legal professionals) begins by describing an overall objective function for the exchange. Even *measuring* this objective can be difficult: for example, if the goal is to maximize the number of days added to patients' lives via kidney transplantation, then calculating the relative quality of a proposed match requires knowing some notion of utility for each edge—representing a potential transplant—in the compatibility graph. We propose to learn this edge weight function $w : E \to \mathbb{R}^+$ from data, and give examples for a variety of objective functions later.

The learned weight function w is then fed into a parameterized (kidney exchange) simulator, calibrated by real data so that it mimics the underlying distribution. This generator in turn feeds training and test sets into a system for learning the potentials of element classes in the compatibility graph, by way of the method introduced in Chapter 6. Intuitively, given an element θ (e.g., vertex, edge, cycle, or chain type), a potential $P_\theta \in \mathbb{R}$ quantifies the expected utility to the exchange of

that element in the future. Potentials are combined with w to quantify an edge-specific quality rating. Here, as in Chapter 6, we learn potentials for the combinations of different blood types for pairs under each of the weight functions we define—albeit with a different learning method than in that previous chapter.

Finally, the fielded clearing algorithm incorporates the combined weight function w and set of potentials $\mathcal{P}_\Theta$ into its myopic weighted matching algorithm. This incorporation comes at very low or no cost to the runtime of the clearing algorithm; indeed, the final "potential-aware" input graph is simply a re-weighted version of the original compatibility graph, using the weights that encode the future.

In the rest of the section, we describe an in-depth implementation of FUTUREMATCH. Our goals are twofold: first, to show the general applicability and tractability of the framework, and second, to mimic a large fielded kidney exchange. Accomplishing this second goal, and leveraging our involvement with fielded kidney exchanges, sets the stage for adoption of sustainability-motivated technology that solves a problem clearly too difficult for humans—a success story for computational sustainability practitioners.

8.1.2 Encoding an objective function

We now discuss in depth the process of defining an objective for FUTUREMATCH. We do this in the context of kidney exchange, but note that the process is general.

The medical and legal communities in kidney exchange are concerned about a wide variety of match characteristics. In our experience, the most frequently discussed include the number of overall matches, the number of overall transplants, the quality of transplants, and whether or not to prefer specific subgroups in the exchange (children, sensitized patients, underrepresented ethnicities) and by how much. Other concerns could include notions of fair treatment among participating centers and minimizing legal exposure.

In this chapter, we consider two different kidney exchange models—deterministic, where post-algorithmic match failures are not quantified in the optimization problem and failure-aware, where they are—and three matching objectives in each of the two models:

1. MAXCARD: Maximize the total number (i.e., cardinality) of patients who are algorithmically matched (in the deterministic model of Part I) or receive transplants in expectation (in the failure-aware model of Chapter 5);
2. MAXCARD-FAIR: Maximize the total number of patients who are algorithmically matched (in the deterministic model) or receive transplants in expectation (in the failure-aware model), where "marginalized" patients are weighted in the objective by some constant factor β more than others, as in Section 7.3.2; and
3. MAXLIFE: Maximize the total time algorithmically-matched (deterministic) or transplanted (failure-aware) donor organs will last in patients.

Each of these objectives amounts to setting weights on edges in the input graph. Next, we detail edge weighting algorithms for these example objectives.

In our experience, when committees debate priority points for today's exchanges, the discussions confound the goal and the means. For example, a goal could be to maximize matches and

a means could be to prioritize sensitized patients because they are harder to match in the future. On the other hand, many argue that sensitized patients should be inherently preferred, and it seems that most do not make a clear distinction between means and ends. In contrast, FUTUREMATCH clearly *separates ends and means*. Our objective (i.e., the "end") lives in the space of weights on edges, which the committee can clearly debate. On the other hand, our framework automatically optimizes, via learning, the potentials (the "means") that are used as the means for enabling the algorithm to make good future-aware failure-aware decisions. The committee does not need to debate these potentials, whose quantitative impact on performance is hard for a human to predict or even understand.

Defining MAXCARD and MAXCARD-FAIR. The MAXCARD-FAIR objective can be viewed as a generalized form of MAXCARD (that is, MAXCARD is just MAXCARD-FAIR with an empty set of vertices who are preferred by the objective). A natural weighted fairness rule, adapted from Chapter 7, adjusts edge weights by some re-weighting function $\Delta : E \to \mathbb{R}^+$. A simple example re-weighting function is multiplicative:

$$\Delta^{\beta}(e) = \begin{cases} (1+\beta)w_e & \text{if } e \text{ ends in } V_P \\ w_e & \text{otherwise} \end{cases}$$

Here, $V_P \subseteq V$ is the set of preferred vertices (in Chapter 7, these were vertices with highly-sensitized patients; we will define a different subset in the experiments). Intuitively, for some $\beta > 0$, this function scales the weight of edges ending in marginalized vertices by $(1+\beta)$. For example, if $\beta = 1.5$, then the optimization algorithm will value edges that result in a marginalized patient receiving a transplant at 250% of their initial weight (possibly scaled by factors such as edge failure probability or chain position, as we discuss later).

For any $M \in \mathcal{M}$, let M' be the matching such that every edge $e \in E$ has augmented weight $\Delta^{\beta}(e)$. Then the MAXCARD-FAIR utility function u_{Δ} is defined in terms of the utilitarian MAXCARD utility function u applied to the augmented matching M', such that $u_{\Delta}(M) = u(M')$. In the experiments, we vary the parameter β to empirically quantify its effects on each of the three objective functions.

Optimizing for MAXLIFE via learning to predict graft survival from data. With the MAXLIFE objective we are interested in maximizing how long the transplanted kidneys, in aggregate, survive in the patients.[1] To do so, we must first determine an empirically sound estimate of the lifespan of a transplant as a function of donor and recipient attributes.

Delen et al. [69] compare a variety of techniques for predicting breast cancer survivability; unlike their study, we are interested in predicting the survival *length* of a kidney graft, as opposed to whether or not a patient survives treatment at all. Data mining models are also actively being developed to predict the risk of readmission for congestive heart failure patients [157]. Most related to our work is the Kidney Donor Profile Index (KDPI), which is currently under development by

[1] Another objective would be to maximize aggregate increase in life duration. This would involve subtracting out the expected life duration without a transplant from the expected life duration with the transplant, and could incorporate the possibility of additional transplants after graft failure.

UNOS for use in the deceased donor allocation process [133]. The KDPI score of a deceased donor kidney measures the estimated quality of the donor organ being allocated to the *average* recipient. In contrast, our predictor, which we will describe next, provides a unique quality score not just based on donor attributes but also based on attributes of the specific potential recipient.

We look at all $75{,}264$ *living donor* transplant events in the US between October 1, 1987 and June 30, 2013. This data includes medical characteristics of the recipient and donor at the time of transplantation, as well as follow-up data regarding the health of the recipient and the recipient's new kidney; this follow-up data is updated at least annually.

Conditioned on a kidney graft being marked as failed in our dataset, the average graft lifetime is about 1912.7 days, or slightly over 5 years. However, due in large part to the marked increase in kidney failure since the late 1980s, nearly 75% of grafts in the dataset are not marked as failed. This occurs because either (i) the recipient is still alive with a functioning donated kidney or (ii) the recipient has died, but for a non-kidney-failure-related reason. Thus, we use *survival analysis* to estimate the lasting power of a graft.

Features of both the recipient and donor have a large effect on graft survival. For example, tissue type (HLA) testing measures the closeness of match between antigens in the cells of a donor and patient. Figure 8.2 gives a Kaplan-Meier estimator of the survival functions of (i) kidney transplants resulting from a donor and recipient being a perfect HLA match and (ii) those resulting from imperfect HLA matchings. Clearly, a kidney that is a perfect tissue type match is more desirable than an imperfectly matched one; indeed, the model estimates a median survival time of 5808 days for a perfect match compared to 4300 for an imperfect match. A log-rank test revealed that the difference between the two distributions was significant ($p \ll 0.0001$).

In our experiments, we use a Cox proportional hazards regression analysis to explore the effect of multiple features on survivability. At a high level, this method regresses the survival time of the graft against explanatory features of the donor and recipient. More specifically, define the *hazard* H at time t days after a transplant as follows:

$$H(t) = H_0(t) \times \exp(b_1 X_1 + b_2 X_2 + \ldots + b_k X_k) \tag{8.1}$$

Here, each X_i is a predictor variable corresponding to a single feature of the donor or recipient, and $H_0(t)$ is a baseline hazard rate at time t for a recipient with $X_i = 0$ for $i \in \{1, \ldots, k\}$. Then $H(t)$ represents the instantaneous risk of graft failure at time t. We want to learn this function.

To begin, we include the following features: recipient age, difference in donor and recipient's age, donor HLA profile, recipient HLA profile, donor and recipient blood type compatibility. The HLA profile of a donor or recipient is separated into three integral features—HLA-A, HLA-B, and HLA-DR—that can take values in $\{0, 1, 2\}$, representing 0, 1, or 2 mismatches. By separating the general HLA mismatch feature into three separate mismatch features, we complicate (but increase the power of) the model [168]; this separation is motivated by evidence that mismatches at the HLA-A, -B, and -DR level have varyingly negative impact on survival.

We ran a Cox proportional hazards regression on this unpruned feature space. This used $74{,}244$ live donor transplantations during which there were $18{,}714$ graft failures (920 live donation events were dropped due to one or more missing features). Our initial regression showed that increases in the HLA-B mismatch feature did not have a significant effect on the dependent variable ($p = 0.22$).

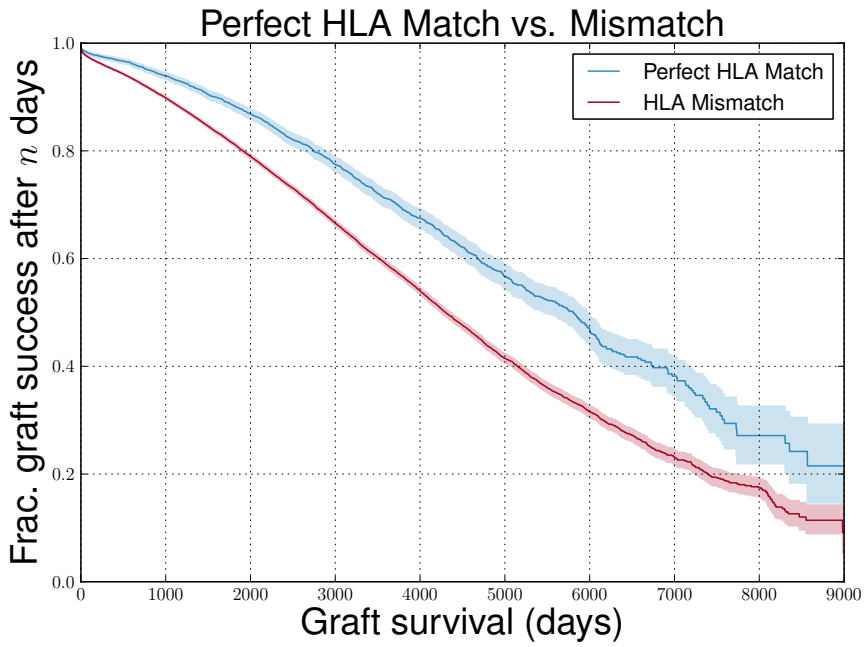


Figure 8.2: Kaplan-Meier estimator of the survival function for kidney transplants whose donors have zero HLA mismatches versus those with at least one HLA mismatch, with 95% confidence intervals.

Prior research from the mid-1980s on *cadaveric* donation found a significant relationship between the combined feature of HLA-B and HLA-DR mismatches on graft outcome [168]; we find that this does not hold on living donor data in the present. After selecting significant variables in this initial run—that is, all of the attributes previously discussed *except* HLA-B mismatch—we re-ran a Cox proportional hazard regression. Results are reported in Table 8.1.

feature	$\exp(b_i)$	$\text{SE}(b_i)$	z	p
recipient age	1.00753	0.0008	9.715	$< 2 \times 10^{-16}$
age diff.	1.00525	0.0007	7.766	8.10×10^{-15}
HLA-A	1.05273	0.0120	4.297	1.73×10^{-5}
HLA-DR	1.08680	0.0119	6.984	2.86×10^{-12}
ABO incomp.	1.37871	0.0748	4.295	1.74×10^{-5}

Table 8.1: Learned weights via Cox regression after feature pruning for statistical significance.

Table 8.1 gives the standard error, z-score, and corresponding p-value for each of our pruned features; each clearly has a statistically significant effect on graft survival. To interpret the results, as an example first consider the HLA-DR feature. We see that $\exp(b_{\text{HLA-DR}}) \approx 1.087$; recalling Equation 8.1, a unit increase in the HLA-DR mismatch feature will result in a factor of 1.087 increase over the baseline hazard rate. Varying either recipient age or the difference in donor and recipient age was also statistically significant, with a unit increase in recipient age having a larger

effect on the hazard rate. As might be expected, blood type compatibility plays the largest role in hazard rate, where an incompatible (and thus heavily immunosuppressed) transplant has a factor 1.379 increase over the base hazard rate.

Using this data, we can estimate $S_e(t)$, the survival probability at time t for a potential transplant $e \in E$ between a recipient and donor with features x_i^e, as follows:

$$S_e(t) = \exp\left(-H_0(t) \times \sum_i x_i^e b_i\right). \tag{8.2}$$

Building on Equation 8.2, we define a weight function $w : E \to \mathbb{R}^+$ as

$$w(e) \propto \exp\left(-\sum_i x_i^e b_i\right). \tag{8.3}$$

Intuitively, the weight function w assigns higher relative weight to edges with lower risk, in turn biasing the optimizer toward transplants with longer expected graft survival. In our experiments, we explicitly instantiate $w(e) = 100 \times \exp\left(-\sum_i x_i^e b_i\right)$ to avoid numerical instability in our linear program subsolvers as a byproduct of small floating point coefficients in the objective.

8.1.3 Learning the potentials

We learn potentials on the blood types of patients and donors, as in Chapter 6. There are $4 \times 4 = 16$ combinations of patient and donor blood types, and 4 possible blood types of altruists. So, we have 20 different kinds of vertices. We want to learn a potential for each of those 20 vertex types. Formally, the types of vertex are $\Theta_{\text{ABO}} = \{\text{O-O}, \text{O-A}, \ldots, \text{AB-B}, \text{AB-AB}\} \cup \{\text{O}, \ldots, \text{AB}\}$, and we want to learn values P_θ for $\theta \in \Theta_{\text{ABO}}$.

We combine the learned potentials P_θ with the weight function w learned earlier using a function $f_w : E \to \mathbb{R}$. It balances the *myopic* value of an edge encoded by w with the *future value* of an edge encoded by potentials. The idea is that the revised weight, f_w, of an edge is its immediate value if matched minus the potentials of its vertices because if those vertices are matched now, they cannot contribute to future matches. Specifically, $f_w(e) = w(e) \cdot (1 - P_{\theta_d} - P_{\theta_p})$, where d and p are the vertices adjacent to edge e.

We use SMAC [114], a state-of-the-art model-based algorithm configuration tool that searches through a parameter space to optimize a given objective. SMAC guides its navigation in the space of parameter vectors by constructing a model that predicts algorithm performance as a function of the parameter vector (e.g., $g : \Theta_{\text{ABO}} \to \mathbb{R}$, where Θ_{ABO} is our space of potentials and $\mathbb{R}$ is the aggregate evaluation of whatever objective is being used). We found the SMAC-based approach to work substantially better than the ParamILS tuner used in Chapter 6.

In our setting, the parameter vector is the vector of potentials. At each parameter vector that SMAC navigates to, we run a large number of trials of our simulator of a kidney exchange to see how the batch matching algorithm would perform in a dynamic setting using that parameter vector. That performance number is then fed back to SMAC, and SMAC navigates to the next parameter vector to continue the search. We learned potentials in two models—deterministic, as

in Part I, where post-algorithmic match failures are not quantified in the optimization problem and failure-aware, as in Chapter 5, where they are—using a realistic dynamic simulator we built based on historical data from the UNOS kidney exchange [132].

Figure 5.7 gives a graphical overview of the dynamic model we used. A matching is determined at each time period based on either a deterministic or failure-aware clearing algorithm. Recall that both models compute an optimal matching $M^* = \arg\max_{M \in \mathcal{M}} u(M)$, where $u(M) = \sum_{c \in M} u(c)$. Given the function f_w that combines our learned potentials P_θ with the myopic edge weighting function w, in the deterministic model, $u(c) = \sum_{e \in c} f_w(e)$: that is, the sum of the weights of the constituent edges in a cycle or chain subject to the weight function w and potentials learned earlier.

In the failure-aware model, given edge success probabilities $q_e \in [0, 1]$ for each edge e, the potential-aware discounted utility for a cycle c is $u(c) = \left[\sum_{e \in c} f_w(e)\right] \cdot \left[\prod_{e \in c} q_e\right]$, and the discounted utility for a chain $c = \langle e_0, e_1, \ldots, e_{k-1} \rangle$ is

$$u(c) = \left[\sum_{i=1}^{k-1} (1 - q_i) \sum_{j=0}^{i-1} f_w(e_j) \prod_{j=0}^{i-1} q_j\right] + \left[\sum_{i=0}^{k-1} f_w(e_i) \prod_{i=0}^{k-1} q_i\right].$$

8.2 Experiments

We validated FUTUREMATCH experimentally on data from the UNOS nationwide kidney exchange. We explore the effect each of the three objectives—MAXCARD, MAXCARD-FAIR, and MAXLIFE—has on a variety of metrics under FUTUREMATCH and under myopic deterministic matching, which is the fielded state of the art. The latter does not take edge failure or learned potentials into account during optimization; as described earlier, it finds a maximum weight matching (i.e., for each chain or cycle c, $u(c) = \sum_{e \in c} w_e$) during each period separately.

In the fairness-weighted expariments, we adapt our matching algorithm using the re-weighting function Δ^β described earlier. The preferred set of vertices V_P includes those with a pediatric or highly-sensitized patient. These preferences are commonly used in kidney exchanges, albeit not in sophisticated, quantitative ways. For kidney exchange it has explicitly been articulated that pediatric patients should be preferred not only because they have a lot of life left (barring their kidney disease) but also because having poor kidney function stunts growth. Similarly, some patients are *highly sensitized*, which means they are extremely unlikely to be medically compatible with a random organ. For these patients, finding a kidney is difficult [212]. In fielded exchanges, both of these "marginalized" patient types are prioritized. We quantitatively explore how this should be done and what the impact is.

8.2.1 Results

We compare FUTUREMATCH against a baseline of myopic deterministic matching under each of the objectives. Conservatively, statistical significance was determined using the Wilcoxon signed-rank test, which is a nonparametric alternative to the paired t-test. Table 8.2 shows the median expected gain in the overall number of transplants from using FUTUREMATCH under each of the

objectives. Each column labeled $|V| = k$ corresponds to a simulation over k patient-donor pairs and altruists; we test over increasing values of k because kidney exchanges (both in the US and worldwide) are still expanding toward their steady-state sizes.

Total	$\lvert V\rvert = 300$		$\lvert V\rvert = 400$		$\lvert V\rvert = 500$		$\lvert V\rvert = 600$		$\lvert V\rvert = 700$		$\lvert V\rvert = 800$		$\lvert V\rvert = 900$	
	Gain	p	*Gain*	p	*Gain*	p	*Gain*	p	*Gain*	p	*Gain*	p	*Gain*	p
MAXCARD	+2	✓	+4	✓	+5	✓	+6	✓	+10	✓	+11	✓	+13	✓
MAXCARD-FAIR, $\beta = 1$	+1	✓	+4	✓	+6	✓	+8	✓	+9	✓	+11	✓	+12	✓
MAXCARD-FAIR, $\beta = 2$	+1		+2	✓	+3	✓	+3	✓	+5	✓	+6	✓	+10	✓
MAXCARD-FAIR, $\beta = 3$	+1		+0		+3	✓	+1		+1	✓	+3	✓	+2	
MAXCARD-FAIR, $\beta = 4$	-1		+1		+1		+1		+3	✓	+3		+2	
MAXCARD-FAIR, $\beta = 5$	+0		+0		+1		+1		+1		+2		+3	
MAXLIFE	+2	✓	+3	✓	+6	✓	+8	✓	+7	✓	+11	✓	+9	✓

Table 8.2: Median gains in expected total number of transplants under FUTUREMATCH. A ✓ or ✗ represents statistical significance (Wilcoxon signed-rank test, $p \ll 0.01$).

Marginalized	$\lvert V\rvert = 300$		$\lvert V\rvert = 400$		$\lvert V\rvert = 500$		$\lvert V\rvert = 600$		$\lvert V\rvert = 700$		$\lvert V\rvert = 800$		$\lvert V\rvert = 900$	
	Gain	p	*Gain*	p	*Gain*	p	*Gain*	p	*Gain*	p	*Gain*	p	*Gain*	p
MAXCARD	-2	✗	-2	✗	-3	✗	-4	✗	-6	✗	-7	✗	-9	✗
MAXCARD-FAIR, $\beta = 1$	-1	✗	-1	✗	-1	✗	-2	✗	-3	✗	-3	✗	-5	✗
MAXCARD-FAIR, $\beta = 2$	+0		+0		+1	✓	+1	✓	+2	✓	+1		+1	
MAXCARD-FAIR, $\beta = 3$	+1	✓	+1	✓	+3	✓	+3	✓	+3	✓	+5	✓	+4	✓
MAXCARD-FAIR, $\beta = 4$	+1	✓	+2	✓	+3	✓	+4	✓	+4	✓	+5	✓	+5	✓
MAXCARD-FAIR, $\beta = 5$	+1	✓	+2	✓	+3	✓	+4	✓	+5	✓	+7	✓	+5	✓
MAXLIFE	-1	✗	-3	✗	-3	✗	-5	✗	-6	✗	-6	✗	-9	✗

Table 8.3: Median gains in expected total number of marginalized transplants under FUTUREMATCH. A ✓ or ✗ represents statistical significance (Wilcoxon signed-rank test, $p \ll 0.01$).

Table 8.2 shows that the objectives that do not regard fairness—MAXCARD and MAXLIFE—significantly beat myopic deterministic matching under the same objective. Interestingly, so too does MAXCARD-FAIR for low values of β. As β increases, the gain in *overall* number of transplants decreases (although it never drops below the deterministic matching algorithm with significance). This decrease in overall gain is incurred because marginalized patients, who (i) generally have lower in-degree, and (ii) have a higher probability of match failure, are being weighted more than easier-to-match pairs.

Table 8.3 explores this tradeoff between fairness and efficiency explicitly. For the fairness-agnostic and lightly fairness-preferring objectives, a relative loss of a few marginalized transplants is realized—although this loss of marginalized transplants is always less (typically much less) than the overall gain in transplants. Increasing the optimizer's preference for marginalized patients results in statistically significant gains in the number of marginalized transplants at *no* statistically significant loss in the overall expected number of transplants. In fact, for a middle ground around $\beta = 2$, FUTUREMATCH often shows statistically significant gains in *both* overall transplant and marginalized transplant counts—a clear win over myopia.

Our experiments support the following conclusions:

- FUTUREMATCH under MAXCARD and MAXCARD-FAIR with low $\beta = 1$ results in a significant increase in the *overall* number of transplants compared to myopic, at the cost of a smaller decrease in the number of *marginalized* transplants.
- FUTUREMATCH under MAXCARD-FAIR with high β results in a significant increase in *marginalized* transplants, at *no cost* to the overall number of transplants under myopic matching.
- For a middle ground around $\beta = 2$, FUTUREMATCH can result in both more overall expected transplants and more marginalized transplants.

We note that we are *not* making policy recommendations; rather, we are giving a proof of concept that our framework can effectively balance conflicting wants in an exchange. Indeed, the exact fairness quantification β that most effectively balances efficiency and fairness is a function of the underlying graph dynamics, which vertices are considered marginalized, and the ethical and legal wants of an exchange. All of these dimensions can be effectively encoded, validated, compared, and fielded through FUTUREMATCH.

8.3 Conclusions & future research

We presented FUTUREMATCH, a framework for learning to do complex matching in a general dynamic model. Motivated by our experience running the computational side of a large nationwide kidney exchange, we showed how to instantiate FUTUREMATCH to mimic an exchange under three different matching objectives and under two models of kidney exchange. We validated FUTUREMATCH on real data drawn from 94 match runs of the US nationwide exchange, between Oct. 2010 and Jan. 2014, and found that dynamic matching results in statistically significant increases in each of these objectives. Perhaps most critically, we showed that the framework yields better efficiency

and better fairness than deterministic myopic matching algorithms—which are the status quo class of algorithm in practice.

With regard to future research, any improvents made to any of the modules in the FUTUREMATCH framework would be both easily incorporated into the greater framework, and of potential real-world impact; indeed, since its initial presentation, FUTUREMATCH has been used to provide sensitivity analysis of matching policies at the UNOS exchange. For instance, while the method used for learning potentials in this chapter was an improvement over that used in the initial work of Chapter 6, convergence still was not reached—even for just the 20 blood type potentials. Recently, Kahng [124] adapted the FUTUREMATCH framework to a richer setting with both blood type potentials and those associated with a vertex's entrance time; given some knowledge of the vertex's eventual departure time, this allows FUTUREMATCH to become "timing aware" as well, hopefully matching those patients who are likely to perish before they do, indeed, leave the pool. That work finds that timing-aware matching can help reduce expected waiting time in the pool, but operated in a reduced 2-cycles-only setting in part because the learning process took "prohibitively long to converge" [124]. Better learning methods, like those discussed in the final section of Chapter 6, would be of great help in expanding FUTUREMATCH to a setting with more expressive potentials.

PART IV:

New Paradigms for General Organ & Barter Exchange

The whole is other than the sum of the parts.

– Kurt Koffka

9 Liver & multi-organ exchange

In this chapter, we address general organ exchange. The transplantation of organs from a deceased donor to a needy living candidate first occurred nearly sixty years ago, but only became popular in the 1970s due to the introduction of immunosuppressants that help prevent the rejection of foreign organs in a patient's body. Since then, the majority of transplantation has occurred through a deceased donor waiting list consisting of needy patients who wait for any willing donor to die, resulting in the harvesting and subsequent transfer of a compatible organ from the donor's cadaver to the living patient. There is a great supply shortage of cadaveric organs in most societies (including the US), and the imbalance between supply and demand keeps growing. As of July 5, 2015, there were 101,257 patients waiting for a kidney, 15,268 waiting for a liver, and 9073 for another organ (e.g., pancreas, joint pancreas-kidney, heart, lung, intestine) in the US alone.

In recent years, *live* donation of organs has significantly increased the total number of organ transplants. In live donation, a donor gives one of his two kidneys, one of his liver lobes, or a part of an intestine, etc., to the patient so both the donor and patient can live. The effect of live donation has been most prominent in kidney donation, where kidney exchange—which we discussed extensively in Parts I, II, and III of this thesis—has provided renewed hope to even "hard to match" patients.

We now explore the creation of living donor exchanges involving organs other than kidneys. We first explore large-scale *liver* exchange, which is similar to kidney exchange in some ways, but remains unexplored from a computational point of view.[1] The major difference between kidney and liver exchange rests in the increased risk to live donors, with very high rates of donor morbidity (24%), "near-miss" events in surgery (1.1%), and mortality (0.2%) compared to live donor kidney transplantation [58]. Fielded kidney exchanges derive significant value from *altruistic* donors, who

[1] Recently, small-scale liver exchanges have been manually arranged by medical professionals. In Korea, 16 candidates swapped *by hand* willing donors in a single hospital over the course of six years [115]; similarly, in Hong Kong, 2 candidates hand-swapped willing donors [57]. This shows the feasibility of the idea at a small scale [193].

enter the exchange without a paired needy candidate and trigger long "chains" of donations within the pool. With such a high risk of complication from surgery in liver transplantation, we expect significantly fewer (or no, if deemed unethical by the medical community) altruistic donors in liver exchange. The lack of altruistic donation, along with novel characteristics of the (in)compatibility of participants in a demographically-accurate liver exchange, leads to different matching behavior in theory and in practice.

With this in mind, we propose *multi-organ* exchange, where candidates in need of either kidneys or livers can swap donors in the same pool. We show theoretically that this combination provides linearly more transplants than running separate kidney and liver exchanges; this linear gain is a product of altruistic kidney donors creating chains that thread through the liver pool, and is present even when only a small but constant fraction of one side of the combined pool is willing to donate to a pair on the other side. We support this result experimentally on demographically accurate kidney, liver, and cross-organ exchanges. We conclude with thoughts regarding the fielding of a nationwide liver or joint liver-kidney exchange from a legal and computational point of view.

This chapter provides the first foray into the theory and computational methods necessary to set the groundwork for a fielded nationwide liver or multi-organ exchange. It is clear that such exchanges would be highly beneficial for sustaining life and creating value in society.

Related Publications

Some of the work in this section appeared at AAAI-14; the rest will appear in the Journal of Artificial Intelligence Research (JAIR). It is a collaboration between Dickerson and Sandholm [72].

9.1 Preliminaries

In order to develop a nationwide liver or multi-organ exchange, we must first accurately model the realities of such an exchange. We now extend the compatibility graph model, used in earlier sections of this thesis to represent a kidney-only exchange, to the case with two organs: kidneys and livers. Patient-donor pairs then enter the pool in need of either a kidney or a liver.

As before, begin by encoding an n-patient organ exchange as a directed graph. Construct one vertex for each incompatible candidate-donor pair. Add an edge e from one candidate-donor vertex v_i to another v_j, if the candidate at v_j can take a liver lobe or kidney from the donor at v_i, and is in need of the specific organ(s) being offered by the source vertex. We maintain that a matching be a collection of *disjoint* cycles; no vertex can give out more than one item (e.g., more than one kidney or liver lobe).

Due to significantly increased medical risk to living donors of livers, we do not expect many (or possibly any) altruistic donors outside of kidney exchanges [58]. We build on this assumption in depth throughout the chapter.

Figure 9.1—adapted from the kidneys-only compatibility graph given as Figure 1.1 in Section 1.3.1—gives an example organ exchange compatibility graph, where pairs on the left, shown with a dark boundary, have patients in need of a kidney while pairs on the right, shown with a light

boundary, have pairs in need of a liver. Possible cycles exist entirely in either the kidney or liver pool (for example, the 2-cycle $\langle (d_2 \to p_3), (d_3 \to p_2) \rangle$ and 2-cycle $\langle (d_5 \to p_6), (d_6 \to p_5) \rangle$, respectively) or between the two pools (for example, the 3-cycle $\langle (d_2 \to p_5), (d_5 \to p_3), (d_3 \to p2) \rangle$, which involves a single liver pair and two kidney pairs). Additionally, a single altruistic donor a exists in the pool and is willing to give his or her kidney—but not liver—to a patient, whose paired donor will then either donate a kidney *or* liver to a compatible patient in the pool (for example, via the chain $\langle (a \to p_1), (d_1 \to p_4), (d_4 \to p_7), (d_7 \to \cdot) \rangle$, with the final donor d_7 either donating to the deceased donor waiting list or remaining in the pool as a future altruistic donor).

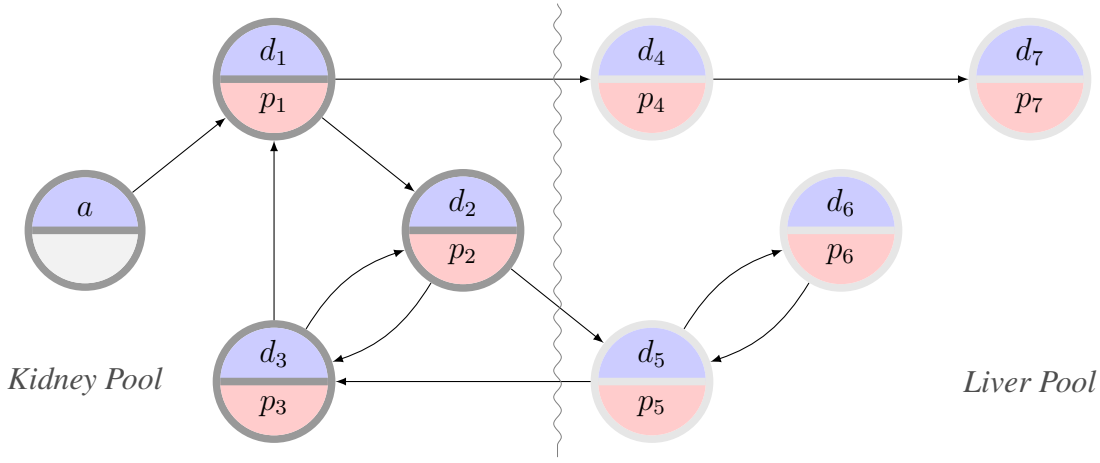


Figure 9.1: An example joint liver-kidney compatibility graph.

9.1.1 The clearing problem for multi-organ exchange

The clearing problem we wish to solve is still to find a maximum weight packing of cycles (possibly with some length cap L) and chains (possibly with or without a length cap K). In the small example compatibility graph shown in Figure 9.1, with $L = 3$, a maximum cardinality matching *without* chains includes five pairs via the 3-cycle and 2-cycle:

$$\{\langle (d_1 \to p_2), (d_2 \to p_3), (d_3 \to p_1) \rangle, \langle (d_5 \to p_6), (d_6 \to p_5) \rangle\}.$$

With chains—but maintaining two separate pools instead of combining the pools into a joint exchange—we can achieve a maximum cardinality matching with the same number of pairs but with a lower cycle cap ($L = 2$ instead of $L = 3$) via one chain and two 2-cycles:

$$\{\langle (a \to p_1), (d_1 \to \cdot) \rangle, \langle (d_2 \to p_3), (d_3 \to p_2) \rangle, \langle (d_5 \to p_6), (d_6 \to p_5) \rangle\}.$$

Finally, if we allow a combined liver-kidney exchange, the cardinality of the maximum matching increases to seven pairs. This is achieved by "threading" a chain started by the altruist a through a kidney-needing pair into the greater liver pool to match two previously unmatchable pairs, as well as by using the same two 2-cycles as before:

$$\{\langle (a \to p_1), (d_1 \to p_4), (d_4 \to p_7), (d_7 \to \cdot) \rangle, \langle (d_2 \to p_3), (d_3 \to p_2) \rangle, \langle (d_5 \to p_6), (d_6 \to p_5) \rangle\}.$$

We will explore the expected gains from combining exchanges theoretically in Section 9.2, and then via realistic simulation in Section 9.4.

9.1.2 Related work in multi-hospital kidney exchange

We are not the first to consider combining exchanges in general; rather, we are the first to consider combining *exchanges corresponding to different organs*, and we are the first to approach the intricacies of that combination from a theoretical and empirical point of view. Indeed, fielded centralized *kidney* exchanges typically consist of the merged pools of multiple participating hospitals. For example, the United Network for Organ Sharing (UNOS) kidney exchange includes 143 hospitals in the US. Each of these hospitals enters (possibly some subset of) the set of candidate-donor pairs and altruistic donors that have registered at their center into the centralized exchange, and then the exchange recommends a matching based on its clearing engine. Each hospital can be seen as having its own private exchange; then, questions can be asked about the possible gains in match efficacy based on the number or size of participating hospitals, or the truthfulness of their reporting.

Ashlagi et al. [25] look at the multi-hospital exchange problem from a game-theoretic point of view, where participating hospitals can manipulate the exchange by misreporting their private set of candidate-donor pairs. They show that in the worst case no deterministic strategy-proof mechanism can provide more than 1/2 of the truthful maximum matching, and no randomized strategy-proof mechanism can provide more than 7/8 of the truthful maximum matching. That bound was tightened for the two hospital case by Caragiannis et al. [54]. That model looks at 2-cycles only (represented as an undirected graph), while ours looks at 2- and 3-cycles and altruist-initiated chains. They also operate in a dense theoretical model only, while we give results in both dense and an arguably more realistic sparse model.

Ashlagi and Roth [19] and Toulis and Parkes [209] also analyze the multi-hospital exchange problem from a game-theoretic point of view. Ashlagi and Roth [19] show that, in general, a lack of participation of all hospitals can be very costly (although it is possible, at the cost of a few "lost" matches, to guarantee individual rationality and strategy proofness). Those results are in a dense model with both 2- and 3-cycles, but no chains. Toulis and Parkes [209] present a new multi-hospital mechanism for multi-hospital exchange and compares against their results.

As we do not consider incentive issues in the present work, perhaps most related to our work is a general matching result by Toulis and Parkes [209]. They show, in a dense kidney exchange model (which we overview in Section 9.2.2), that given m transplant centers, each with n candidate-donor pairs, the expected gain in number of matches to an individual hospital by entering a centralized exchange with full participation is roughly $\Omega(\sqrt{n})$. This result is in a model without chains; indeed, we will show that the inclusion of chains results in a linear gain in number of matches.

We are now ready to present the results of the chapter. Section 9.2 addresses liver and multi-organ exchange in adaptations of the two most common theoretical models of kidney exchange, and proves results in both models regarding the efficiency gains of multi-organ exchange over independent single-organ exchanges. Section 9.3 moves from theory to practice and presents our method of generating demographically accurate kidney, liver, and joint kidney-liver compatibility graphs. (Appendix D.1 provides more detail about this process, as well as a quantitative compar-

ison of the resulting graphs against the status quo kidney exchange generator.) Section 9.3 also presents the clearing algorithm we use to solve the multi-organ exchange problem in practice. Section 9.4 shows experimental results on liver and multi-organ exchanges, and gives strong empirical support to our earlier theoretical results showing that multi-organ exchange results in a greater number of matches than two independent exchanges. (Appendix D.2 presents additional experimental results.) We conclude in Section 9.5 with some thoughts on fielding a liver or multi-organ exchange, as well as future research directions.

9.2 Combining exchanges yields linearly more matches

In this section, we prove that combining independent liver and kidney exchanges leads to a linear gain in the aggregate number of matches. We do this in multi-organ adaptations of the two standard models of kidney exchange. Section 9.2.1 works in a newer sparse model adapted from Ashlagi et al. [23] (a similar model is used in Chapter 5), while Section 9.2.2 works in an older dense model like that presented by Ashlagi and Roth [19][2] (and used in, e.g., Chapters 2 and 7). We obtain similar—but not identical—theoretical results in both models.

9.2.1 Sparse model

We begin by adapting to the multi-organ exchange case a version of a recent random graph model for kidney exchange due to Ashlagi et al. [23]. They adapt sparse Erdős-Rènyi graphs to a model of kidney exchange with two classes of candidate: those with many incoming edges and those with very few incoming edges (intuitively, "easy-to-match" and "hard-to-match" candidates). That model mimics the basic structure of compatibility graphs seen in fielded kidney exchanges.

They build a random directed compatibility graph $D(n, \lambda, t(n), p_L, p_H)$ with n candidate-donor pairs, $t(n)$ altruistic donors, a fraction $\lambda < 1$ of the n candidate-donor pairs—representing *lowly-sensitized*, easy-to-match patients—who have probability p_L of an incoming edge from each vertex in the pool, and a fraction $1-\lambda > 0$ of the n candidate-donor pairs—representing *highly-sensitized*, hard-to-match patients—who have probability p_H of an incoming edge from each vertex in the pool. They assume $p_L > 0$ is constant, and $p_H = \frac{c}{n}$ for some constant $c > 1$; thus, the graph induced by only those $1 - \lambda$ fraction of (sensitized) vertices with incoming edge probability p_H is sparse.

We assume, for kidney exchange compatibility graphs D_K with n_K pairs, $t(n_K) > 0$; however, for liver exchange graphs D_L with n_L pairs, $t(n_L) = 0$ (i.e., there are no altruistic liver donors). Furthermore, we introduce an additional constant probability $p_{K \to L} > 0$ to address the likelihood that some paired kidney donors will be unwilling in any scenario to donate a liver instead of a kidney. For notational simplicity, we will not introduce the complementary probability $p_{L \to K}$, which would represent the probability that a paired liver donor would be willing to donate a kidney if matched, because in practice we believe any potential liver donor would prefer donating the

[2]While the most recent publication date of Ashlagi and Roth [19] is after that of Ashlagi et al. [23], the former paper appeared in 2011 as a conference paper, while the latter appeared as a conference paper in 2012 and is still under submission as a final journal paper.

vastly "easier" kidney to donating a liver (i.e., $p_{L\to K} = 1$ in practice). Still, were this not to be the case, the qualitative results to follow would still hold. We do not assume that the p_L (resp. p_H) for D_K equals the p_L (resp. p_H) for D_L. Formally, let $p_{L,\{K,L\}} \in (0, 1]$ be different constants and $p_{H,\{K,L\}} = c_{\{K,L\}}/n_{\{K,L\}}$ for D_K and D_L and positive constants c_K and c_L. When the usage is obvious from context, for expositional ease we will still use p_H, p_L, and c.

Now, define the graph join operator $D = join(D_K, D_L, p_{K\to L})$ between a kidney exchange graph D_K and liver exchange graph D_L as follows. Flip a $p_{K\to L}$-weighted coin for each patient-donor pair in D_K; if heads, this pair is willing to give a liver to a pair in D_L if matched (or a kidney to a different pair in D_K), otherwise the paired donor is only willing to give a kidney. Next, add directed edges between candidate-donor pairs in both pools in accordance with each pair's associated probability (e.g., $p_{L,L}$ from any kidney pair to a lowly-sensitized liver pair or $p_{H,K}$ from any liver pair to a highly-sensitized kidney pair), *except* for those vertices with paired kidney donors who are unwilling to donate livers. Do not add any edges from the $t(n_K)$ altruistic donors in D_K to vertices in D_L (since altruistic kidney donors are unwilling to donate a liver).

In the following theoretical results, we consider cycles of length at most some constant but chains of any length; this mimics current practice in kidney exchange, and would likely mimic that of fielded liver exchange. Thus, an *efficient matching* allocates the maximum number of transplants in cycles of size no more than some constant and chains of any length. Both results build on the work of Ashlagi et al. [23], which considers only a single kidney exchange.

Proposition 7 assumes a linear (in the number of candidate-donor pairs) number of altruistic donors, while Proposition 8 works with just a constant number of altruistic donors. We contrast both theoretical results at the end of this section.

Proposition 7. *Consider $\beta > 0$ and $\gamma > 0$, sparse kidney compatibility graph D_K with n_K pairs and $t(n_K) = \beta n_K$ altruistic donors, and sparse liver compatibility graph D_L with $n_L = \gamma n_K$ pairs. Then for any constant cycle cap and $p_{K\to L} > 0$, any efficient matching on $D = join(D_K, D_L, p_{K\to L})$ matches $\Omega(n_K)$ more pairs than the aggregate of any such efficient matchings on D_K and D_L (with probability approaching 1 as n_K approaches ∞).*

Proof. The proposition follows from the proof of Theorem 5.4 of Ashlagi et al. [23], which directly supports a similar result as Theorem 5.2 of Ashlagi et al. [23]. In that Theorem 5.4 (which assumes a kidney exchange graph similar to ours, with no altruistic donors), they show that there are a linear in n ($n_{\{K,L\}}$ for us) number of "good cycles" of some constant length z. These "good cycles" have a single vertex u in the lowly-sensitized portion of $D_{\{K,L\}}$ that is *only* connected to a single vertex v_1 in the highly-sensitized portion of $D_{\{K,L\}}$ (and possibly other vertices in the lowly-sensitized portion). From v_1 there then exists a path $\langle v_1, \ldots, v_{z-1}\rangle$ of highly-sensitized vertices with out- and in-degree one such that v_{z-1} connects back to u. Finding that path $\langle v_1, \ldots, v_{z-1}\rangle$ relies on a well-known result (see, e.g., [120]) that there exist linearly many isolated tree-like structures in a sparse graph (like the one induced by our highly-sensitized vertices). They show an additive linear gain in increasing cycle caps by first taking some optimal cover of cycles of length at most z and augmenting it to include enough of these "good cycles" of length at most $z + 1$—of which there are linearly many in n—resulting in the gain.

We assume a constant cycle cap of z and no chain cap, which mimics real-world kidney exchanges and would probably be the case in a fielded liver exchange (if altruistic donors existed).

Note that regardless of cycle cap, any efficient matching will match all lowly-sensitized pairs (w.h.p. as n grows), via direct application of well-known matching results on dense Erdős-Rènyi graphs (see, e.g., [120]). Under this constant cycle cap assumption, there exist a linear number of highly-sensitized vertices in the liver pool D_L that remain unmatched by an efficient matching of cycles of length at most z (recall there are no chains in the liver pool). These are the linearly many isolated highly-sensitized paths that are part of "good cycles" of length strictly greater than z and thus cannot be legally matched. By gluing the two pools D_L and D_K together, these isolated vertices gain access—through chains that start in D_K, of which there are linearly many in n_K—to a linear number of altruists who, as in Theorem 5.2 of Ashlagi et al. [23], act as the u vertex in "good cycles" of length greater than z that are now no longer required to connect back to u.

Formally, fix an efficient matching M_K^* in D_K alone and an efficient matching M_L^* in D_L alone, which is disjoint from M_K^* (by construction). The aggregate size $m_I = |M_K^* \cup M_L^*| = |M_K^*| + |M_L^*|$ of these two matchings is the size of the efficient matching under the setting of independent liver and kidney exchanges. We will show that by combining exchanges, all pairs matched in $M_I^* = M_K^* \cup M_L^*$ can be matched *while also matching a linear number of previously unmatched pairs*.

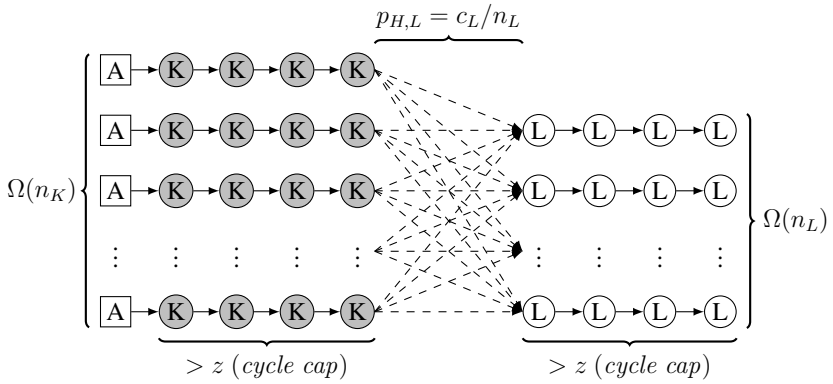


Figure 9.2: A linear number of chains threading into the liver pool. Altruistic donors are shown as a boxes, while pairs in D_K and D_L are shown as circles with inscribed Ks and Ls, respectively. Pairs that are matched *only* when exchanges are combined are shown in white.

Figure 9.2 overviews the augmented matching we will construct. On the left side is a linear-in-n_K number a of chains created from the $t(n_K)$ kidney altruists and only pairs in D_K; these structures exist and are already in the efficient matching M_K^*—and thus in M_I^* [23]. Now, for each of the a chains, consider the final kidney pair in the chain. By assumption, with constant positive probability $p_{K \to L}$, that pair is willing to participate in a donation that crosses into the liver pool. This preference is determined independently across all pairs, so in expectation $(p_{K \to L})a$ chains will be willing to thread into the liver pool. We will extend these chains in our combined matching, and will leave the remaining $(1 - p_{K \to L})a$ in expectation kidney-only chains allocated as they were in the original matching M_K^*. These estimates are concentrated about their mean via standard concentration bounds.

Given the original constant cycle cap z, for any larger constant integer $z' > z$, there exist

$b = \beta n_L \in \Omega(n_L)$ isolated paths in the liver pool as well, where β is a positive constant [23]; these are shown on the right side of the graph in Figure 9.2. These consist entirely of pairs that are not matched in M_L^*, and thus are not matched in M_I^*, either. Now, for each of those $(p_{K\to L})a$ kidney pairs at the end of a chain in M_K^* who are willing to donate a liver, the probability p_1 that at least one edge exists from that pair to at least one liver pair at the head of one of the βn_L isolated paths of length z' in D_L is:

$$p_1 = 1 - (1 - c_L/n_L)^{\beta n_L} =_{n_L\to\infty} 1 - e^{-c_L\beta} \in O(1). \tag{9.1}$$

This connectivity is determined independently for each of the $(p_{K\to L})a$ kidney pairs, leading to $p_1(p_{K\to L})a \in \Omega(n_K)$ in expectation pairs with at least one connection to a liver pair at the head of an isolated path in D_L. Edges that may exist between the two sets of paths—the chains in M_K^* and the isolated unmatched paths in D_L—are shown as dashed lines in Figure 9.2. Then all that is left to do is lower bound the size of a maximum bipartite matching on the graph induced by those dashed edges that actually exist, which we denote by G. We prove this bound with a balls and bins argument.

As a lower bound on the size of that matching, assume that each of the $p_1(p_{K\to L})a$ kidney pairs with at least one edge crossing into the liver pool has *exactly* one edge crossing into the liver pool, selected uniformly at random from its true set of edges. This induces an injective mapping of kidney pairs to liver pairs, and also a subgraph $G' \subset G$ of the full bipartite graph over which we are performing a maximum matching; thus, the cardinality of a maximum matching in this reduced subgraph G' is a lower bound on the cardinality of a maximum matching in the fully realized bipartite graph G. Furthermore, the size of a maximum matching in this subgraph is equal to the number of pairs on the liver side with at least one incoming edge. We calculate that now, treating each kidney pair as one of $p_1(p_{K\to L})a$ balls being dropped uniformly at random into one of the βn_L bins (representing a liver pair with at least one incoming edge in the full bipartite graph G).

Formally, index the liver pairs $[I] = \{1, 2, \ldots, \beta n_L\}$, let random variable Y represent the number of liver pairs with zero incoming edges in the subgraph G', and let X_i be a binary random variable that is set if pair $i \in [I]$ has no incoming edges. Then $\mathbb{E}[X_i] = (1 - 1/\beta n_L)^{p_1(p_{K\to L})a} \in O(1)$. Thus, with constant $\mathbb{E}[X_i] < 1$, and $\mathbb{E}[Y] = \mathbb{E}[\sum_{i\in[I]} X_i] = \sum_{i\in[I]} \mathbb{E}[X_i]$ by linearity of expectation, we have $E[Y] = \beta n_L E[X_i]$, a constant fraction (strictly less than one) of the liver pairs.

Finally, let random variable Z represent the number of liver pairs with at least one incoming edge in the subgraph. Then $\beta n_L = Z + Y$, and $\mathbb{E}[Z] = \beta n_L - E[Y] = \beta n_L(1 - E[X_i]) \in \Omega(n_L)$. With $\mathbb{E}[Z]$ a constant positive fraction of the liver pairs, we have a maximum matching of size $\Omega(n_L)$ in G', and thus a maximum matching of size $\Omega(n_L)$ in G.

We now construct our final matching M_C^* in the combined pool. Take $M_C^* = M_I^*$. For those kidney-only chains in $M_K^* \subseteq M_I^*$ ending in (i) a kidney pair willing to donate a liver with (ii) at least one edge into the liver pool, add edges in accordance with a fixed maximum matching in G. Each of these $\Omega(n_L)$ edges connects to an isolated path of length $z' > z \geq 0$ of *previously unmatched* pairs in the liver pool. Add each of these pairs to M_C^*, resulting in a linear gain in matching size over M_I^*.

□

As shown in Proposition 7, the presence of a linear number of altruistic kidney donors in a multi-organ exchange results in a linear gain in the overall number of pairs matched in an efficient matching relative to the aggregate of efficient matchings in independent kidney and liver exchanges. This is realized specifically by giving those highly-sensitized liver pairs who are unmatchable using only cycles in the liver pool access to more flexible, longer altruist-initiated chains that thread out of the kidney pool. In the following Proposition 8, we restrict the number of altruistic kidney donors to a constant and show that even in this constrained setting, with constant positive probability, chains that thread out of the kidney pool into the liver pool allow for a linear gain in overall number of matches.

Proposition 8. *Consider $\gamma > 0$, sparse kidney compatibility graph D_K with n_K pairs and constant $t > 0$ altruistic donors, and sparse liver compatibility graph D_L with $n_L = \gamma n_K$ pairs. Then there exists $\lambda' > 0$ such that for all sensitization probabilities $\lambda < \lambda'$, for any constant cycle cap and $p_{K \to L} > 0$, any efficient matching on $D = join(D_K, D_L, p_{K \to L})$ matches $\Omega(n_K)$ more pairs than the aggregate of any efficient matchings on D_K and D_L separately (with constant positive probability).*

Proof. For small enough λ and large enough c_K, where $p_{H,K} = c_K/n_K$, with high probability there exists a set S_K (of size at least $n_K/2$) of highly-sensitized pairs in D_K that are "too far" away from lowly-sensitized pairs in D_K to be matched in a cycle of capped length and must be matched in a chain triggered by an altruist a or not matched at all [23]. By similar reasoning, for large enough c_L, where $p_{H,L} = c_L/n_L$, there exists a set S_L of pairs in D_L that cannot be matched in a cycle of constant capped length and would have to be matched in a chain or not matched at all—since we assume that no altruists willing to directly donate a liver exist, these pairs would go unmatched if exchanges for different organs operated independently.

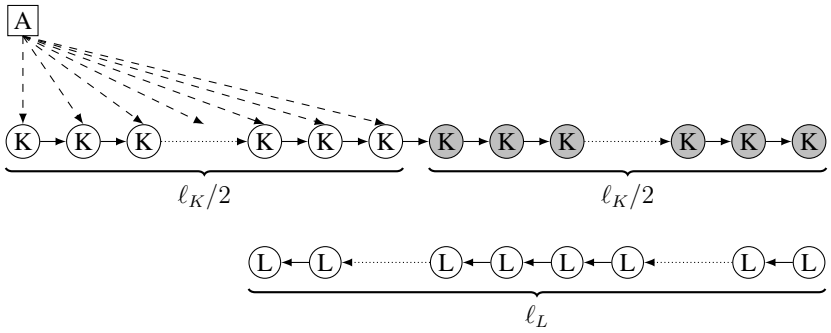


Figure 9.3: Relevant portion of the maximum matching for $S_K \subseteq D_K$ and $S_L \subseteq D_L$ in the *independent* exchanges case. An altruist is shown as a box, while pairs in S_K and S_L are shown as circles with inscribed Ks and Ls, respectively. Pairs that are matched with constant positive probability are shown in gray. Note that no pairs in S_L are matched.

To aid the reader, Figures 9.3 and 9.4 accompany the statements in this part of the proof; Figure 9.3 corresponds to our status quo case of two separate kidney and liver exchanges, while Figure 9.4 corresponds to the setting of this proof, where the exchanges are combined via the *join*

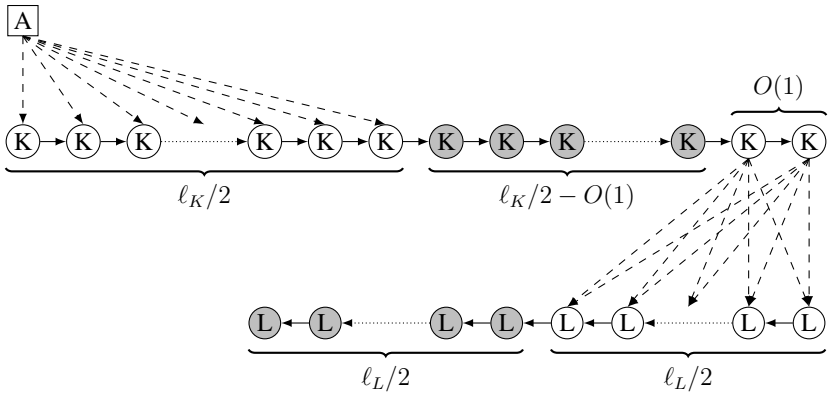


Figure 9.4: Relevant portion of the maximum matching for $S_K \subseteq D_K$ and $S_L \subseteq D_L$ in the *combined* exchange case. An altruist is shown as a box, while pairs in S_K and S_L are shown as circles with inscribed Ks and Ls, respectively. Pairs that are matched with constant positive probability are shown in gray. Note that a linear portion of the chain in S_L is matched with constant positive probability.

operator. We will make use of a general result on sparse random directed graphs from Krivelevich et al. [138]: as $c_{\{K,L\}}$ increases, a directed path of length approaching $|S_{\{K,L\}}|$ in $S_{\{K,L\}}$ exists.

We first show the existence of long paths of vertices that must be matched by chains in each of D_K and D_L. Take the set $S_K \subseteq D_K$ of kidney pairs that must be matched via chains or not at all; then $|S_K| \geq n_K/2 \in \Omega(n_K)$ [23]. Similarly, take the set $S_L \subseteq D_L$ of liver pairs that must be matched via chains (threaded, in this case, through a willing kidney pair as determined by $p_{K\to L}$); then $|S_L| \geq n_L/2 \in \Omega(n_K)$. Via the results of Krivelevich et al. [138], there exists a directed chain C_K of length $\ell_K \in \Omega(|S_K|)$ in S_K, and similarly there exists a directed chain C_L of length $\ell_L \in \Omega(|S_L|)$ in S_L. Then there exists constants α_K and α_L such that $\ell_K = \alpha_K n_K$ (the length of the chain in S_K) and $\ell_L = \alpha_L n_L$ (the length of the chain in S_L), respectively.

Take the first $\ell_K/2$ pairs in the head of C_K; then, the probability p_1 that a given altruistic donor has at least one outgoing edge to a pair in that head is

$$p_1 = 1 - (1 - c_K/n_K)^{\ell_K/2} = 1 - (1 - c_K/n_K)^{\alpha_K n_K/2} =_{n_K\to\infty} 1 - e^{-c_K\alpha_K/2} \in O(1). \quad (9.2)$$

Then, with constant positive probability p_1, the altruistic donor in the *independent exchanges* case matches m_I pairs, where $\ell_K/2 < m_I \leq \ell_K$; exactly 0 pairs in the chain C_L are matched. That matching is visualized in Figure 9.3.

In the *combined exchanges* case, a given pair in C_K has an outgoing edge to a given pair in C_L with probability $p_{K\to L}c_L/n_L > 0$. Take a positive constant $t > 0$ number of pairs in the tail of the C_K chain. Then, the probability p_2 that at least one of the t pairs in that tail has at least one outgoing edge to at least one pair in the first $\ell_L/2$ pairs at the head of the liver chain C_L is

$$\begin{aligned} p_2 = 1 - \left[(1 - p_{K\to L}c_L/n_L)^{\ell_L/2}\right]^t &= 1 - \left[(1 - p_{K\to L}c_L/n_L)^{\alpha_L n_L/2}\right]^t \\ &=_{n_L\to\infty} 1 - \left[e^{-p_{K\to L}c_L\alpha_L/2}\right]^t \in O(1). \end{aligned} \quad (9.3)$$

The independence assumptions above are valid because (i) the willingness of a kidney pair to give a liver is determined independently via the $p_{K\to L}$ parameter and (ii) the initial edge compatibility check between pairs in D_K and D_L is independent of the results of the $p_{K\to L}$ coin flip. Then, with constant positive probability $p_1 p_2$, the altruistic donor in the *combined exchanges* case matches m_C pairs, where $m_C > m_I + \ell_L/2 - t$, the guaranteed matched pairs in C_K minus a constant t pairs in that tail plus the guaranteed matched pairs in the tail of C_L. That matching is visualized in Figure 9.4.

Recall $\ell_L/2 \in \Omega(|S_L|)$, and with constant t, $\ell_L/2 - t \in \Omega(|S_L|)$. So, the gain in matches between the matching in Figure 9.3 and that in Figure 9.4 is

$$m_C - m_I > \ell_2/2 - t = \alpha_L n_L/2 - t = \alpha_L \gamma n_K/2 - t \in \Omega(n_K) \tag{9.4}$$

which occurs with constant probability at least $p_1 p_2 > 0$, where p_1 and p_2 are given in Equations 9.2 and 9.3, respectively. □

Discussion of theoretical results in the sparse model

Intuitively, Propositions 7 and 8 show the theoretical efficacy of combining kidney exchange with alternate organ exchanges (where altruistic donation is less likely to be popular or deemed ethically acceptable). While Proposition 8 may seem like a stronger result due to its relaxed reliance on a constant number of altruistic kidney donors (instead of the linear number in Proposition 7), the numerator c in $p_H = c/n$ may be required to be quite large (although still constant), the λ sensitivity constant quite small, and the result also holds with merely constant positive probability instead of holding with probability approaching one. We feel this makes Proposition 7 a more relevant result overall than Proposition 8 for the composition (in terms of pool sensitization and number of altruistic donors) of currently fielded kidney exchanges.

9.2.2 Dense model

Initial research on random graph models for organ exchange adapted dense (constant probability of an edge existing) Erdős-Rènyi graphs to kidney exchange [19, 74]. Fielded exchanges have proven to be somewhere in between dense—as we discuss now—and sparse—as in the theory above—in practice, and thus actual pools and their optimal matchings do not align with these dense models [23, 24, 75, 78]. Still, we show that the efficiency results in the dense model with chains (Theorem 1 of Chapter 2) can be applied to independent liver exchange and multi-organ exchange to yield efficient matchings with linear expected overall gain from combining the pools (given a linear number of altruists) for large enough compatibility graphs. We derive these results now.

We begin by refreshing the reader on some terminology from the dense model of kidney exchange, which models blood types for patients and donors. An *under-demanded* pair is any pair such that the donor is not ABO-compatible with the patient. If an under-demanded pair contains only type A and B blood (e.g., a pair with A-type patient and B-type donor, or vice versa), it is called *reciprocal*. Any pair in the pool such that the donor is ABO-compatible with the candidate is called *over-demanded*. Furthermore, if a donor and candidate share the same blood type, they

are a *self-demanded* pair. Under-demanded and reciprocal pairs are intuitively "harder" to match than over-demanded and self-demanded pairs. This is not necessarily the case if sensitization, the probability of matching with a random donor, is considered. For example, an A-type patient who is lowly sensitized is typically easier to match than an O-type patient who is highly-sensitized; however, the dense model does not consider different degrees of sensitization. The dense model critically assumes that a donor and patient who are blood-type compatible are tissue type incompatible with *constant* probability $\bar{p}$. This differs from the model we used in Propositions 7 and 8, where lowly-sensitized patients had a constant edge probability while highly-sensitized patients did not. The dense model also denotes by μ_X the frequency of blood type X, and assumes $\mu_{\text{O}} < 3\mu_{\text{A}}/2$ and an ordering $\mu_{\text{O}} > \mu_{\text{A}} > \mu_{\text{B}} > \mu_{\text{AB}}$. The United States national blood type distribution satisfies these constraints. As in Chapter 7, we will use $V^{X\text{-}Y}_{\{K,L\}}$ to refer to the subset of vertices with patient and donor of blood type X and Y, respectively, in the kidney and liver compatibility graphs, and $V^{X}_{\{K,L\}}$ for the subset of vertices with altruistic donors with blood type X in the kidney and liver compatibility graphs.

Under the realistic assumptions on blood type distributions stated above, but assuming no chains and only patients who need kidneys, Proposition 5.1 of Ashlagi and Roth [19] states that an efficient allocation exists (with high probability) that uses only cycles of length at most 3. That result is proven only with respect to cycles; that is, it assumes there are no altruistic donors. Theorem 1 of Chapter 2 extends that result into a pool that also has chains (but still only patients who need kidneys), stating that an efficient allocation exists (with high probability) using only cycles and chains of length at most 3. Both of these results are "in the large" and rely on the fact that the size of a set $S \in \{V^{X\text{-}Y}_{\{K,L\}}\} \cup \{V^{X}_{\{K,L\}}\}$ for any blood types X and Y will be very close to its expectation as $|S| \to \infty$.

Only livers

We first look at dense *liver* exchange in this model. The blood type distributional requirement is satisfied by patients in need of livers, just as it is with patients who need kidneys. Thus, under the dense model, a liver-only compatibility graph looks exactly the same as a kidney-only compatibility graph (albeit with no chains). Thus, the efficiency result of Ashlagi and Roth [19] can be applied directly to liver-only compatibility graphs. If altruistic liver donors existed in a liver-only compatibility graph, then Theorem 1 would be directly applicable instead.

Multi-organ exchange

Next, we consider dense *multi-organ* exchange in this model. In this model, there will exist altruistic donors willing to give a kidney but not a liver, as motivated earlier in the chapter.

We assume the same blood type distribution and ordering as Ashlagi and Roth [19] for both liver and kidney patients and donors. We also assume a directed multi-organ dense compatibility graph D, with n_K pairs needing a kidney and $n_L = \gamma n_K$ pairs needing a liver, for some constant $\gamma > 0$. As motivated earlier in this chapter, altruistic kidney donors will not donate directly to liver patients, but may trigger chains that result in a kidney pair donating to a liver pair. Each kidney pair is willing to give to a compatible liver pair with some constant probability $p_{K \to L} > 0$. Thus, there

are no outgoing edges in D from altruistic kidney donors to pairs needing a liver, but potentially some edges from kidney pairs to compatible liver pairs.

In Proposition 9, we show that if there are enough altruistic kidney donors, the size of an efficient matching on D is larger by an additive linear fraction than the size of the aggregate of efficient matchings on D_L and D_K, the subgraphs induced by only the vertices consisting of pairs needing livers and kidneys, respectively. We achieve this linear gain via a similar high-level strategy to what was used in Propositions 7 and 8, which threaded kidney-altruist-initiated chains through willing non-altruist kidney pairs into the liver pool. Formally, let V_K^X be the subset of vertices in D_K containing only altruistic kidney donors of blood type $X \in \{\text{O}, \text{A}, \text{B}, \text{AB}\}$.

Proposition 9. *Consider $\beta^A = \mu_A\mu_{AB}$, $\beta^B = \mu_B\mu_{AB}$, constants $\gamma > 0$ and $p_{K\to L} > 0$, dense kidney compatibility graph D_K with n_K pairs, and dense liver compatibility graph D_L with $n_L = \gamma n_K$ pairs. If at least one of $|V_K^A| > \beta^A n_K$ or $|V_K^B| > \beta^B n_K$, then any efficient matching on $D = join(D_K, D_L, p_{K\to L})$ matches $\Omega(n_k)$ more pairs than the aggregate of any such efficient matchings on D_K and D_L (with probability approaching 1 as $n_K \to \infty$).*

Proof. We begin by adopting vocabulary from Ashlagi and Roth [19]; specifically, if a vertex v participates in an exchange with some under-demanded vertex v', then we say v *helps* v'. Pairs denoted by X-Y have X-type patients and and Y-type donors, for $X, Y \in \{\text{O}, \text{A}, \text{B}, \text{AB}\}$. Note that AB-altruists cannot help under-demanded pairs, A- and B-altruists can only help A-AB and B-AB under-demanded pairs, respectively, and O-donors can trigger two types of chains of length 3 containing under-demanded pairs: ⟨O-altruist, O-A pair, A-AB pair⟩ or ⟨O-altruist, O-B pair, B-AB pair⟩.

First, take the efficient matching result of Ashlagi and Roth [19] and apply it to D_L. Only (some) under-demanded liver vertices remain unmatched. Second, apply the efficient matching result of Theorem 1 to D_K. Again, only (some) under-demanded kidney vertices remain unmatched.

Figure 9.5 provides a visual representation of the full allocation we will construct by augmenting the two allocations mentioned above.[3] Altruists are shown as rectangles and candidate-donor pairs as ovals; over-demanded pairs are gray; under-demanded and self-demanded pairs are white; and reciprocal pairs are black. Solid edges represent donations that are in the original allocations, while dashed edges are those added by the allocation we generate for the joint pool.

As in Theorem 1, since applying the two initial matchings results in all over-demanded, self-demanded, and reciprocally-demanded pairs being matched (assuming $|S|$ approaches its expectation as $|S| \to \infty$ for any set $S \in V_{\{K,L\}}^{X\text{-}Y}, X, Y \in \{\text{O}, \text{A}, \text{B}, \text{AB}\}$), we must only exhaustively consider all ways of matching under-demanded pairs. We do this in the list below: bolded items trigger a linear gain in the combined efficient match, while all other items show no efficiency loss. This guarantees a linear gain overall.

- *AB-altruists:* Altruistic AB-donors can only help over- and self-demanded (AB-AB) pairs, both of which are matched entirely already in the separate exchanges.

[3] Figure 9.5 shows the full allocation up to symmetries between A-B and B-A pairs. By assumption, $\mathbb{E}[|V_K^{\text{A-B}}|] = \mathbb{E}[|V_K^{\text{B-A}}|]$ and $\mathbb{E}[|V_L^{\text{A-B}}|] = \mathbb{E}[|V_L^{\text{B-A}}|]$, but it could be the case in practice that one subgraph is larger than the other. We assume WLOG in Figure 9.5 that $|V_K^{\text{A-B}}| \geq |V_K^{\text{B-A}}|$ and $|V_L^{\text{A-B}}| \geq |V_L^{\text{B-A}}|$.

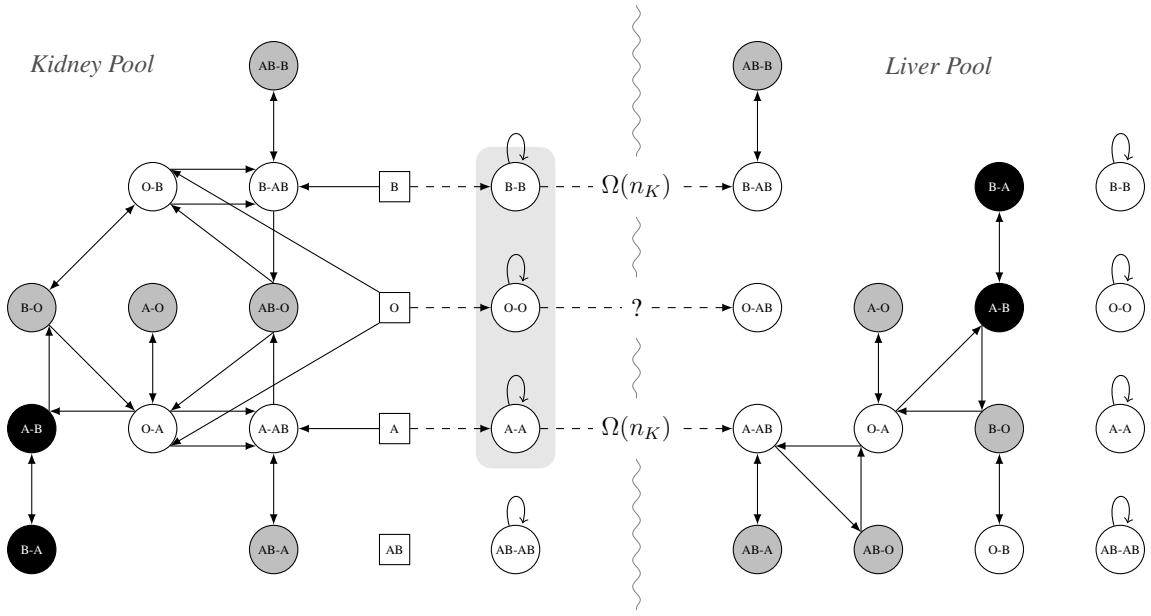


Figure 9.5: Our constructed matching that directly combines the allocations of Theorem 1 and Ashlagi and Roth [19]—which it applies initially to the kidney pool and liver pool, respectively—and then threads leftover altruistic kidney donors through the kidney pool into unmatched portions of the liver pool.

- **A-altruists:** Of the under-demanded pairs, altruistic A-donors can only help A-AB pairs.[4] In the matching from Theorem 1, A-donors donate to the A-AB pairs until one of the two sets is exhausted. Under our assumption, $|V_K^{\text{A}}| > \mu_{\text{A}}\mu_{\text{AB}}n_K = |V_K^{\text{A-AB}}|$, so the A-AB set will be exhausted, leaving some A-donors unallocated. These remaining A-donors can be threaded into the liver pool through A-A kidney pairs to match with the remainder of under-demanded A-AB liver pairs.

  Given constant probability $p_{K\to L} > 0$ of a non-altruist kidney donor being willing to donate a liver, and constant probability $\bar{p} > 0$ of an otherwise blood type compatible pair being tissue type incompatible, a constant fraction of the pairs in $V_K^{\text{A-A}}$, specifically $p_{K\to L}|V_K^{\text{A-A}}| = p_{K\to L}\bar{p}\mu_A\mu_A n_K$ pairs in expectation, are willing to give a liver to a liver pair. The use of an A-A kidney pair via an A-altruist-initiated chain results in 0 efficiency loss relative to the initial efficient matching, since there remains a perfect matching in $V_K^{\text{A-A}}$ by well-known results on dense Erdős-Rènyi graphs (see, e.g., [120]). Thus, we gain 1 match for each of the remaining

$$\min\left\{p_{K\to L}|V_K^{\text{A-A}}|, |V_K^{\text{A}}| - |V_K^{\text{A-AB}}|\right\} \tag{9.5}$$

  A-donors. The first input in the minimization in Equation 9.5 is of size $\Omega(n_K)$, because a constant fraction of a set that is linear in n_K is still linear in n_K. The second input is never

[4] In the original compatibility graph, altruistic A-donors can also help reciprocal A-B pairs; however, by the earlier applications of the efficient matchings due to Ashlagi and Roth [19] to the liver pool and the efficient matching from Theorem 1 to the kidney pool, all reciprocal pairs are already matched.

negative and is potentially $\Omega(n_K)$ by the theorem statement's assumption on the number of altruistic A-donors or B-donors; we address this uncertainty in the next paragraph.

- **B-altruists:** Of the under-demanded pairs, altruistic B-donors can only help B-AB pairs. Under a symmetric argument as the A-donors above, combining pools yields

$$\min\left\{p_{K\to L}|V_K^{\text{B-B}}|, |V_K^{\text{B}}| - |V_K^{\text{B-AB}}|\right\} \tag{9.6}$$

additional matches by threading through willing B-B kidney pairs into the unallocated under-demanded liver pool. By similar logic to the above A-donor case, the first input to the minimization in Equation 9.6 is again assuredly $\Omega(n_K)$, while the second is never negative and is potentially $\Omega(n_K)$. The theorem statement ensures that at least one of the sets of A-donors or B-donors is large enough—specifically, $|V_K^{\text{A}}| > \mu_{\text{A}}\mu_{\text{AB}}n_K$ or $|V_K^{\text{B}}| > \mu_{\text{B}}\mu_{\text{AB}}n_K$—to trigger a linear gain in at least one of Equations 9.5 or 9.6.

- *O-altruists:* In the matching of Theorem 1, some O-donors may be used in 2-chains with remaining under-demanded pairs in D_K. It is possible that these O-donors could be threaded through an under-demanded kidney pair into an under-demanded liver pair to form a 3-chain at utility gain of 1 (but not necessary for this proof). Similarly, because we are not making assumptions on the number of O-donors, if there are so many O-donors in D_K that all under-demanded pairs (e.g., pairs of type O-AB in D_K) are matched, then these O-donors can be threaded directly into the liver pool by way of self-demanded O-O kidney pairs who are willing to give livers (at no efficiency loss, as a perfect matching will remain in $V^{\text{O-O}}$) for a gain of at least 1 under-demanded match in D_L (but this is also not necessary for this proof).
- *Non-altruistic (i.e., paired) vertices:* Self-demanded and reciprocally-demanded pairs cannot help under-demanded pairs without involving altruistic donors or over-demanded pairs. AB-O vertices are the only pairs that can help at most two under-demanded pairs (either O-A and A-AB, or O-B and B-AB). In the Theorem 1 allocation, most AB-O pairs are used in 3-cycles with two under-demanded pairs; however, some may be used in 2-cycles with a single under-demanded pair. Reallocating these are not necessary for this proof.

Since at least one of the minimum size constraints on the set of altruistic A-donors ($|V_K^{\text{A}}| > \mu_{\text{A}}\mu_{\text{AB}}n_K$) or B-donors ($|V_K^{\text{B}}| > \mu_{\text{B}}\mu_{\text{AB}}n_K$) is satisfied by the proposition statement's assumptions, we are guaranteed $\Omega(n_K)$ additional matches by combining both pools by way of Equations 9.5 and 9.6. □

The theoretical results presented in this section motivate the combination of independent kidney and liver exchanges and show that such a joint exchange would allow for the use of altruistic kidney donors at great gain to overall social welfare. Still, both models are significant simplifications of real organ exchange; the push for a fielded liver or multi-organ exchange in reality will require extensive *realistic* simulations showing expected gains in number of matches, among other statistics. We address this in the rest of the chapter. Section 9.3 describes our method for generating and clearing demographically accurate bi-organ compatibility graphs and Section 9.4 presents experimental results on (i) liver exchange alone and (ii) independent liver and kidney exchanges versus a combined multi-organ exchange.

9.3 Generating and clearing demographically accurate pools

In this section, we describe our method for generating realistic organ exchange graphs for programs in steady state, and compare these generated graphs to those produced by the current status quo steady-state kidney exchange generator. We also briefly describe a generator built for early-stage exchanges that have not yet reached steady state; this generator draws from real data from the United Network for Organ Sharing (UNOS) nationwide kidney exchange. We then describe the standard kidney exchange clearing algorithm and, motivated by generated realistic steady-state liver and kidney exchange graphs, present a tweak to this algorithm to decrease liver exchange solution time.

9.3.1 Data generation

In order to create an at-scale nationwide liver or multi-organ exchange, we first have to develop a compatibility graph generator with which we can run simulations. First, we draw data from reliable sources (here, specific to the US). Second, this data is fed into a graph creation algorithm that probabilistically determines the existence of compatible and incompatible candidate-donor pairs, as well as compatibility constraints between different candidate-donor pairs. In the large, with high probability, graphs generated by this algorithm will mimic the demographics that would prevail in a large-scale fielded exchange in the US. (Plugging different raw data (e.g., age, weight, blood type distributions) into the generator algorithm would provide realistic generation of non-US compatibility graphs.) These graphs will mimic organ exchange in *steady state*; in Section 9.3.3, we will briefly describe the differences in compatibility graph composition that we have witnessed in the creation of a nascent kidney exchange.

For our dense graph experiments, we generate kidney exchange compatibility graphs in accordance with Saidman et al. [191]; however, the compatibility of a potential liver donor with a candidate differs from that of a potential kidney donor in three critical ways. While a donor and candidate must be blood-type (ABO) compatible, (a) they need not be HLA-compatible,[5] (b) the age of the donor and candidate makes a significant difference in transplant success [86], and (c) the portion of the donor liver that is cut out and transplanted into the candidate must be large enough to keep the candidate alive, while the remainder of the liver in the donor must be large enough to keep the donor alive. A proxy for liver size is the weight of the candidate or donor; intuitively, larger people need larger livers. Thus, we assume a donor must be at least as heavy as his or her matched candidate (or else the donor's liver, which must be cut in two before transplantation, will not be large enough to support the donor and candidate).

Graph generation is performed as follows. For each candidate and donor, we draw a gender (from the 2010 US Census Report[6]); conditioned on gender, we then draw candidate blood types

[5] In kidney exchange, tissue type (HLA antibodies and antigens) are an important determinant of compatibility. A candidate and donor sharing antigen encodings on the same locus are more likely to result in a rejected kidney. This is a drastically less important factor in liver transplantation, and is typically not taken into account in liver transplantation in practice or theory.

[6] http://www.census.gov/compendia/statab/cats/population.html

from the OPTN (Organ Procurement and Transplantation Network[7]) distribution and donor blood types from the overall US population.[8] We sample ages (dependent on gender) for candidates from the OPTN pool and for the donors from the 2010 US Census at a granularity level of one year. Then, given the age and gender (generated separately from OPTN data for candidate and US Census data for donors, as described earlier), we sample from a fine-grained table of weights released by the Center for Disease Control [155]. For candidates requiring a kidney, HLA is sampled from the OPTN databases. During edge generation, we include an exogenous "incompatibility factor" $f \in [0, 1]$ that randomly determines an edge failure even in the case of a compatibility success. This factor is common in the kidney literature [22], and is used to account for incompleteness of medical knowledge and temporal fluctuations in candidate-donor compatibility.

Appendix D.1 provides a much more in-depth detailing of the steps we take for data generation, as well as a formal compatibility graph generation algorithm. Next, we compare steady-state liver exchange graphs generated by our algorithm to kidney exchange graphs produced by the standard steady-state generator due to Saidman et al. [191]. Our generator is a generalization of that one.

9.3.2 Comparison to steady-state kidney exchange

In most empirical kidney exchange research, traits of the family of compatibility graphs used in experiments—like the average in- and out-degree of vertices or number of long paths in the graph—have typically had a large effect on both the performance of clearing engines and the qualitative results obtained (see, e.g., [14, 63, 75, 78, 134], or the discussion in Chapter 2). With that in mind, we now compare our steady-state generator to the current state of the art kidney exchange generator [191], which was meant to mimic a kidney exchange running in the United States in steady state. While the generators and data are similar in spirit, the medical differences between kidney and liver compatibility create distinctly different compatibility graphs both at the small and large scale. We will discuss those differences below.

Figure 9.6 plots the average number of edges in the liver-only compatibility graphs, using the generator in this chapter, against the average number of edges in the kidney compatibility graphs generated by the state of the art, as the number of candidate-donor pairs increases. The kidney compatibility graphs are, for graph sizes above 64, denser than comparably-sized liver compatibility graphs. This is interesting because it shows that, even though the liver exchange graphs do not need to take %PRA (i.e., HLA incompatibility) into account, their sensitivity to age and weight distributions proves to be more constricting than HLA sensitivity! Regardless, neither the liver nor the kidney graphs are sparse in the classical sense of the word: at $|V| = 1024$, the number of edges in the liver graph is 26% of the total possible edges in a 1024-clique. This lack of sparsity drives the experimental computational complexity of solving the real-world clearing problem (as exemplified by, e.g., the initial experimental results in Chapter 2, where the clearing problem with *unbounded* chains was easily solved on sparse real-world graphs, but the clearing problem with even *bounded* chains became computationally intractable).

Figure 9.7 enumerates the differences in the out-degree of the vertices in compatibility graphs

[7] http://optn.transplant.hrsa.gov/data/
[8] http://bloodcenter.stanford.edu/about_blood/blood_types.html

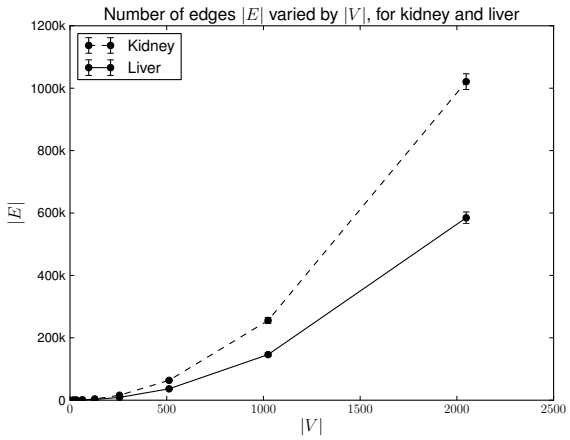


Figure 9.6: #Edges (in thousands) in generated liver and kidney compatibility graphs (100 graphs per $|V|$). The generated kidney graphs are denser than the liver graphs.

for liver-only exchange generated using our algorithm (shown in white) and compatibility graphs for kidney exchange from the Saidman et al. generator (shown in gray). The size of the graph, $|V|$, is held constant along the rows, while the exogenous incompatibility rate (f) between two otherwise compatible candidates and donors is held constant in each column. We vary $|V|$ and $f \in \{0.0, 0.2, 0.4, 0.8, 0.9\}$. Note that there is no notion of an exogenous incompatibility rate in the kidney graphs (although the %PRA virtual crossmatch simulation is similar to an exogenous incompatibility rate, but not parameterized); as such, the kidney exchange graphs vary only in terms of cardinality.

The cumulative distribution functions over the out-degrees of vertices, shown in Figure 9.7, exhibit interesting behavior. For example, there are more vertices with low degree in the liver exchange graphs than in the kidney exchange graphs. More interesting is the behavior exhibited by the kidney exchange graphs as $|V|$ increases. For instance, when $|V| = 1024$, we see three distinct out-degree sections in the kidney exchange graphs. These are an artifact of the somewhat ad-hoc method of doing %PRA virtual crossmatch tests in the Saidman et al. generator. The generator groups pairs into three sensitivity levels ("high", "medium", and "low"). As $|V|$ increases, those patients who are highly sensitized tend toward very few edges, while those at the medium and low sensitivity levels tend toward a medium and high number of edges, respectively. This is an artifact of the generator by Saidman et al. [191] and is not representative of the real kidney exchange data. Our generator (even if used for kidneys) does not have such coarse artifacting because it can bucket sensitivity into finer-grained classes.

9.3.3 Sparse generated compatibility graphs

While the transplantation of each organ is unique in its own way, we can draw from the experience of nascent kidney exchanges in the US and abroad when considering the makeup of a hypothetical liver or multi-organ exchange. As discussed in Section 9.2.2, early theoretical models of kidney

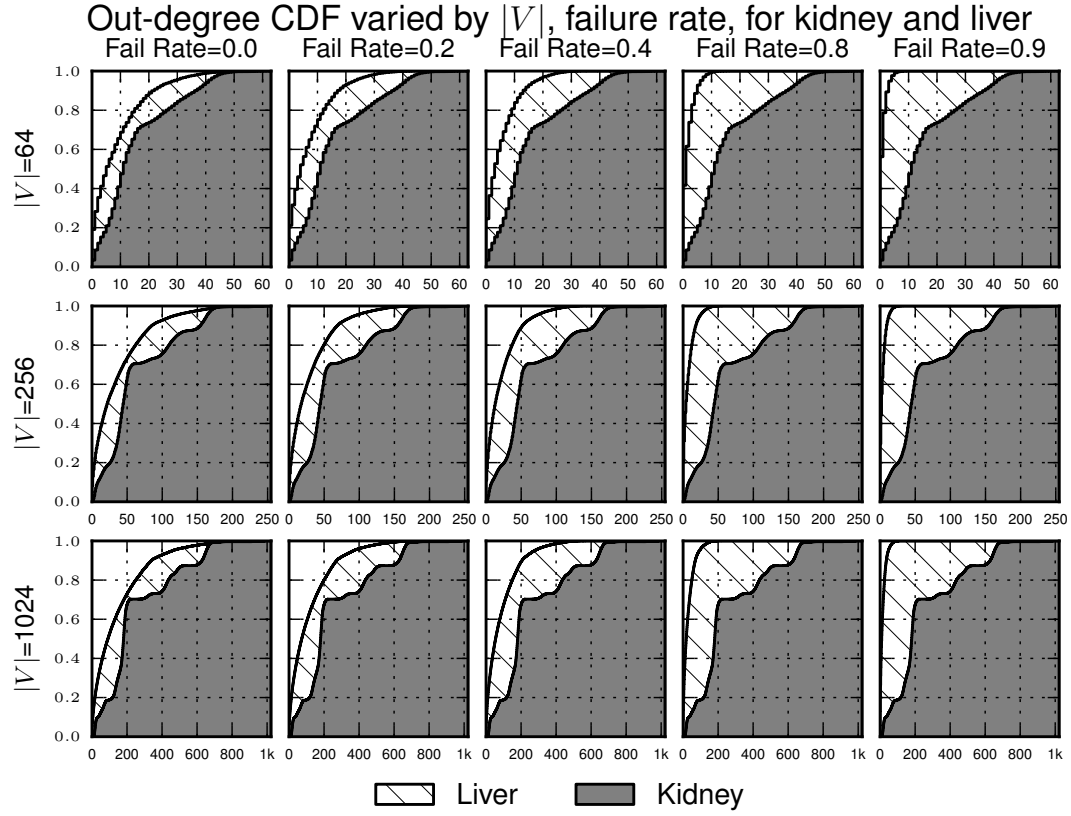


Figure 9.7: Cumulative distribution functions of the out-degree of vertices as we increase $|V|$ (varies per row) and exogenous incompatibility rate f (varies by column), shown for the liver graphs (in white) and kidney graphs (in gray). Note the divergence between kidney and liver graph as the exogenous incompatibility rate increases, as well as the three qualitative sections in the kidney graphs due to the three different %PRA classes.

exchange ended up behaving substantially differently than fielded exchanges; indeed, as practioners uncovered logistical constraints and medical features of kidney exchange, economic and computational models adapted to better reflect reality.

The data generation process described in Section 9.3.1 produces a "best guess" at what a steady-state liver or multi-organ exchange would look like. That generation process can easily be adapted to unforeseen features of those exchanges as they arise. Indeed, we address the unforeseen in a general way using the exogeneous incompatibility rate $f \in [0, 1]$. This incompatibility rate affects each pair independently. In reality, some pairs may be much harder to match than others, sometimes for poorly-understood reasons from a medical point of view. This is the case in kidney exchange (see discussions in, e.g., [23, 24, 75]).

With this in mind, in this chapter we also perform experiments directly on compatibility graphs drawn from the United Network for Organ Sharing (UNOS) nationwide kidney exchange, which is a large, fielded real-world kidney exchange that currently includes 143 transplant centers in the US. In other kidney-exchange-specific work, the authors built a compatibility graph generator that accurately mimics the UNOS nationwide exchange [77]. In the present work, we seed this generator with the first 192 match runs (October 2010 through March 2015) of the UNOS exchange and feed those graphs into our static and dynamic organ exchange simulators. To simulate multiple organs, we mark pairs as needing either a kidney or liver using demographic information from the most recent OPTN reports on waiting lists for kidneys and livers, respectively, and attach edges from real-world altruists in the UNOS pool only to those pairs marked as needing kidneys.

Obviously, we would not expect compatibility graphs generated in this manner to adhere at a fine-grained level to those of a fielded multi-organ exchange; indeed, these generated graphs contain no notion of different organs (beside kidneys) beyond a simple coin flip. That said, using these compatibility graphs allows us to remove the dependence on the exogeneous incompatibility factor f; indeed, it has already been taken into account by the real world in these graphs! Thus, by including experimental results on this second distribution of graphs, we hope to show *qualitatively* if not quantitatively that gains from multi-organ exchange still hold under a more intricate notion of exogeneous incompatibility rates—because certainly a fielded multi-organ exchange will encounter unpredictable constraints at run time, as was the case in the nascency of kidney exchange.

In the experimental results of Section 9.4, we will refer to those dense compatibility graphs (sparsified in a parameterized way using f and $p_{K \to L}$) generated in accordance with Section 9.3.1 as DENSE, and those exogeneously sparse compatibility graphs (sparsified further only by $p_{K \to L}$) drawn from real data as described in this section as UNOS.

9.4 Experimental results

We now provide computational results for a hypothetical nationwide liver or multi-organ exchange, using the realistic data generated above. First, we describe timing and matching results in the *static* case, where the algorithm sees the problem in its entirety up front. Second, we describe results for the *dynamic* case, where candidate-donor pairs arrive in the pool over time and are either matched or expire while waiting. We show results at sizes mirroring an estimated steady-state size of a US-wide liver exchange. Finally, we explore the possibility of a multi-organ exchange, where both

liver- and kidney-needing candidates can swap donors in the same pool. This results in more lives being saved than would be by running two separate nationwide liver and kidney exchanges.

9.4.1 Static liver exchange experiments

In the static case, the generator outputs a single graph and the optimization engine solves the clearing problem on this graph exactly once. Figure 9.8 shows timing results on liver exchange graphs of various sizes $|V|$ and exogenous incompatibility rates f drawn from the DENSE distribution. Intuitively, when f is low (or zero), the optimizer must consider many more edges than when f is high, resulting in longer run times for denser graphs. As expected, the computation time increases drastically with graph size—although our solver is still able to solve large problems to optimality.

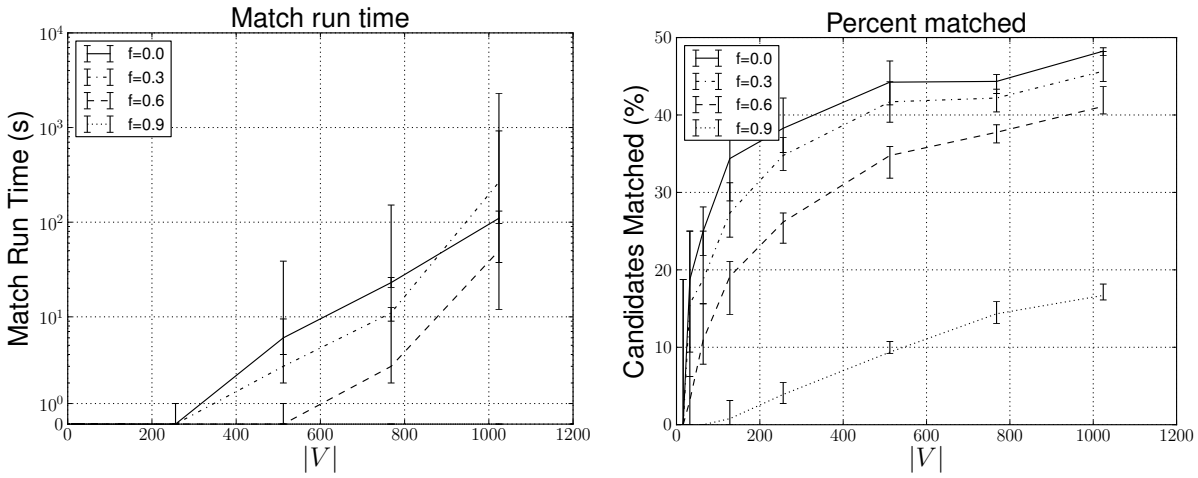


Figure 9.8: Median match run time (left) and median percentage of candidates matched (right), varying incompatibility rate f and graph size $|V|$, with first and third quartile error bars, for DENSE compatibility graphs.

Figure 9.8 also shows the percentage of candidates matched (the number of candidates matched by the algorithm divided by the total number of candidates in the pool) as a function of compatibility graph size $|V|$ and exogenous incompatibility rate f. Intuitively, when f is held low, the percentage of candidates matched is higher than when the incompatibility rate is high. Of interest is the match behavior as $|V|$ increases. Regardless of f, the percentage of candidates matched increases with the size of the underlying compatibility graph. This behavior is similar to that seen in kidney exchange and motivates the need for a large (nationwide or international) liver exchange.

Addressing the needs of society

The estimated steady-state monthly size of the nationwide kidney exchange is 10,000 candidate-donor pairs [6]. The rate of live liver donation is 1/8th of the rate of live kidney donation (5% of all liver transplants in the US involve live donors, compared to 40% for kidneys [50]), although this number would hopefully increase due to the publicity of a successful exchange—we will

conservatively estimate a factor of 1/2 as many live liver donors as kidney donors in steady-state. With 101,257 candidates currently waiting for a kidney and 15,268 candidates waiting for a liver in the US—and half as many live donors available—the steady-state for a US-wide liver exchange can be estimated at approximately half of 15,268 / 101,257 $\approx$ 7.5% of 10,000, or roughly 750 candidates. So, our clearing algorithm should be able to handle batch runs of a nationwide liver exchange.

9.4.2 Dynamic liver exchange experiments

In the dynamic case, a variable number of candidates enter and leave the pool over a period of multiple time units. While the fielded UNOS nationwide kidney exchange and others currently operate under the static paradigm described earlier, recent work in the kidney exchange community—some of which is presented in Chapters 6 and 8—has shown that optimizing in the dynamic setting leads to both more realistic and higher cardinality matchings over time [9, 15, 17, 24, 27, 70, 73, 213]. Regardless of the optimization method used, organ exchange is inherently dynamic, with candidates and donors arriving and departing over time; we work in such a setting here.

We start with a pool of $|V| = 400$ pairs assumed to contain highly-sensitized patients who built up in the system over time. These are matched myopically at each time period, including the final time period. Given a matched cycle by the algorithm, we then simulate that transplant actually succeeding in real life via an exogenous parameter set to 0.7 (this post-match, pre-transplant failure probability is drawn from real data, as motivated in Chapter 5). If any edge in a cycle fails, that entire cycle fails, and all candidates are returned to the pool (with the failed edge or edges removed). We simulate candidates leaving the pool (either through finding a transplant or dying). While 12% of patients in need of a kidney will be alive after 10 years via dialysis while waiting for a kidney [107], no such treatment exists for livers; thus, life expectancy drops to 1–2 years, which we simulate. In expectation $|V_{new}| = 233$ new candidates arrive in the pool per month, and the algorithm continues. We test over 24 months.

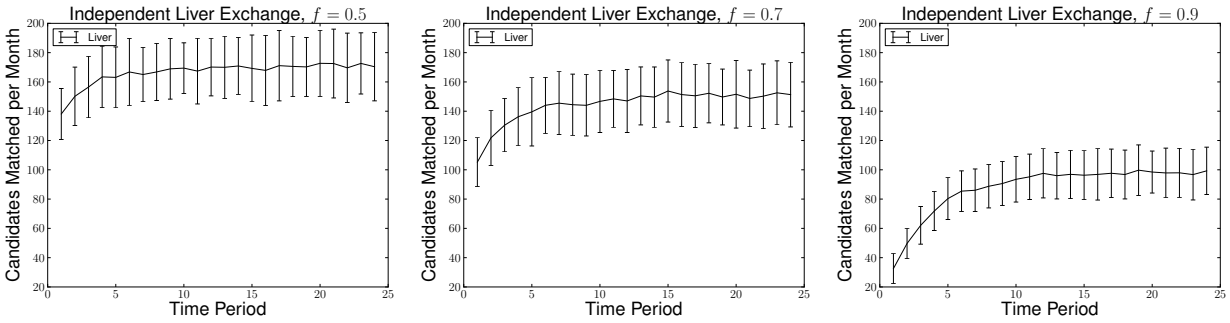


Figure 9.9: Number of candidates matched per time period in a dynamic setting over $T = 24$ months, for exogeneous incompatibility rates $f \in \{0.5, 0.7, 0.9\}$, for compatibility graphs drawn from the DENSE distribution.

Figure 9.9 shows the number of candidates matched by the clearing engine at each time period for dynamic graphs drawn from the DENSE distribution with exogeneous incompatibility rate

$f \in \{0.5, 0.7, 0.9\}$. Shown in the figure is the number of candidates matched *by the algorithm*, but before the virtual post-match failures are taken into account. Initially, there is a period of a few months during which the dynamic pool builds density as more candidate-donor pairs enter, followed by a relatively constant steady state. Intuitively, those simulated graphs with lower incompatibility rates f result in a larger number of matches per time period and overall, which qualitatively aligns with the static results of Section 9.4.1. Clearing times ranged between two and three minutes per time period.

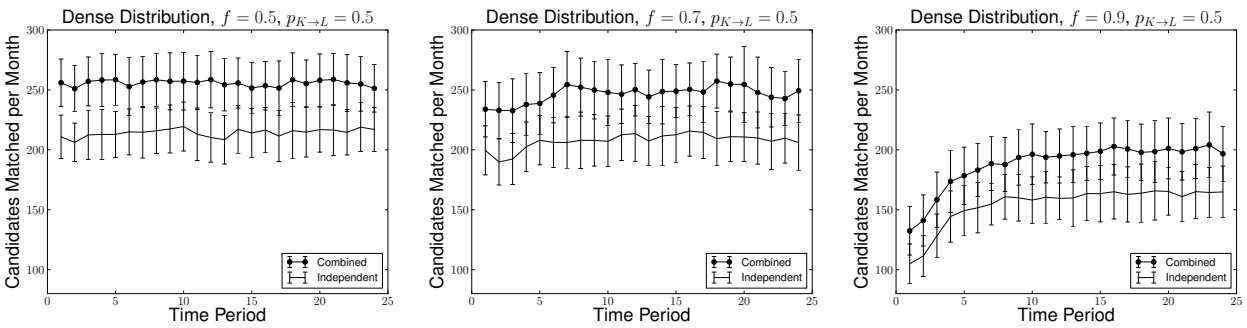
9.4.3 Dynamic bi-organ exchange experiments

In this section, we expand beyond simulating a dynamic liver exchange to the novel concept of multi-organ exchange. In the long run, one could imagine exchanges of multiple different kinds of organs. However, to our knowledge, only kidneys and livers have ever been swapped (and only separately). Ongoing work by Ergin et al. [90] is attempting to exchange *lungs*, another related but different organ exchange problem (e.g., typically two donors are required per candidate); it is likely that the first such hand-organized exchange will take place in Japan. Therefore, in this section we will focus on kidneys and livers. We show that combining an independent nationwide liver exchange with a nationwide kidney exchange into a joint kidney-liver exchange results in a statistically significant increase in the number of organ transplants, which aligns with Propositions 7, 8, and 9.

We simulate a demographically accurate bi-organ exchange featuring candidates in need of either a kidney or a liver who can swap donors in a *combined* candidate-donor pool. Approximately 85% of the candidates in the simulated pool need kidneys, while the other 15% need livers, as determined by OPTN waitlist data. We mimic the experiments in the previous section, with a starting pool size of $|V| = 400$ candidates who are highly sensitized and are assumed to have built up in the pool over time; we also include 100 altruistic kidney donors who enter the combined pool at an expected constant rate. We use the same post-match failure rate (0.7) as in the previous section, and simulate candidate-donor pairs entering and exiting the pool in a similar fashion. For DENSE experiments, to generate the candidates, we draw from the two different US distributions based on whether the candidate needs a kidney or a liver. Naturally, donors are drawn from the same US distribution in the two cases. For UNOS experiments, we draw from the UNOS generator as described in Section 9.3.3. We test over 24 months.

Figure 9.10 shows the number of candidates matched each month in the combined bi-organ exchange, as well as the aggregate number of candidates matched while keeping both liver- and kidney-needing candidates in separate pools. We set $p_{K \to L} = 0.5$, but relax this assumption later. Clearly evident is the loss of life resulting from keeping both the liver and kidney pools independent, with the bi-organ exchange matching roughly 40 more candidates per month, depending on exogeneous incompatibility rate f, when compared to the two independent exchanges.

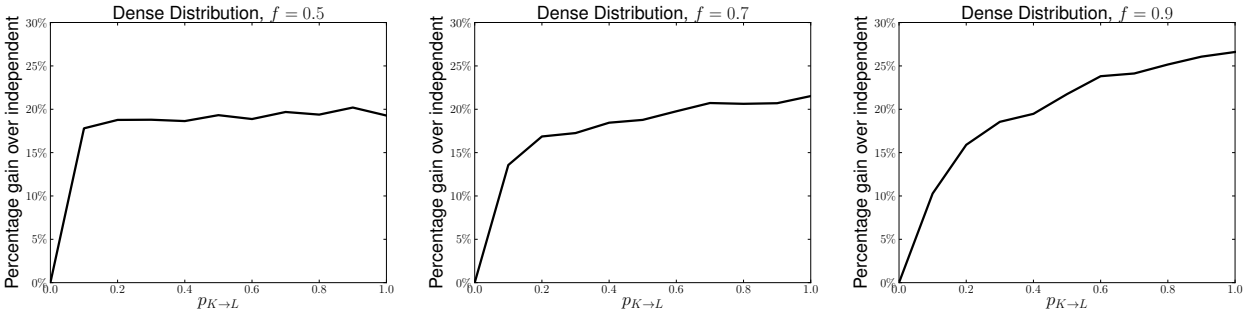
When we compare the *total* number of matches made over the entire period simulated above, the difference in lives saved between two independent pools and the combined bi-organ pool is more stark. In the experiments of Figure 9.10 with $p_{K \to L} = 0.5$, the combined bi-organ pool produced roughly 20% more matches than the sum of the two independent organ pools—specifically, 19.3%, 18.8%, and 21.8% for each of $f = 0.5$, $f = 0.7$, and $f = 0.9$, respectively. Independent



(a) $f = 0.5$ **(b)** $f = 0.7$ **(c)** $f = 0.9$

Figure 9.10: Number of matches per time period in independent liver and kidney exchanges and a combined multi-organ exchange in a dynamic setting over $T = 24$ months, for graphs drawn from the DENSE distribution with $f \in \{0.5, 0.7, 0.9\}$ and $p_{K\to L} = 0.5$

samples t-tests revealed that the difference between the aggregate number of lives saved using independent, simultaneous liver and kidney exchanges and using a combined multi-organ exchange was significant ($t(73) = 44.141$, $t(42) = 38.872$, and $t(81) = 41.651$ for each of $f = 0.5$, $f = 0.7$, and $f = 0.9$, respectively, with two-tailed $p \ll 0.0001$ for each). Qualitatively, this behavior is repeated for other values of $p_{K\to L} \in (0, 1.0]$, which we explore next.



(a) $f = 0.5$ **(b)** $f = 0.7$ **(c)** $f = 0.9$

Figure 9.11: Percentage gain in number of matches over two independent exchanges for a combined exchange as $p_{K\to L}$ increases, for generated graphs from the DENSE distribution and $f \in \{0.5, 0.7, 0.9\}$.

Figure 9.11 shows the percentage gain in total number of matches of a combined exchange over two independent exchanges as $p_{K\to L}$, the probability of a kidney-paired donor being willing to donate a liver, increases. As expected, higher rates of $p_{K\to L}$ result in better aggregate matching performance due to chains triggered by kidney-yielding altruists having more options to thread into the liver pool, and as longer (i.e., more valuable) chains. The effect of the exogeneous incompatibility rate f on the immediacy of this increase in match performance is noticeable; the relatively dense $f = 0.5$ DENSE graphs appear to maximize the use of chains for very low values

of $p_{K\to L} = 0.1$, while the sparser compatibility graphs see an increase across the entire range of values for $p_{K\to L} > 0$.

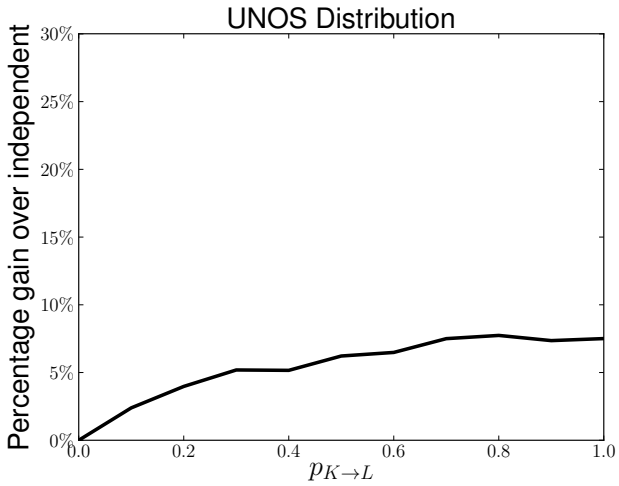


Figure 9.12: Percentage gain in number of matches over two independent exchanges for a combined exchange as $p_{K\to L}$ increases, for generated graphs from the UNOS distribution.

Finally, Figure 9.12 shows similar results on dynamic graphs taken from the UNOS distribution of graphs. While the absolute efficiency gains are less than in the DENSE experiments, gains of up to 7.6% are witnessed. This lower gain may be due in part to other intricacies specific to the UNOS generator, which mimics some of the operational constraints of the fielded UNOS exchange. For example, pairs have the ability to specify a maximum cycle or chain length in which they would be willing to participate; indeed, some pairs express a preference *not* to participate in chains at all, which we honor in these simulations. If pairs in the liver pool choose not to participate in chains, then the gains seen from combining pools will be lower than those that would be seen if all pairs chose to participate in all potential matching structures. Donors also are associated with specific transplant centers and can specify maximum travel distance or a list of (un)acceptable transplant centers where a donation can take place; many donors in the UNOS pool take advantage of this expressiveness, which further constrains the set of possible matches in graphs drawn from the UNOS distribution.

Additional experimental results and tabulated statistical significance testing for the data in Figures 9.10, 9.11 and 9.12 are presented in Appendix D.2.

The curves in Figures 9.11 and 9.12 are in part a function of the number of altruistic donors available in the pool. A greater number of altruists would result in lower necessary values of $p_{K\to L}$. Participation rates of altruists in kidney exchange are still in flux, the value of $p_{K\to L}$ is not yet known, and the level and type of sparsity of a real-world multi-organ exchange cannot truly be determined until one has been fielded; that said, the results of Figure 9.11 and 9.12 conclusively support the gain in number of matches from combining exchanges across a large variety of values for these unknown parameters.

9.5 Conclusions & future research

We explored the possibility of extending large-scale organ exchange to include liver lobes, either in conjunction with or independently of presently fielded kidney exchange. On demographically accurate data, vetted kidney exchange clearing algorithms (with a small tweak) can also clear liver exchanges at a projected US nationwide size. We explored the prospect of multi-organ exchange, where candidates needing either a liver or kidney can swap willing donors in the same pool. We showed that such a combination matches linearly more candidates than maintaining two separate exchanges; this linear gain is a product of altruistic kidney donors creating chains that thread through the liver pool. This result is supported experimentally on demographically accurate multi-organ exchanges with high statistical significance.

The work in this chapter is intended as a first foray into automated liver and multi-organ exchange. As such, there is much room for future research (much of which is applicable to other organ exchange and even to barter exchanges beyond organs), and is motivated by experiences fielding the nationwide kidney exchange. One direction of future work is to take on the slow and politics-laden task of founding a liver exchange, or including livers in currently fielded kidney exchanges. Recent and ongoing work by Ergin et al. [90] is attempting to do this for a *lung exchange*, another related but different organ exchange problem (e.g., typically two donors are required per candidate), with the first trial likely to occur in Japan. Recent work by Luo and Tang [149] approaches lung exchange from a game-theoretic point of view. Another direction is to develop scalable computational methods for the dynamic problem in the context of other organs. Indeed, due to the shorter expected maximum waiting time for candidates in need of a liver, such dynamic models may result in greater relative gains in number of matches compared to static models.

Finally, this chapter (and most papers on kidney exchange) deals with optimizing algorithmic organ matches; in reality, most algorithmic matches in fielded kidney exchanges do not result in an actual transplant. We expect this would be the case in liver and multi-organ exchange as well, although the exact failure rates for liver and multi-organ exchanges would be different than the observed failure rates in currently fielded kidney exchanges due to the medical and logistical differences in the organs and the transplant processes. Making organ exchange failure-aware is a critical step toward improving yield; recent work by Glorie [99] is an initial foray into learning a better estimate of the probability of a transplant failure between a patient and a donor, but much is left to be done, especially with organs other than kidneys. Using failure-aware techniques like those in Chapter 5, combined with techniques like those in Chapters 7 and 8, could both improve overall yield while balancing some of the fairness issues that would arise when comparing the different risk profiles of donating kidneys, livers, and other organs.

Regardless, the urgent societal need for liver exchange is there today, and we hope to be able to address it through a dedicated or combined liver- or multi-organ exchange.

PART V:

Conclusions & Future Research

Where is this place?

– Jean-Luc Picard

Where none have gone before.

– Data

10 Conclusions

This thesis addressed the design, analysis, and real-world fielding of dynamic matching markets and barter exchanges. We presented new mathematical models for static and dynamic barter exchange that more accurately reflect reality, proved theoretical statements about the characteristics and behavior of these markets, and developed provably optimal market clearing algorithms for models of these markets that can be deployed in practice. We showed that taking a holistic approach to balancing efficiency and fairness can often practically circumvent negative theoretical results, and gave a unified framework—FUTUREMATCH—that automatically learns to strike this balance in a dynamic environment. We supported the theoretical claims made in this thesis with extensive experiments on data from the United Network for Organ Sharing (UNOS) Kidney Paired Donation Pilot Program, a large kidney exchange clearinghouse in the US with which we have been, and are still, actively involved.

In Part I, we focused on theoretical properties of, and optimal mathematical-programming-based methods for solving, the static, deterministic clearing problem. A clearinghouse that does not consider the future when matching in the present is said to solve the *static* clearing problem; similarly, a clearinghouse that does not consider uncertainty over the existence of market participants, nor uncertainty over the existence of potential trades between participants, is said to solve the *deterministic* clearing problem. Solving the static, deterministic clearing problem is of both theoretical interest and practical importance, as it most accurately mimics the current reality of most fielded kidney exchanges' matching policies.

Chapter 2 introduced a dense random graph model of kidney exchange, proved a result about efficient matchings in that model, and then showed on real and simulated data that this result does not accurately mimic reality. This is a common story in matching market and kidney exchange research, and motivates the following chapters' focus on optimization methods for clearing exchanges with longer chains.

Chapter 3 addressed methods for solving the static, deterministic clearing problem to optimality. It first identified a bug in the previously leading branch-and-price-based solvers and proved a hardness result showing that those papers' approaches are likely not to work in general. To counter this negative result, it presented new models for kidney exchange that scale *substantially* better than the prior approaches when clearing exchanges with long, finite chains—including the now provably non-optimal prior leading solvers!

Chapter 4 presented a novel model for kidney exchange in which, under a light assumption that is true in reality, the clearing problem is solvable in polynomial time. This model is quite general and can be easily augmented to accept new developments in the organ exchange world, like how to optimally add edges via costly immunosuppression. We showed on real data that this new model accurately mimics reality.

Next, in Part II, we relaxed both the "static" and "deterministic" aspects of the clearing problem addressed by Part I. While fielded clearinghouses tend to solve the static, deterministic clearing problem, there is building theoretical and experimental consensus that considering both short-term and long-term uncertainty explicitly in the optimization model can result in tremendous gains in efficiency.

Chapter 5 addressed short-term uncertainty by way of the *failure-aware* model of barter exchange, which explicitly considers edge failure probability in the clearing optimization problem. It showed that in theory and in practice this results in large gains over standard deterministic clearing. It presented a novel branch-and-price-based optimal clearing algorithm specifically for this probabilistic exchange clearing problem and showed that this new solver scales well on large simulated kidney exchange data, unlike prior clearing algorithms. It also showed experimentally that taking failed parts from an initial match and instantaneously *re-matching* them with other vertices still in the waiting pool can result in significant gains, and that using a very small number of rounds of either *non-adaptive* and *adaptive* pre-match edge testing can help post-failure matching performance.

Chapter 6 addressed long-term uncertainty found in *dynamic* barter exchange, where vertices and edges arrive and depart over time, and the clearinghouse's problem is to maximize some utility function in the long run, possibly at the expense of immediate utility. It introduced a natural, general policy parameterization approach called *potentials*, and gave techniques for operationalizing it via black box parameter tuning. It presented some theoretical results comparing expressive power of different types of potentials. Experimentally, this method scaled to orders of magnitude larger than previous state-of-the-art methods, and results in gains in the objective relative to myopic optimization.

Part III addressed balancing the economic efficiency gained by techniques from Parts I and II with other concerns, like equity or other human-created objectives, in a theoretically-sound and computationally-tractable fashion.

Chapter 7 adapted the price of fairness metric to kidney exchange, and showed—in theory—that the price of fairness under very strict "fair" rules should be far. However, on real data, the price of fairness varies dramatically based on assumptions made (like the underlying sparsity of the graph or the failure rates being correlated with already-marginalized participants). It concluded with an empirical exploration of the tradeoff between matching more hard-

to-match patients and the overall utility of a utilitarian matching, on data from the UNOS nationwide kidney exchange and simulated data from each of the standard kidney exchange distributions.

Chapter 8 presented FUTUREMATCH, a framework for learning to match in a general dynamic model. FUTUREMATCH takes as input a high-level objective decided on by experts, then automatically (i) learns based on data how to make this objective concrete and (ii) learns the "means" to accomplish this goal—a task, in our experience, that humans handle poorly. It instantiated FUTUREMATCH in the kidney exchange realm and validated it on UNOS exchange data. FUTUREMATCH results in higher values of the objective. Furthermore, even under economically inefficient objectives that enforce equity, like those presented in the preceding Chapter 7, it yields better solutions for the *efficient* objective than traditional myopic matching that uses the efficiency objective.

Finally, Part IV presented new directions in general organ exchange.

Chapter 9 presented a model for liver and for multi-organ exchange. It showed that the technology to clear such exchanges in the large is ready to be fielded now. It also showed, through both theoretical random graph models and realistic experimental results in a dynamic setting, that combining separate liver and kidney exchanges into a unified multi-organ exchange will result in efficiency gains over running two individual organ exchanges.

Throughout the thesis, we used theory to support new research directions in the practical clearing and fielding of new exchange ideas; we also used our learnings from interaction with large fielded matching markets like the UNOS nationwide kidney exchange to drive new theoretical progress. We addressed various dimensions of fielded barter exchanges independently, and then presented a framework that learns to balance these dimensions, keeping a group of human experts' preferences in the loop at a high level. We showed theoretically and experimentally that this balance can be struck, providing strong evidence for our initial thesis statement.

Thesis statement

Competing dimensions—equity, efficiency, and computational tractability—in dynamic matching markets and barter exchanges can be balanced holistically through computational optimization methods and informed by random graph models.

We shall have to evolve
problem-solvers galore —
since each problem they solve
creates ten problems more.

– Piet Hein

11

Future research directions & open questions

In this chapter, we present some future research directions and a list of open questions that arose from the research performed as part of this thesis, or from discussions with other researchers and practioners in the area.

11.1 Managing uncertainty

In practice, deployed matching markets and barter exchanges are subject to a variety of types of uncertainty, like uncertainty over the existence of potential trades (as addressed in Chapter 5) or the longer-term future composition of the market (as addressed in Chapter 6). We list some future research directions targeting the management of such short- and long-term uncertainty now.

- The failure-aware model introduced in Chapter 5 can be viewed as a form of stochastic optimization with limited recourse. An alternate method due to Anderson et al. [16] that appeared after that work was published looks at a two-stage model of kidney exchange, where some edges are selected in the first stage, are shown to exist or not exist, and the goal is to maximize the size of the matching in the second stage that is built from those edges that were shown to exist. An alternative approach to optimization under uncertainty is *robust optimization* [32, 34], which may be more scalable in some cases as well as less dependent on having exact knowledge of, for example, the probability of edges or vertices existing in the compatibility graph.

Research Direction 1: Fully-robust clearing of barter exchanges

Formulate a good robust optimization model for kidney exchange. What is the uncertainty set? How does it perform against stochastic optimization methods, in terms of scalability and sensitivity to uncertainty?

Related papers: Anderson et al. [16], Bertsimas and Sim [33], Bertsimas et al. [34], Dickerson et al. [75, 80], Glorie et al. [98], Manlove and O'Malley [152]

- We estimated edge failure rates roughly from data in Chapter 5; however, even the correlation between a simple metric like CPRA and edge failure rate is still poorly understood [24, 99]—and real edges have much more in-depth information associated with them than a single number. As we saw repeatedly in simulation results using different edge failure distributions, assumed failure rates can make a large difference in the recommended matching. Better sensitivity analysis results would be helpful, as would more accurate methods for either predicting edge failure rates from past data, or choosing a small set of edges to query in advance of the matching as we did in Chapter 5 (see also work by Blum et al. [43, 44], Ryzhov and Powell [190]).

Research Direction 2: Learning failure rates

Create a better predictor of edge failure and vertex failure (either domain specific or general).

Related papers: Ashlagi et al. [24], De Klerk et al. [68], Dickerson et al. [75], Fumo et al. [94], Glorie [99], Leishman et al. [144]

- The full dynamic optimization problem for kidney exchange and many barter exchange applications is quite difficult. In Chapters 6 and 8, we presented one approximate method for computing good dynamic matching policies; however, in both of those chapters, our learning method did not converge. One could potentially use adaptations of recent techniques from reinforcement learning [136, 161, 217] to attempt to learn a policy.

  Of both theoretical and likely practical interest would be further exploration of the dynamic kidney exchange problem in the formal Markov Decision Process (MDP) setting. Here, we can view the (infinitely-large) set of all possible compatibility graphs as the state space of the MDP, the set of all feasible matchings for a particular compatibility graph as the action space per state, the reward function as, e.g., the number of matched pairs in the deterministic model or the expected number of transplants in the failure-aware model, and the transition function as a function of both the matching chosen and the underlying exchange dynamics. The discount factor would be set in accordance with, in the case of kidney exchange, the cost to a patient's health of waiting and, perhaps, the cost of dialysis as they wait.

  We note that without abstraction this MDP has an infinite state space, and a set of feasible actions that is exponential in the size of each state, so is likely intractable to solve exactly;

yet, intuitively, the results from Chapter 6 hint that abstraction techniques could yield good empirical results. Can we make any strong theoretical statements about this? Is there a level of abstraction that is both fine-grained enough to yield good gains in efficiency but coarse enough that learning the policy is tractable?

Research Direction 3: Better dynamic matching policies

Create better methods for learning dynamic matching policies, with better theoretical guarantees.

Related papers: Akbarpour et al. [9], Anderson et al. [15], Ashlagi et al. [24], Dickerson and Sandholm [70]

- When humans are also in the decision making loop at a clearinghouse, the interpretability of a proposed matching policy is important. While complex dynamic matching policies may lead to greater long-term efficiency, such increased complexity may not be palatable to decision makers—humans often wish to understand how a system works rather than simply trusting a black box. This intuition, combined with our experience working with fielded kidney exchanges, motivated the potentials idea in Chapter 6; after learning potentials, it is easy to map, e.g., types of vertices or types of edges to a single weight, which can then be presented to and discussed by a committee of human decision makers. Striking a balance between the interpretability of a matching policy—or any other model used in the market, like that which is used for edge failure prediction or potential trade quality assessment—and its expressive power is of both theoretical and practical interest.

Research Direction 4: Better *interpretable* dynamic matching policies

Create better methods for learning dynamic matching policies that can be easily explained to humans.

Related papers: Dickerson and Sandholm [70]

11.2 Incentives and privacy

A major challenge in kidney exchange, and in market and mechanism design in general, is incentivizing participation—meaning either participation *at all*, or *truthful* participation. There are many classes of agents who participate in an exchange; in the kidney exchange case, these may include patients, donors (altruistic or paired), specific surgeons or chairs of surgery departments, transplant centers, and the clearinghouses themselves.

- In kidney exchange, transplant centers hide some of their donor-patient pairs and altruistic donors from the exchange and instead try to match them locally. This is a major problem in practice. For example, of the pairs revealed to the UNOS exchange from its beginning

in October 2010 to May 2012, *none* could have been locally matched in their transplant centers [200]. In other words, the centers did not reveal any of their pairs that could be locally matched to the exchange. There is no perfect mechanism design solution to that problem when viewed in a deterministic, static setting (see, e.g., Ashlagi and Roth [19], Ashlagi et al. [25], Sönmez and Ünver [198]); some ongoing research we have with Hajaj et al. [104] works to circumvent these impossibility results by moving into a different model. Similarly, recent work by Blum et al. [45] makes use of randomization to get good results in a static setting. Yet, many open questions remain, and theoretically-sound ideas in that space that are *also* able to be practically fielded have the potential to greatly impact today's kidney exchange landscape.

Research Direction 5: Increasing transplant center participation

Design mechanisms that both mimic reality and incentivize truthful (or otherwise "good") behavior by transplant centers.

Related papers: Ashlagi and Roth [19], Ashlagi et al. [25], Blum et al. [45], Carvalho et al. [55], Hajaj et al. [104], Toulis and Parkes [209]

- Potential donors can also be incentivized to join exchanges—or, at an even simpler level, be incentivized to register as deceased donors. In 2013, Chiquinho Scarpa, a wealthy Brazilian, announced on Facebook that he would bury one of his luxury vehicles, a Bentley Flying Spur, so that he might drive it in the afterlife.[1] After some initial public outcry, he released the following statement: "I have not buried my car, but everyone thought it absurd when I said I'd do it. It is absurd to bury their bodies, which can save many lives. Nothing is more valuable. Be a donor, tell your family." While we may not all have luxury vehicles to bury, nor the ability to make exotic statements like Mr. Scarpa, there are other effective methods to increase donor participation in organ transplantation mechanisms.

  Sites like Organize [169] use social network effects to publicize organ donation, and streamline the registration process, but are still in their nascency. A formal study of how social network effects influence organ donation signups in kidney exchange is of interest; one possible step might be to interact with Facebook, a social networking site that allows users to publicly state their (deceased) donor registration status, to see if trends like "cascading donor signups" exist. The incentives in deceased donation are different than in paired donation; formalizing that dichotomy is also of interest. Furthermore, and obviously, fielding new methods that increase donor signups would have great impact on the state of organ donation.

[1] http://www.irishtimes.com/business/media-and-marketing/grotesque-burying-of-bentley-proves-winner-at-ad-oscars-1.1842354

Research Direction 6: Donor participation

Increase donor participation in kidney exchange.

Related papers: Stoler et al. [201], Woodle et al. [219]

- Notions of stability have been central in matching market design; paying heed to the ability of two or more agents participating in a centralized clearinghouse to deviate from that recommended matching is thought to have contributed to the prevention of market unraveling in a variety of instances [167, 189]. There has been some work exploring notions of stability in kidney exchange, but as yet this has been in reduced models of the full problem. Myriad impossibility results abound with regard to achieving matchings that are both stable and "good" for all involved parties in traditional matching markets like school choice; however, recent work uses differential privacy to approximately circumvent some of these impossibility results in traditional matching markets [126]. What do these (im)possibility results look like in the context of barter exchange? Can we implement private barter exchange in practice, if not in theory [112, 125]?

Research Direction 7: Private exchange & notions of stability

Explore practical notions of privacy and stability in barter exchange.

Related papers: Hsu et al. [112], Kannan et al. [125, 126], Liu et al. [148]

11.3 Other open questions & research directions

We now discuss other open research questions and future research directions that are related to the topics covered in this thesis.

- Fairness in healthcare applications is of paramount importance. The work of Chapter 7 extended the general price of fairness to the kidney exchange setting; however, that work explicitly considered orderings over classes of agents, and its theoretical results were not particularly realistic. Hooker and Williams [110] give a general method for combining equity and economic efficiency in a single mathematical programming model; applying this to the barter exchange setting is of interest. One concrete first step would be to apply the work of Hooker and Williams [110] to model the equity tradeoffs in joint kidney-liver exchange. As discussed in Chapter 9, the risk of donating a liver is generally substantially higher than donating a kidney; we modeled this via a coin flip stating whether or not a kidney-paired donor would be willing to give a liver, but more general methods would be informative.

Research Direction 8: Fairness in organ exchange

Create more general models of fairness in organ exchange. Analyze them theoretically. Emphasize models that can be deployed in practice.

Related papers: Bertsimas et al. [37], Dickerson and Sandholm [72], Hooker and Williams [110], Li et al. [146]

- As the experiments of Chapter 3 showed, different solvers perform well (or not well) under different models of kidney exchange; for example, the solver due to Anderson et al. [16] is fastest with uncapped chains, but cannot solve even small instances for "long but capped" chains. Similarly, the solver due to Klimentova et al. [134] was fastest for many real UK instances *without* chains, which is not the current state of the art. Our PICEF model from Chapter 3 was fastest for the current state of the practice in kidney exchange—that is, short cycles and long but finite chains. However, there is currently no principled method for selecting the "best" solver for the job, given a set of business constraints.

  From a theoretical point of view, extending the comparison of LPRs to a complete ordering of all LPRs amongst models of kidney exchange—especially for different parameterizations of the underlying model, like the inclusion of chains or edge failures—would give insight as to which solver is best suited for an exchange running under a specific set of business constraints. From a practical point of view, using a portfolio-based approach to selecting the right solver for a class of instances would be of interest. This approach has routinely worked in the combinatorial optimization space (see, e.g., SATzilla for SAT [220], or `sunny-cp` for Constraint Programming [13]). Indeed, one could parameterize static solvers and then use similar black box tuning techniques to those we used in Chapter 6 to learn potentials.

Research Direction 9: When to use a specific solver

Create a complete ordering of linear program relaxations of various models under different constraints (e.g., cycle size, chain size, etc).

Related papers: Abraham et al. [6], Anderson et al. [16], Constantino et al. [63], Dickerson et al. [80], Mak-Hau [151]

- While the exchange of money for organs is viewed as repugnant in nearly all countries, money nonetheless plays a large role in organ transplantation worldwide. Much of the mathematical work on kidney exchange has ignored entirely the presence of money in kidney exchange. Effectively understanding the flow and power of money in kidney exchange could lead to new mechanism design problems where the transplant centers' utility functions take not just number of matches into account, but also potential profit or loss; this adjustment to the model is both more realistic and may allow us to circumvent some of the standard impossibility results in mechanism design in kidney exchange.

Research Direction 10: Financing barter exchanges

Create accurate models of the flow of money in realistic models of barter exchange.

Related papers: Akbarpour et al. [10], Fang et al. [93], Goldberg et al. [102]

- We showed in Chapter 9 that the technology exists to field general organ exchanges; yet, the process to move from academic paper to, e.g., fielded liver exchange would be arduous and likely different from that of fielding a kidney exchange (due, in part, to less demand and less supply of livers, higher morbidity and mortality, and fewer transplant centers being willing to perform liver transplants). That said, Chapter 9 showed that combining kidney and liver exchanges would result in great gains in number of transplants, so the idea is worth seriously considering.

Research Direction 11: Field new organ exchanges

Field a liver exchange. Field a multi-organ exchange.

Related papers: Dickerson and Sandholm [72], Ergin et al. [90, 91]

A

Additional theoretical and experimental results for position-indexed formulations

A.1 Additional proofs for the PIEF model

We now provide additional theoretical results pertaining to the position-indexed edge formulation (PIEF) model, and proofs to theoretical results stated in Section 3.3.1 of Chapter 3.

A.1.1 Validity of the PIEF model

Lemma 4. *Any assignment of values to the x^l_{ijk} that respects the PIEF constraints yields a vertex-disjoint set of cycles of length no greater than L.*

Proof. We show this by demonstrating that in each graph copy, the set of selected edges is either empty, or composes a single cycle of length no greater than L.

Let $l \in P$ be given such that at least one edge is selected in graph copy D^l, and let k_{max} be the highest position k such that $x^l_{ijk} = 1$ for some i, j. Any selected edge (i, j) at position k_{max} in graph copy D^l must point to l, as otherwise the flow conservation constraint (3.1c) would be violated at vertex j. Furthermore, there must be no more than one edge selected at position k_{max} in graph copy D^l, as otherwise the capacity constraint (3.1c) for vertex l would be violated.

The flow conservation constraints (3.1b) ensure that we can follow a path backwards from the selected edge at position k_{max} to a selected edge at position 1, and also that at most one edge is selected at each position. Since the edge at position 1 must start at vertex l by the construction of $\mathcal{K}(i, j, l)$, we have shown that graph copy D^l contains a selected cycle beginning and ending at l, and that this graph copy does not contain any other selected edges.

Constraint (3.1b) ensures that the vertex-disjointness condition is satisfied. □

Lemma 5. *For any vertex-disjoint set of cycles of length no greater than L, there is an assignment of values to the x^l_{ijk} respecting the PIEF constraints.*

Proof. This assignment can be constructed trivially. □

Theorem 17. *The PIEF model yields an optimal solution to the kidney exchange problem.*

A.1.2 Proofs for the LPR of PIEF

The following lemma is used in the proof of Theorem 4, and its proof is included here for completeness.

Lemma 6. *Let a sequence of edges $W = (a_1, \ldots, a_{|W|})$ in a directed graph D be given, such that W is a closed walk—that is, the target vertex of each edge is the source vertex of the following edge, and the sequence starts and ends at the same vertex. (It is permitted for an edge to appear more than once in W.) Let X be the multiset $\{a_1, \ldots, a_{|W|}\}$. Then we can partition X into $C = \{c_1, \ldots, c_{|C|}\}$, where each $c_i \in C$ is a set of edges that form a cycle in D.*

Proof. The following algorithm can be used to construct the set C.

1. Let $C = \{\}$.
2. For each $i \in \{1, \ldots, |W|\}$, let s_i be the source vertex of a_i.
3. If the sequence $(s_i)_{1 \leq i \leq |W|}$ contains no repeated vertices, then W must be a cycle; go to step 5.
4. Choose $i, j \in \{1, \ldots, |W|\}$ with $i < j$, such that $s_i = s_j$ and all the s_k are distinct for $i \leq k < j$. The edges $(a_k)_{i \leq k < j}$ form a cycle; remove these from W and add the set of removed edges to C. (Observe that W remains a closed walk). Re-index the edges in the new, shorter W as $1, \ldots, |W|$. Return to step 2.
5. Add $\{a : a \text{ appears in } W\}$ to C and terminate.

□

Theorem 4. *$Z_{CF} = Z_{PIEF}$ (without chains).*

Proof. $Z_{CF} \preceq Z_{PIEF}$. Let $(z^*_c)_{c \in \mathcal{C}(L,0)}$ be an optimal solution to the LPR of the cycle formulation, with objective value Z_{CF}. We will construct a solution to the LPR of PIEF whose objective value is also Z_{CF}. We translate the z^*_c into an assignment of values to the x^l_{ijk} in a natural way, as follows. For each $c \in \mathcal{C}(L, 0)$, let l be the index of the lowest-numbered vertex appearing in c. Number the positions of edges of c in order as $\{1, \ldots, |c|\}$, beginning with the edge leaving l.

For each vertex $l \in V$, each edge $(i, j) \in A^l$, and each position $k \in \mathcal{K}(i, j, l)$, let $\mathcal{C}(L, 0)(i, j, k, l)$ be the set of cycles in $\mathcal{C}(L, 0)$ whose lowest-numbered vertex is l and which contain (i, j) at position k. Let

$$x^l_{ijk} = \sum_{c \in \mathcal{C}(L,0)(i,j,k,l)} z^*_c.$$

This construction yields a solution which satisfies the PIEF constraints and has objective value Z_{CF}.

$Z_{PIEF} \preceq Z_{CF}$. Let an optimal solution (x^l_{ijk}) to the LPR of PIEF be given, with objective value Z_{PIEF}. Our strategy is to begin by assigning zero to each cycle formulation variable $z_c (c \in \mathcal{C}(L, 0))$, and to make a series of decreases in PIEF variables and corresponding increases in cycle formulation variables, ending when all of the PIEF variables are set to zero. We maintain three invariants after each such step. First, the sum of the PIEF and cycle formulation objective values remains Z_{PIEF}. Second, the constraints of the relaxed PIEF are satisfied. Third, the following vertex capacity constraint—which combines the capacity constraints from the PIEF and the cycle formulation—is satisfied for each vertex.

$$\sum_{l \in V} \sum_{i:(i,j) \in A^l} \sum_{k \in \mathcal{K}(i,j,l)} x^l_{ijk} + \sum_{c: j \in c} z_c \leq 1 \qquad j \in P$$

If any of the PIEF variables takes a non-zero value, then we can select i_1, i_2, l such that $i_1 = l$ and $x^l_{i_1 i_2 1} > 0$. By the PIEF flow conservation constraints (3.1c), we can select a closed walk $W = ((i_1, i_2), (i_2, i_3), \ldots, (i_{k'-1}, i_{k'}))$ of at most L edges in D^l such that $x^l_{i_k i_{k+1} k} > 0$ for $1 \leq k < k'$, and such that $i_1 = i_{k'}$. Let $x_{\min}$ be the smallest non-zero value taken by any of the $x^l_{i_k i_{k+1} k}$.

By Lemma 6, the set of edge positions $\{1, \ldots, k' - 1\}$ can be decomposed into a set of sets C, such that for each $c \in C$ we have that the edges $\{a_k : k \in c\}$ can be arranged to form a cycle; we denote this as $\text{cyc}(c)$. For each $c \in C$, we subtract $x_{\min}$ from $x^l_{i_k i_{k+1} k}$ for each $k \in c$, and we add $x_{\min}$ to $z_{\text{cyc}(c)}$. This transformation strictly decreases the count of x^l_{ijk} variables that take a non-zero value. By repeatedly carrying out this step, we will reach a point where all of the x^l_{ijk} variables take the value zero, and where the z_c variables respect the cycle formulation constraints and give objective value Z_{PIEF}. □

A.2 Additional proofs for the PICEF model

We now provide additional theoretical results pertaining to the position-indexed chain-edge formulation (PICEF) model, and proofs to theoretical results stated in Section 3.3.2 of Chapter 3.

A.2.1 Validity of the PICEF model

Lemma 7. *Any assignment of values to the y_{ijk} that respects constraints (3.3c) and (3.3d) and such that*

$$\sum_{j:(j,i) \in A} \sum_{k \in \mathcal{K}(j,i)} y_{jik} \leq 1 \tag{A.1}$$

for all $i \in P$, yields a vertex-disjoint set chains of length no greater than K.

Proof. We say that edge (i, j) is *selected* at position k if and only if $z_{ijk} = 1$.

Our proof has three parts. We first give a procedure to construct a set S of chains of length no greater than K, where each chain in S consists only of selected edges. We then show that these chains are vertex-disjoint, and that any selected edge appears in some chain in S.

By constraint (3.3c), each $i \in N$ has at most one selected outgoing edge. For each $i \in N$ that has an outgoing edge (i, j_1), we begin to construct chain c by letting $c = ((i, j_1))$—a sequence containing one edge. Vertex j_1 has at most one selected outgoing edge at position 2, by constraint (3.3d). If such an edge (j_1, j_2) exists, we add it to our chain. We continue to add edges $(j_2, j_3), (j_3, j_4), \ldots$, until we reach k such that a selected edge from j_k at position $k + 1$ does not exist. The chain c will therefore be a path of selected edges at positions $1, \ldots, |c|$, where the length of c can be no greater than K since no variable in the model has position greater than K. Add the chain c to S.

By constraint (3.3c), no vertex in N can appear in two chains in S. By constraint (3.3b), the same is true for vertices in P. Hence, the chains in S are vertex disjoint.

To complete the proof, we show that any selected edge must be part of one of these chains in S. Let a variable z_{ijk} taking the value 1 be given. By applying constraint (3.3d) repeatedly, we can see that there must exist be a path of length k from an NDD h to j, containing only selected edges. Let $c \in S$ be the chain starting at h. Since no vertex in c has two selected outgoing edges, there must must exist a unique path of length k from h, and (i, j) must therefore be the kth edge of c. □

Lemma 8. *Any assignment of values to the y_{ijk} and z_c that respects the PICEF constraints yields a vertex-disjoint set cycles of length no greater than L and chains no greater than K.*

Proof. We call a cycle c such that $z_c = 1$ a *selected cycle*, and an edge (i, j) such that $z_{ijk} = 1$ for some k a *selected edge*.

By (3.3b), the selected cycles are vertex disjoint. By Lemma 7 the selected edges compose a set of vertex-disjoint chains, each of which has length bounded by L (The conditions of the lemma are satisfied since constraint (3.3b) implies (A.1)).

It remains to show that no selected cycle shares a vertex with a selected edge. Suppose, to the contrary, that some selected cycle c shares vertex $i \in P$ with a selected edge a. Vertex i cannot be the target of a, since constraint (3.3b) would be violated if i appears both in selected cycle c and as the target of selected edge a. Hence $a = (i, j)$ for some $j \in P$. By constraint (3.3d), i must be the target of another selected edge, a'. Therefore, i appears in c and is the target of a', violating constraint (3.3b). □

Lemma 9. *For any valid set of vertex-disjoint cycles and chains, there is an assignment of values to the y_{ijk} and z_c respecting the PICEF constraints.*

Proof. This assignment can be constructed trivially. □

Theorem 18. *The PICEF model yields an optimal solution to the kidney exchange problem.*

A.2.2 Proofs for the LPR of PICEF

Theorem 5. *$Z_{CF} \prec Z_{PICEF}$ (with chains).*

Proof. $Z_{CF} \preceq Z_{PICEF}$. Consider an optimal solution to the LPR of the cycle formulation. We show how to construct an equivalent (optimal) solution to the LPR of PICEF. For $c \in \mathcal{C}(L, 0)$, we transfer the value of z_c directly from the cycle formulation solution to the PICEF solution. For each $(i, j) \in A$ and each $k \in \mathcal{K}'(i, j)$, let

$$y_{ijk} = \sum_{(i,j) \text{ appears at position } k \text{ of } c} z_c.$$

This solution has the same objective value as the cycle formulation solution, and satisfies the constraints of the LPR of PICEF.

$Z_{CF} \prec Z_{PICEF}$. Figure A.1 shows a graph for which Z_{PICEF} is strictly greater (i.e., worse) than Z_{CF}. Let $L = 2$ and $K = 4$. In this instance, $N = \{1\}$ and $P = \{2, \ldots, 7\}$.

In the cycle formulation, this instance has no admissible cycles, and the only admissible chains are $1 \to 2 \to 3 \to 4$, $1 \to 5 \to 6 \to 7$, and their prefixes. Since the longest chain has length 3 and any the sum of chain-variables containing vertex 1 may not exceed 1, we can see that the optimal objective value to the LPR of the cycle formulation is 3.

We can achieve an objective value of $7/2$ to the LPR of PICEF, by letting $y_{121} = y_{232} = y_{343} = y_{151} = y_{562} = y_{673} = y_{754} = 1/2$. □

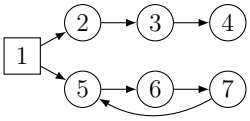


Figure A.1: A graph where Z_{PICEF} is strictly greater than Z_{CF}.

Theorem 6. *Let $z \in \mathbb{R}^+$ be given. There exists a problem instance for which $Z_{PICEF}/Z_{CF} > z$, where Z_{PICEF} is the objective value of the LPR of PICEF and Z_{CF} is the objective value of the LPR of the cycle formulation.*

Proof. We give a family of graphs, parameterized by L and K, for which Z_{PICEF} is strictly greater than Z_{CF}. Given a constant cycle cap of L and a chain cap of K (which can effectively be infinite, if $K = |V|$), the graphs are constructed as follows. For $i \in [K - L - 1]$, create a cycle $\langle v_1^i, v_2^i, \ldots, v_{L+1}^i \rangle$ such that $v_1^{i+1} = v_2^i$ for each $i \in [1, K - L - 2]$; the cycle is otherwise disjoint from the rest of the graph. Connect a single altruist a to v_1^1; the altruist is otherwise disjoint from the rest of the graph. Figure A.2 visualizes the constructed graph.

The maximum cardinality disjoint packing of cycles of length at most L and chains of length at most K is the unique chain $(a, v_1^1, v_1^2, \ldots, v_1^{K-L-1}, v_2^{K-L-1}, \ldots, v_{L+1}^{K-L-1})$. Thus, $OPT = K$, where OPT is the optimal objective value to the integer program. Indeed, there are no legal cycles of length at most L in the graph, and at most one chain can be in any feasible solution due to the shared altruist a, so the (unique, by construction) longest chain is optimal.

The LPR of the PICEF representation of this instance will assign weight of $1/2$ to each edge in the graph, for a total objective of $Z_{PICEF} = 1/2 + (K + L - 1)(\frac{L+1}{2})$.

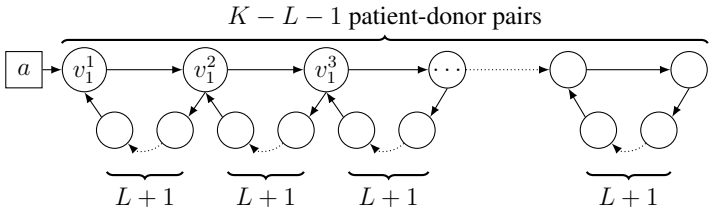


Figure A.2: Family of graphs where Z_{PICEF} is strictly looser than Z_{CF}.

The LPR of the cycle formulation representation will create variables for each feasible cycle and chain in the graph. There are no feasible cycles in the graph. All chains in the graph share the edge (a, v_1^1); thus all chains intersect and all chains contain a. Thus, the sole binding constraint in the LPR of the cycle formulation is that the altruist node a appears in at most one chain. For chain decision variables $x_c \in [0, 1]$, this problem can be rewritten as

$$\max \sum_c |c| x_c \text{ subject to } \sum_c x_c \leq 1$$

This constraint matrix is totally unimodular, and thus the LP optimum is integral (and is the IP optimum, or $Z_{CF} = OPT = K$).

The ratio of Z_{PICEF} to Z_{CF} is thus

$$\frac{1}{2K} + \left(1 + \frac{L-1}{K}\right)\left(\frac{L+1}{2}\right)$$

which can be made arbitrarily large by increasing L. □

A.3 Additional proofs for the HPIEF model

A.3.1 Validity of the HPIEF model

Lemma 10. *Any assignment of values to the x_{ijk}^l and y_{ijk} that respects the HPIEF constraints yields a vertex-disjoint set cycles of length no greater than L and chains no greater than K.*

Proof. (Sketch.) Clearly, if the HPIEF constraints are satisfied then the PIEF constraints (3.1b-3.1d) are satisfied also. Therefore, by Theorem 4, the edges selected by the x_{ijk}^l form a vertex-disjoint set of cycles of length no greater than L.

By Lemma 7, the selected edges compose a set of vertex-disjoint chains, each of which has length bounded by K.

It remains to show that the selected cycles and chains are vertex-disjoint. This can be showed straightforwardly, along similar lines to the proof for Theorem 8. □

Lemma 11. *For any valid set of vertex-disjoint cycles and chains, there is an assignment of values to the x_{ijk}^l and y_{ijk} respecting the HPIEF constraints.*

Proof. This assignment can be constructed trivially. □

Theorem 19. *The HPIEF model yields an optimal solution to the kidney exchange problem.*

A.3.2 LPR comparison of HPIEF and PICEF

Theorem 20. $Z_{HPIEF} = Z_{PICEF}$

The proof is similar to the proof for Theorem 4, and is therefore omitted.

A.4 Additional background and proofs for the failure-aware PICEF model

In this section, we provide a proof of correctness of Algorithm 2—which implements polynomial-time pricing of cycles for branch and price in the augmented failure-aware PICEF model—and discuss by way of counterexample why the basic deterministic polynomial-time cycle pricing algorithms of Glorie et al. [100] and Plaut et al. [174] cannot be directly used for this case.

A.4.1 Proof of Theorem 7

Theorem 7. *If there is a discounted positive price cycle in the graph, Algorithm 2 will return at least one discounted positive price cycle.*

Proof. Let $c = \langle v_1, v_2, \ldots, v_n \rangle$ be a discounted positive price cycle. Then $p^n \sum_{(i,j)\in c} w_{ij} - \sum_{j\in c} \delta_j > 0$. Therefore $\sum_{j\in c} \delta_j - p^n \sum_{(i,j)\in c} w_{ij} < 0$. Then by definition of w_k, we have $\sum_{(i,j)\in c}(\delta_j - p^n w_{ij}) = \sum_{(i,j)\in c} w_n(i,j) < 0$.

This implies that c is a negative cycle in D on the $k = n$ iteration of Algorithm 2. By the correctness of GETNEGATIVECYCLES, if there is a negative cycle in the graph, GETNEGATIVECYCLES(D, n, w_n) will return at least one negative cycle of length at most n.

Let c' be a returned cycle. Since c' is negative in D on the $k = n$ iteration, we have $\sum_{j\in c'} \delta_j - p^n \sum_{(i,j)\in c'} w_{ij} < 0$. Therefore $p^n \sum_{(i,j)\in c'} w_{ij} - \sum_{j\in c'} \delta_j > 0$.

Since $|c'| \leq n$ by the correctness of GETNEGATIVECYCLES, we have $p^{|c'|} \geq p^n$. Because all edge weights in the original graph are nonnegative, $\sum_{(i,j)\in c'} w_{ij} \geq 0$. Therefore $p^{|c'|} \sum_{(i,j)\in c'} w_{ij} \geq p^n \sum_{(i,j)\in c'} w_{ij}$. Then $p^{|c'|} \sum_{(i,j)\in c'} w_{ij} - \sum_{j\in c'} \delta_j \geq p^n \sum_{(i,j)\in c'} w_{ij} - \sum_{j\in c'} \delta_j > 0$, so c' is indeed discounted positive price.

Therefore Algorithm 2 returns at least one discounted positive price cycle. □

A.4.2 Insufficiency of previous algorithms for the failure-aware pricing problem

The pricing problem in the deterministic context, where post-match failures are not considered, is known to be solvable in polynomial time for cycles [100, 174] but not chains [175]. In Section 3.3.5, we presented Algorithm 2, a polynomial-time algorithm for the pricing problem for

cycles in the failure-aware context, for uniform success probability. In this section, we show how the basic algorithm for the deterministic pricing problem is not sufficient for the failure-aware context.

The algorithm for the deterministic setting initiates a Bellman-Ford style search to find negative cycles. Bellman-Ford is run P times: on each iteration a different vertex representing a donor-patient pair is the source. After Bellman-Ford has been run from the source s for $L-1$ steps, suppose there is a path ρ from s to some vertex v with weight $w(\rho)$, and there is an edge from v back to s with weight w_{vs}. If $w(\rho)+w_{vs}<0$, then $\rho \cup (v,s)$ is a negative cycle [100, 174].

For consistency, in this section we discuss finding discounted *negative* price cycles, which is trivially equivalent to finding discounted positive price cycles by reversing the signs on all edge weights and dual values. Therefore, we are looking for cycles c satisfying $\sum_{j\in c}\delta_j - p^n \sum_{(i,j)\in c} w_{ij} < 0$.

Consider a straightforward modification to the algorithm from the deterministic setting, where each path now separately remembers its accumulated sum of dual values, sum of edge weights, and length. All of these can be easily recorded during the Bellman-Ford update step without altering the algorithm's complexity.

The issue arises when comparing paths. Figure A.3 gives an example of this. Consider running Bellman-Ford with s as the source and $L=3$. The path (s,v_2,v_3) is preferable to (s,v_1,v_3), since we will end with the 3-cycle $\langle s,v_2,v_3\rangle$ which has weight $p^3(\frac{-\eta}{p^3})+\eta-1=-1$. However, suppose $L=4$, and we removed the edge (v_3,s). Then $\langle s,v_2,v_3\rangle$ is no longer a cycle, and the path (s,v_1,v_3,v_4) will have weight $p^4(\frac{\eta}{p^3}-1)+\eta-1=\eta-p\eta-p^4-1>0$, assuming η is large and p is not close to 1. However, the path (s,v_1,v_3,v_4) would lead to a discounted negative cycle with weight $-p^4$. The algorithm from the deterministic setting cannot compare two paths without knowing the final cycle length.

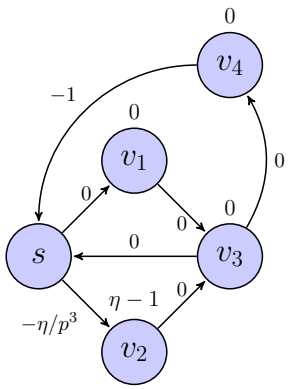


Figure A.3: Example demonstrating that multiple possible final path lengths must be considered.

A.4.3 Pseudocode for the corrected polynomial pricing scheme

We provide as Algorithm 4 the full pseudocode for our adapted version of the polynomial pricing algorithm provided by Glorie et al. [100]. This is called as a subroutine by the failure-aware PICEF

adaptation in Section 3.3.5. As discussed in Chapter 3, this can be used to find positive price cycles but not chains.

In Algorithm 4, for a fixed source, let $d_i(v)$ represent the computed distance from that source to v after the ith step of the algorithm, where $d_0(v)$ represents the distances before any steps are performed. Distance is defined as the sum of the edge weights in the computed path. Let L be the maximum allowable cycle length. Finally, let A be the set of altruist donors and let P be the set of donor-patient pairs. The function GETNEGATIVECYCLES is called with the reduced graph $G = (V, E)$—with altruistic donors and their incident edges removed—and cycle cap L.

Algorithm 4 Corrected polynomial-time Bellman-Ford search for negative weight cycles.

function GETNEGATIVECYCLES($G = (V, E), L$)
  $\mathcal{C} \leftarrow \emptyset$ ▷ Accumulator set for negative weight cycles
  for each $s \in P$ **do**
    $N \leftarrow L - 1$ ▷ Set maximum step number based on cycle cap
    $pred_0(v) = \emptyset \ \ \forall v \in P$
    $d_0(s) = 0$ ▷ Distance from source to source is zero
    $d_0(v) = \infty \ \ \forall v \neq s \in P$ ▷ Distance at step 0 to other vertices is infinite
    for $i \in \{1, \ldots, N\}$ **do**
      $d_i(v) = d_{i-1}(v) \ \ \forall v \neq s \in P$
      $pred_i(v) = pred_{i-1}(v) \ \ \forall v \neq s \in P$
      for each $(u, v) \in E$ **do**
        if $v \notin$ TRAVERSEPREDS$(u, pred, i-1)$ **then** ▷ Avoid loops in path
          if $d_{i-1}(u) + w(u, v) < d_i(v)$ **then** ▷ If this step decreases the distance to node
            $d_i(v) \leftarrow d_{i-1}(u) + w(u, v)$ ▷ Update to shorter distance
            $pred_i(v) \leftarrow (u, i-1)$ ▷ Store correct predecessor
    for each $v \neq s \in P$ **do** ▷ Find negative weight cycles with s as the source
      if $d_N(v) + w(v, s) < 0$ **then**
        $\mathcal{C} \leftarrow \mathcal{C} \cup$ TRAVERSEPREDS$(v, pred, N)$
  return $\mathcal{C}$
function TRAVERSEPREDS($v, pred, n$)
  $c \leftarrow []$ ▷ Start with an empty list (representing a cycle)
  $curr \leftarrow v$
  while $curr \neq \emptyset$ **do** ▷ Until we reach the source node ...
    $c \leftarrow curr + c$ ▷ Add predecessor to path
    $(u, i) \leftarrow pred_n(curr)$ ▷ Get predecessor of predecessor
    $curr \leftarrow u; \ \ n \leftarrow i$
  return c

Constrained to the case of finding cycles only—and recalling that Section 3.2 shows that this algorithm cannot be used for chains—Algorithm 4 can be seen as a slightly modified version of the pricing algorithm initially proposed by Glorie et al. [100]. They did not prove correctness for

that algorithm; we do that now in Theorem 21 for our modified Algorithm 4.

Theorem 21. *If there is a negative cycle in the graph, Algorithm 4 will return at least one negative cycle.*

Proof. We will show that if there is a negative cycle c that we do not find, there must exist a negative cycle with strictly fewer vertices. Thus, for any negative cycle c that we do not return, there must exist a negative cycle p^*q^* with fewer vertices. So, there exists a negative cycle with no negative cycles smaller than it, which our algorithm finds and returns.

Say $c = \langle v_1, v_2, \ldots, v_n \rangle$ is that negative cycle that we do not return. Without loss of generality, assume that c contains the shortest path from v_1 to v_n; if it does not, then that cycle containing the shortest path is also a negative cycle.

Consider running the modified Bellman-Ford method with v_1 as the source. Since by assumption the algorithm does not find c, it must compute a different path from v_1 to v_n than the one in c. We know that the computed path is not shorter, since c contains the shortest path to v_n. Without loss of generality, assume it is strictly longer; were it equal in weight, we would be done (as this is a negative cycle that is found by the algorithm as well).

The only way our modified Bellman-Ford method does not compute the shortest path to v_n is if there exists some vertex v_{split}, where $v_{split} \in c$, but the shortest path to v_{split} is not in c. This can occur due to the modification that prevents loops in shortest paths. Let p be the shorter path from v_1 to v_{split}, and let p_c be the path from v_1 to v_{split} in c. Let q be the path from v_{split} to v_n in c, plus the edge (v_n, v_1). This is shown in Figure A.4.

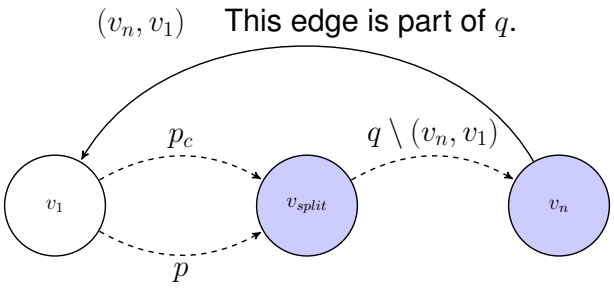


Figure A.4: Widget with a negative cycle and existence of a shorter negative cycle. Dotted arrows are paths that contain zero or more vertices (and thus one or more edges).

Then $c = p_c v_{split} q$. Also, since the weights on the paths are $w(p) < w(p_c)$, we have $w(pq) < w(p_c q) = w(c) < 0$.

We know that $c = p_c q$ satisfies the cycle size cap, since it is valid by assumption. For any path ρ, let $|\rho|$ represent the number of vertices in that path.

Claim 21.1. $|p| \leq |p_c|$.

Proof: By way of contradiction, assume $|p_c| < |p|$. Then, the sequence of updates along p_c will reach v_{split} before p does—which means we will have computed p_c. Even though we may compute p later, we will still be able to build off of path p_c: this is because we maintain the full 2D

predecessor array, which is necessary for other reasons. Therefore we will go on to compute the full $p_c q$, which is a contradiction.

This issue may arise again when computing a path to v_n with p_c as the base; in the process of computing q with p_c as the base, there may exist some vertex v'_{split} that causes the same issue as v_{split}. In that case, our logic can be applied recursively until no such vertex like v'_{split} exists. ■

Using Claim 21.1, we can ignore the cycle cap for the rest of the proof, since all cycles discussed will have size $|pq| \leq |p_c q| = |c|$, which is legal by assumption.

At this point, we have p and q such that pq is a *circuit* (i.e., a path that starts and ends at the same vertex but which might not be a cycle because it might visit some vertices more than once), and $w(pq) < 0$. Claim 21.2 gives a tool that we will use to finish the proof of the theorem through repeated use.

Claim 21.2. *In a directed graph, if there exists a circuit π that is not a cycle and $w(\pi) < 0$, then there exists a circuit π' where $w(\pi') < 0$ and $|\pi'| < |\pi|$.*

Proof: Because π is a circuit but not a cycle, it must have an intersection. Therefore one can split π into two non-empty paths, α and β, where α and β intersect. Thus there exists $v_\cap \in \alpha$ where $v_\cap \in \beta$. Then $\alpha = \alpha_1 v_\cap \alpha_2$ and $\beta = \beta_1 v_\cap \beta_2$, where α_1, α_2, β_1, and β_2 are nonempty.

We know that α_1 is a path from some vertex u to $v_\cap$ and that β_2 is a path from $v_\cap$ back to u. Similarly, β_1 is a path from some vertex u' to $v_\cap$ and α_2 is a path from $v_\cap$ back to u'.

This implies that both $\alpha_1\beta_2$ and $\alpha_2\beta_1$ are circuits. Because $\alpha_1, \alpha_2, \beta_1$, and β_2 are nonempty, $\alpha_1\beta_2$ does not contain u', and $\alpha_2\beta_1$ does not contain u. Therefore $|\alpha_1\beta_2| < |\pi|$ and $|\alpha_2\beta_1| < |\pi|$.

Furthermore, since $w(\pi) = w(\alpha_1\beta_2) + w(\alpha_2\beta_1) < 0$, we know that at least one of $\alpha_1\beta_2$ and $\alpha_2\beta_1$ must be negative. This shows the existence of a negative circuit with strictly fewer vertices. ■

We now return to the proof of the theorem. Recall that we have p and q such that pq is a circuit, and $w(pq) < 0$.

By the claim above, the presence of a negative circuit pq implies that p and q do not intersect, or that there exists a negative circuit $p'q'$ that has fewer vertices. If p and q were not intersecting, pq would be a shorter path than $p_c q$, which violates the assumption that c contains the shortest path. Thus, p and q do intersect. Therefore, there exists a negative circuit $p'q'$ that has fewer vertices.

Since we can only shrink pq, $p'q'$, and so on in this fashion a finite number of times, there must exist some negative circuit p^*q^* where p^*q^* does not self-intersect; so, the negative circuit is a cycle. □

A.5 Tabulated experimental results

In this section, we restate the experimental results shown in Figures 3.8 and 3.9 of Section 3.3.4 in the body of the paper, but now including statistics that were not possible to show in that figure.

Table A.1: Position-indexed formulation experiments on real UNOS match runs.

Method		$K=2$	$K=3$	$K=4$	$K=5$	$K=6$	$K=7$	$K=8$	$K=9$	$K=10$	$K=11$	$K=12$
PICEF	Mean	0.45	0.46	0.51	0.58	0.63	0.65	0.71	0.79	0.85	0.93	0.98
	Stdev	0.06	0.10	0.12	0.17	0.25	0.28	0.32	0.44	0.48	0.63	0.62
	Min	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31
	Max	0.62	0.97	1.02	1.42	1.72	2.12	2.22	3.12	3.52	4.82	3.62
HPIEF	Mean	0.98	1.11	1.06	1.22	1.17	1.27	1.38	1.37	1.44	1.50	1.56
	Stdev	0.39	0.42	0.44	0.49	0.53	0.52	0.59	0.55	0.61	0.67	0.73
	Min	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31
	Max	1.87	1.97	2.37	2.57	3.27	2.62	3.58	3.07	3.42	4.07	4.32
BNP-POLY	Mean	0.18	0.19	0.19	0.20	0.21	0.25	0.47	0.73	1.37	0.35	2.99
	Stdev	0.06	0.07	0.08	0.09	0.10	0.56	3.78	8.25	11.60	1.52	45.28
	Min	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
	Max	0.42	0.42	0.57	0.57	0.67	9.33	63.22	138.46	154.74	24.06	759.13
CG-TSP	Mean	3.43	3.54	6.81	14.48	19.13	30.91	27.03	30.79	33.17	29.47	29.22
	Stdev	0.46	1.04	30.62	116.81	215.58	303.41	249.35	303.51	306.18	302.89	302.87
	Min	1.97	1.92	1.92	1.92	1.92	1.92	1.92	1.97	1.92	1.92	1.92
	Max	4.68	13.34	410.07	1401.87	3600.09	3600.10	3600.08	3600.08	3600.13	3600.08	3600.04
BNP-DFS	Mean	0.14	0.14	0.14	0.17	0.30	1.17	26.77	32.13	58.63	78.68	120.93
	Stdev	0.04	0.04	0.06	0.16	1.01	6.17	289.50	263.92	401.73	461.29	605.34
	Min	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
	Max	0.27	0.32	0.52	1.97	14.24	72.14	3600.00	3600.00	3600.00	3600.00	3600.00
BNP-PICEF	Mean	0.25	0.40	0.55	0.77	1.05	1.52	2.55	3.10	3.84	5.98	8.16
	Stdev	0.08	0.14	0.19	0.32	0.43	1.40	10.99	11.09	8.52	21.45	36.19
	Min	0.03	0.06	0.07	0.06	0.07	0.12	0.12	0.11	0.11	0.11	0.17
	Max	0.42	0.77	0.97	2.77	2.42	18.20	185.24	186.95	135.21	283.75	550.37

Table A.2: Position-indexed formulation experiments on real NLDKSS match runs.

Method		$K = 2$	$K = 3$	$K = 4$	$K = 5$	$K = 6$	$K = 7$	$K = 8$	$K = 9$	$K = 10$	$K = 11$	$K = 12$
PICEF	Mean	0.12	0.15	0.18	0.21	0.27	0.32	0.33	0.36	0.43	0.45	0.48
	Stdev	0.02	0.03	0.04	0.05	0.10	0.21	0.12	0.16	0.22	0.19	0.18
	Min	0.09	0.09	0.10	0.12	0.13	0.14	0.16	0.17	0.18	0.17	0.19
	Max	0.17	0.24	0.28	0.37	0.54	1.13	0.69	0.92	1.21	1.05	1.00
HPIEF	Mean	0.26	0.27	0.28	0.31	0.34	0.38	0.43	0.45	0.49	0.54	0.57
	Stdev	0.04	0.05	0.06	0.07	0.09	0.15	0.20	0.17	0.20	0.26	0.25
	Min	0.19	0.19	0.20	0.20	0.21	0.21	0.22	0.22	0.24	0.24	0.26
	Max	0.35	0.40	0.43	0.49	0.57	0.93	1.13	0.97	1.12	1.29	1.40
BNP-POLY	Mean	0.16	0.23	0.18	92.63	0.82	0.24	0.64	0.84	0.85	212.58	216.34
	Stdev	0.06	0.27	0.07	369.70	2.00	0.15	1.11	1.20	1.45	846.86	846.07
	Min	0.07	0.07	0.06	0.05	0.05	0.04	0.10	0.09	0.10	0.11	0.10
	Max	0.28	1.27	0.34	1571.45	8.57	0.72	4.85	4.25	5.10	3600.00	3600.00
CG-TSP	Mean	0.95	41.01	753.96	650.07	255.26	256.39	260.80	129.00	221.10	229.37	4.69
	Stdev	0.42	141.39	1349.06	1167.96	846.22	845.67	844.40	418.79	845.20	843.49	9.84
	Min	0.47	0.58	0.56	0.54	0.58	0.51	0.52	0.47	0.54	0.46	0.50
	Max	2.12	604.14	3600.00	3600.00	3600.00	3600.00	3600.00	1780.46	3600.00	3600.00	42.72
BNP-DFS	Mean	0.13	0.18	0.78	170.12	238.76	329.46	1090.72	1473.75	1888.86	2386.44	2506.19
	Stdev	0.08	0.16	1.37	653.91	841.62	842.37	1430.36	1540.36	1649.63	1619.09	1540.93
	Min	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04
	Max	0.36	0.71	5.83	2784.96	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
BNP-PICEF	Mean	0.19	0.23	0.30	233.49	53.32	0.83	215.66	5.25	3.18	4.89	19.68
	Stdev	0.12	0.08	0.14	845.94	210.51	0.76	846.14	9.95	5.49	10.34	57.00
	Min	0.09	0.09	0.11	0.23	0.27	0.33	0.41	0.47	0.41	0.53	0.65
	Max	0.61	0.42	0.71	3600.00	895.37	3.65	3600.00	40.09	23.65	43.03	245.17

Table A.3: Position-indexed formulation experiments for $|P| = 300$, $|A| = 3$

Method		$K = 2$	$K = 3$	$K = 4$	$K = 5$	$K = 6$	$K = 7$	$K = 8$	$K = 9$	$K = 10$	$K = 11$	$K = 12$
PICEF	Mean	1.00	1.40	2.04	2.89	4.06	5.63	7.73	10.99	14.52	17.87	20.18
	Stdev	0.12	0.32	0.51	0.77	0.92	1.28	2.23	3.13	4.54	5.90	6.89
	Min	0.77	0.94	1.27	1.87	2.69	3.60	5.05	6.30	6.78	9.33	10.28
	Max	1.37	2.42	3.80	5.42	6.76	8.24	13.00	18.33	25.50	33.84	36.46
CG-TSP	Mean	8.44	184.96	918.97	1221.19	1709.03	1682.64	1632.00	1908.47	1839.85	1993.84	1903.64
	Stdev	1.04	403.94	913.85	1031.16	1316.14	1132.89	1075.43	1206.85	1128.78	1173.08	1062.16
	Min	6.48	9.07	12.71	12.07	18.49	22.77	35.26	131.48	86.87	21.13	28.67
	Max	10.62	1804.94	2732.26	3600.07	3600.10	3600.09	3600.08	3600.11	3600.10	3600.10	3600.16
BNP-PICEF	Mean	2.39	14.77	110.11	210.89	635.14	1088.77	1561.96	2420.79	1693.39	2514.56	2791.20
	Stdev	5.18	63.60	330.96	444.70	827.57	869.08	1061.65	877.50	1312.65	1029.71	993.59
	Min	0.72	1.17	1.82	3.10	5.65	12.17	14.63	538.04	37.81	48.04	67.38
	Max	24.74	326.27	1451.10	1639.32	3043.01	3088.56	3601.15	3600.78	3600.65	3600.52	3600.42
BNP-DFS	Mean	14.83	17.13	383.40	694.57	2644.81	3494.87	3600.00	3600.00	3600.00	3600.00	3600.00
	Stdev	69.43	71.79	714.88	793.47	1026.37	285.67	0.00	0.00	0.00	0.00	0.00
	Min	0.49	0.59	1.34	14.21	284.94	2632.71	3600.00	3600.00	3600.00	3600.00	3600.00
	Max	354.99	367.77	1816.74	2166.63	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
HPIEF	Mean	3.04	3.47	4.16	5.14	6.52	8.12	10.40	13.67	16.74	20.59	22.29
	Stdev	0.51	0.79	0.85	1.27	1.98	2.26	2.88	3.67	3.44	5.33	6.07
	Min	2.44	2.52	2.87	3.40	4.35	5.17	6.96	8.83	12.16	9.93	10.11
	Max	4.10	6.35	5.85	8.35	12.59	14.49	17.92	22.78	24.24	31.85	37.55
BNP-POLY	Mean	2.08	81.96	238.82	306.05	631.69	1074.37	1000.78	2300.22	1328.40	2076.98	2515.60
	Stdev	5.30	353.34	581.46	654.90	928.03	931.64	1080.38	1060.62	1174.75	1072.71	1347.17
	Min	0.54	0.57	0.62	0.72	0.89	1.04	0.87	4.68	1.64	2.50	1.87
	Max	26.56	1800.68	1801.42	1801.40	2890.70	3600.61	3600.72	3601.89	3600.86	3601.85	3602.00

Table A.4: Position-indexed formulation experiments for $|P| = 300$, $|A| = 6$

Method		$K = 2$	$K = 3$	$K = 4$	$K = 5$	$K = 6$	$K = 7$	$K = 8$	$K = 9$	$K = 10$	$K = 11$	$K = 12$
PICEF	Mean	1.45	2.14	2.96	4.36	5.86	8.03	10.02	12.69	17.82	23.35	26.19
	Stdev	0.53	0.92	1.04	1.88	2.60	3.31	4.48	6.06	10.81	14.57	17.75
	Min	0.92	1.24	1.89	2.49	3.45	4.47	4.95	6.45	8.45	9.83	12.49
	Max	3.13	5.76	5.93	10.29	12.67	17.20	23.42	29.72	58.28	66.11	96.83
CG-TSP	Mean	63.67	1783.81	2825.67	3274.28	3469.97	3528.59	3507.68	3493.12	3332.71	3320.82	3103.94
	Stdev	249.48	1021.00	835.84	599.43	443.61	350.18	338.77	344.12	613.11	577.68	820.61
	Min	8.01	32.68	1437.28	1811.15	1810.83	1813.06	1989.48	2012.35	1823.48	1821.29	975.77
	Max	1285.66	3600.08	3600.09	3600.10	3600.11	3600.09	3600.08	3600.09	3600.10	3600.09	3600.11
BNP-PICEF	Mean	2.09	124.04	871.35	1533.14	1902.03	2567.05	2786.84	2403.03	2456.11	2041.20	1273.52
	Stdev	2.80	380.96	1139.42	1238.55	1272.68	1220.26	1066.82	1193.46	1212.64	952.01	1046.65
	Min	0.79	1.19	1.97	3.65	7.76	14.43	21.98	43.21	78.77	112.14	168.46
	Max	12.41	1600.58	3262.90	3580.66	3602.58	3601.76	3600.80	3600.62	3600.44	3600.37	3600.24
BNP-DFS	Mean	2.34	440.67	974.91	2324.52	3540.79	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
	Stdev	4.92	765.51	1247.14	1097.18	290.09	0.00	0.00	0.00	0.00	0.00	0.00
	Min	0.54	1.09	12.73	349.24	2119.63	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
	Max	22.03	1805.45	3600.25	3600.02	3600.01	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
HPIEF	Mean	3.22	3.78	4.84	6.44	8.78	13.51	19.45	25.49	53.11	58.12	55.49
	Stdev	0.56	0.67	1.17	2.13	3.38	8.70	14.74	12.80	43.89	44.44	29.12
	Min	2.49	2.92	3.42	4.00	4.80	5.22	6.60	6.90	11.34	11.09	16.52
	Max	4.55	5.63	7.88	11.71	16.27	42.34	81.69	55.29	236.82	197.45	110.75
BNP-POLY	Mean	73.01	222.84	815.13	1168.37	1902.71	2611.16	2513.36	2346.77	2470.76	2264.23	2140.75
	Stdev	352.77	583.10	1171.55	1205.56	1263.29	1166.98	1282.07	1173.08	1073.70	846.97	1011.70
	Min	0.57	0.64	0.74	0.89	1.09	2.14	2.44	12.42	25.30	264.09	94.33
	Max	1801.19	1801.23	3602.15	3601.76	3601.45	3601.79	3601.75	3601.90	3601.85	3601.72	3600.67

Table A.5: Position-indexed formulation experiments for $|P| = 300$, $|A| = 15$

Method		$K=2$	$K=3$	$K=4$	$K=5$	$K=6$	$K=7$	$K=8$	$K=9$	$K=10$	$K=11$	$K=12$
PICEF	Mean	1.36	2.00	2.93	5.06	7.44	14.52	17.20	21.79	24.90	31.46	29.79
	Stdev	0.44	0.64	0.72	2.27	3.94	20.99	13.30	7.76	8.87	13.39	6.64
	Min	1.04	1.49	2.17	2.79	3.87	4.50	5.75	9.18	11.01	15.21	18.42
	Max	3.26	4.15	5.32	12.68	21.23	114.52	77.98	38.47	58.61	85.77	47.16
CG-TSP	Mean	367.16	3515.05	3600.09	3600.09	3535.69	3529.94	3536.72	3430.70	3461.73	3279.96	3220.04
	Stdev	640.14	346.51	0.04	0.03	315.51	343.66	310.45	462.86	469.31	642.91	613.91
	Min	16.67	1857.78	3600.06	3600.06	1990.03	1846.38	2015.84	1948.32	1833.44	1838.52	1968.78
	Max	1822.65	3600.17	3600.20	3600.16	3600.25	3600.21	3600.20	3600.18	3600.15	3600.24	3600.16
BNP-PICEF	Mean	149.79	445.86	856.61	1712.61	1095.22	535.43	201.36	178.40	162.61	159.80	218.11
	Stdev	487.53	754.76	854.16	1192.86	1126.06	803.12	479.10	343.89	83.78	70.62	111.01
	Min	0.87	1.42	2.62	4.75	9.64	18.01	28.09	48.20	70.51	53.85	79.26
	Max	1802.03	1803.15	1830.94	3602.67	3601.98	1867.16	1832.55	1843.38	404.82	339.64	523.09
BNP-DFS	Mean	141.93	683.49	1274.19	3401.45	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
	Stdev	476.11	850.91	966.39	410.86	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Min	0.74	3.95	81.43	2091.66	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
	Max	1801.39	1842.76	3600.22	3600.04	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
HPIEF	Mean	3.11	3.85	4.72	6.33	10.18	13.13	18.62	24.36	29.43	31.45	32.13
	Stdev	1.02	1.57	2.09	3.26	5.96	8.86	13.97	16.56	16.73	17.77	17.75
	Min	1.32	1.32	1.32	1.32	1.32	1.32	1.32	1.32	1.32	1.32	1.32
	Max	4.85	7.20	9.52	13.59	20.95	35.23	68.89	73.93	67.31	58.72	66.14
BNP-POLY	Mean	74.83	611.66	991.62	1663.90	1419.18	632.51	283.39	366.94	475.84	1038.21	1230.37
	Stdev	353.01	867.89	872.00	1390.32	1210.69	837.82	770.75	616.38	623.12	1099.15	1025.71
	Min	0.74	0.89	1.07	1.57	0.39	0.37	0.37	0.37	0.37	0.37	0.37
	Max	1804.07	2318.25	1805.02	3601.10	3600.77	1817.17	3600.05	2021.28	2109.74	3600.14	3600.01

Table A.6: Position-indexed formulation experiments for $|P| = 300$, $|A| = 75$

Method		$K = 2$	$K = 3$	$K = 4$	$K = 5$	$K = 6$	$K = 7$	$K = 8$	$K = 9$	$K = 10$	$K = 11$	$K = 12$
PICEF	Mean	2.06	2.97	4.03	5.42	6.96	8.90	10.38	12.41	13.75	15.14	16.95
	Stdev	0.70	1.13	1.18	1.91	2.41	3.78	2.70	2.53	2.86	2.31	3.40
	Min	1.37	1.97	2.64	3.54	4.32	5.30	6.93	7.75	8.55	10.36	10.34
	Max	4.47	7.43	8.08	13.19	16.37	25.68	17.05	18.69	23.80	19.35	26.39
CG-TSP	Mean	3168.00	3600.18	3544.38	3451.82	3110.08	3056.40	2575.13	2339.98	2222.51	1524.77	1122.56
	Stdev	828.44	0.07	191.05	409.89	539.91	560.83	671.91	799.95	929.42	1030.21	881.64
	Min	523.02	3600.11	2812.72	2097.94	2237.32	2001.36	1339.74	238.59	410.76	22.79	20.75
	Max	3600.25	3600.36	3600.38	3600.43	3600.41	3600.33	3600.61	3600.32	3600.41	3600.16	2753.27
BNP-PICEF	Mean	1.76	2.12	3.51	5.61	7.96	12.27	14.82	18.70	21.42	27.41	33.25
	Stdev	0.11	0.27	1.19	2.12	2.73	3.70	4.64	4.29	5.82	8.47	10.15
	Min	1.52	1.69	1.82	3.05	2.97	6.81	7.46	10.26	11.22	13.93	17.36
	Max	2.05	3.02	5.96	10.74	16.08	19.74	27.74	28.28	37.68	43.10	55.48
BNP-DFS	Mean	4.66	78.58	1984.70	3600.01	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
	Stdev	0.99	22.89	748.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Min	3.15	34.57	776.95	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
	Max	6.70	122.86	3143.17	3600.02	3600.01	3600.00	3600.01	3600.01	3600.01	3600.01	3600.00
HPIEF	Mean	3.85	4.70	5.88	7.38	8.81	11.55	13.47	16.08	18.44	19.86	20.05
	Stdev	1.32	1.38	1.74	2.46	2.75	3.77	3.80	4.08	4.46	4.44	4.78
	Min	2.72	3.17	3.80	4.50	5.15	6.00	7.18	7.78	9.63	11.01	12.66
	Max	9.64	9.98	12.42	16.14	17.15	21.58	20.98	22.90	26.86	28.19	31.60
BNP-POLY	Mean	2.61	3.36	4.41	4.91	5.42	6.97	9.74	20.70	67.07	155.99	223.71
	Stdev	0.19	0.34	0.67	0.94	1.03	2.52	6.76	31.39	94.52	379.20	426.11
	Min	2.17	2.80	3.42	3.63	3.87	3.55	3.80	4.48	6.88	5.25	8.64
	Max	3.02	4.08	5.53	7.36	8.21	14.15	33.77	166.95	398.11	1812.22	1806.62

Table A.7: Position-indexed formulation experiments for $|P| = 500$, $|A| = 5$

Method		$K=2$	$K=3$	$K=4$	$K=5$	$K=6$	$K=7$	$K=8$	$K=9$	$K=10$	$K=11$	$K=12$
PICEF	Mean	3.33	5.69	9.68	16.06	29.55	49.75	79.13	101.60	161.25	324.80	378.28
	Stdev	0.60	1.25	2.73	4.38	10.00	16.02	23.53	32.33	78.74	309.10	405.84
	Min	2.67	4.17	6.35	9.61	12.19	17.20	21.20	24.36	36.23	41.63	50.26
	Max	4.92	8.58	17.21	24.63	50.08	70.96	126.06	160.64	345.49	1333.83	1864.47
CG-TSP	Mean	86.08	1519.67	2888.52	3307.18	3500.92	3542.08	3600.09	3600.09	3600.09	3600.09	3600.10
	Stdev	163.39	1194.67	827.72	675.52	356.92	267.64	0.01	0.01	0.01	0.01	0.02
	Min	17.80	43.13	1528.27	1445.40	1937.79	2233.38	3600.07	3600.07	3600.07	3600.07	3600.07
	Max	870.78	3600.22	3600.18	3600.18	3600.15	3600.13	3600.13	3600.13	3600.14	3600.14	3600.16
BNP-PICEF	Mean	489.87	408.98	1033.60	1748.72	1900.07	2549.32	2856.84	3275.68	3360.62	3546.67	3600.09
	Stdev	915.90	712.01	1148.29	1281.58	1060.35	1105.73	841.48	652.43	561.54	261.92	0.04
	Min	3.05	4.65	8.98	22.14	47.29	98.88	1832.39	1868.01	1892.93	2263.54	3600.02
	Max	3601.12	1805.88	3601.60	3602.22	3600.93	3600.99	3600.45	3600.39	3600.36	3600.21	3600.16
BNP-DFS	Mean	584.00	609.09	1578.12	3595.49	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
	Stdev	978.95	969.92	962.98	22.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Min	3.02	8.48	220.63	3487.09	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
	Max	3600.25	3600.31	3600.07	3600.01	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
HPIEF	Mean	13.14	15.33	17.49	20.30	24.34	30.93	40.59	50.92	102.03	187.22	246.18
	Stdev	2.89	3.46	3.24	3.55	3.62	6.46	10.33	19.93	88.83	175.34	310.59
	Min	9.78	11.01	12.53	14.49	17.72	19.15	22.78	30.32	42.70	34.58	36.96
	Max	23.73	24.60	24.29	26.37	30.25	49.24	73.30	135.55	475.85	831.95	1494.91
BNP-POLY	Mean	476.19	398.72	987.34	1995.51	1882.08	2376.94	2595.14	3027.64	3313.34	3529.60	3600.32
	Stdev	924.35	875.52	1015.36	1191.54	1290.49	1216.20	1025.58	974.20	657.83	346.79	0.08
	Min	2.07	2.24	2.52	3.82	4.10	5.00	10.44	53.25	1804.95	1830.66	3600.12
	Max	3600.11	3600.10	3600.21	3600.42	3600.36	3600.61	3600.61	3600.67	3600.57	3600.50	3600.45

Table A.8: Position-indexed formulation experiments for $|P| = 500$, $|A| = 10$

Method		$K = 2$	$K = 3$	$K = 4$	$K = 5$	$K = 6$	$K = 7$	$K = 8$	$K = 9$	$K = 10$	$K = 11$	$K = 12$
PICEF	Mean	3.87	6.41	10.22	16.54	26.81	54.10	111.83	321.99	286.20	392.19	323.20
	Stdev	0.92	2.03	3.55	6.36	10.77	39.64	124.63	297.04	224.94	280.57	198.14
	Min	2.72	4.12	6.35	9.33	12.19	15.67	19.18	26.11	33.20	67.86	70.61
	Max	6.20	13.39	21.53	33.38	55.39	190.56	518.68	1142.68	879.51	1182.30	927.75
CG-TSP	Mean	406.15	3116.63	3600.10	3600.11	3600.11	3600.11	3600.10	3600.11	3600.11	3600.11	3600.12
	Stdev	630.53	844.15	0.02	0.02	0.02	0.04	0.01	0.03	0.02	0.04	0.07
	Min	53.28	517.29	3600.07	3600.07	3600.07	3600.08	3600.08	3600.08	3600.07	3600.08	3600.07
	Max	1839.96	3600.18	3600.17	3600.16	3600.17	3600.28	3600.13	3600.24	3600.16	3600.29	3600.40
BNP-PICEF	Mean	829.37	873.28	1593.51	2389.64	2910.67	3534.89	3056.63	2928.76	2817.08	2619.23	2645.57
	Stdev	1340.08	1029.25	1054.00	1206.53	1075.62	320.66	1044.62	932.67	988.09	763.68	660.52
	Min	3.20	6.28	11.97	26.78	90.19	1963.99	177.12	520.29	820.70	1037.74	1193.26
	Max	3601.57	3601.87	3601.30	3601.08	3600.76	3600.50	3600.32	3600.23	3600.18	3600.05	3600.04
BNP-DFS	Mean	727.04	1249.56	2754.81	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
	Stdev	992.61	827.81	682.38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Min	2.90	19.27	1571.31	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
	Max	3438.77	1826.38	3600.07	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
HPIEF	Mean	12.79	15.37	19.41	29.69	45.92	94.99	242.75	434.08	484.99	884.31	568.17
	Stdev	1.42	2.33	3.64	7.00	18.54	72.01	268.11	309.07	327.06	695.32	469.58
	Min	11.04	12.04	14.09	16.02	18.48	21.61	28.04	31.14	36.64	127.14	66.01
	Max	16.11	20.78	26.40	41.07	113.75	374.04	1250.39	1096.22	1167.39	2230.86	2155.83
BNP-POLY	Mean	600.79	726.93	1514.72	2378.45	2816.66	3528.67	3034.58	3051.69	2965.69	3229.29	3437.41
	Stdev	969.67	877.81	1099.85	1215.75	1132.17	351.68	1092.85	943.16	868.24	611.16	443.14
	Min	2.27	2.90	3.67	4.28	9.79	1805.82	15.00	76.72	398.26	1948.58	2135.48
	Max	3600.17	1802.41	3600.26	3600.46	3600.59	3600.71	3600.59	3600.52	3600.37	3600.34	3600.25

Table A.9: Position-indexed formulation experiments for $|P| = 500$, $|A| = 25$

Method		$K = 2$	$K = 3$	$K = 4$	$K = 5$	$K = 6$	$K = 7$	$K = 8$	$K = 9$	$K = 10$	$K = 11$	$K = 12$
PICEF	Mean	5.13	8.81	14.79	31.28	55.40	91.01	159.51	210.90	263.30	326.07	321.93
	Stdev	3.10	4.88	7.90	16.29	31.14	40.02	74.17	106.60	110.04	128.59	115.14
	Min	3.22	5.65	8.10	14.42	18.82	28.81	48.51	51.74	74.37	81.94	90.87
	Max	18.24	23.86	40.36	69.45	130.15	172.46	319.40	494.03	520.89	520.57	530.59
CG-TSP	Mean	3148.56	3600.21	3600.23	3600.20	3600.21	3600.22	3600.21	3600.21	3600.20	3600.24	3600.31
	Stdev	728.91	0.14	0.19	0.14	0.13	0.16	0.15	0.16	0.13	0.17	0.18
	Min	1009.67	3600.10	3600.11	3600.10	3600.12	3600.12	3600.11	3600.12	3600.11	3600.09	3600.04
	Max	3600.17	3600.64	3600.85	3600.73	3600.63	3600.66	3600.66	3600.68	3600.59	3600.75	3600.67
BNP-PICEF	Mean	513.65	881.36	1955.77	1417.19	557.22	454.23	677.44	1179.01	1496.40	1578.70	1627.19
	Stdev	804.07	1023.76	1423.30	1143.07	686.23	201.31	347.69	493.79	579.61	534.03	515.77
	Min	3.15	6.13	20.00	47.80	105.26	207.65	292.64	654.54	583.22	720.50	798.70
	Max	1803.61	3601.50	3601.39	3600.55	2011.26	1138.30	2126.49	2554.51	2904.98	2730.44	2797.31
BNP-DFS	Mean	796.98	1108.07	3599.33	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
	Stdev	890.74	975.82	3.35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Min	4.40	78.78	3582.91	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
	Max	1811.33	3600.28	3600.03	3600.01	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
HPIEF	Mean	13.64	16.70	25.11	50.19	64.58	88.26	125.61	171.61	223.86	231.21	207.82
	Stdev	2.96	3.36	7.83	46.74	25.11	33.30	57.05	64.98	75.50	76.25	67.91
	Min	9.86	11.91	14.81	19.35	29.36	42.41	47.14	56.01	67.62	90.86	105.80
	Max	21.23	24.67	44.65	229.70	122.98	155.02	248.62	286.95	357.62	369.33	312.53
BNP-POLY	Mean	610.17	939.96	1883.09	1242.21	621.96	428.89	618.50	896.95	1988.17	2418.15	2784.83
	Stdev	823.88	1032.01	1470.65	1100.86	957.44	721.69	879.42	957.15	1234.27	1160.63	942.70
	Min	3.12	4.00	6.65	9.96	13.62	14.65	21.21	31.33	39.06	68.33	151.56
	Max	1802.71	3600.11	3600.38	3600.47	3600.23	2307.41	3600.21	3600.13	3600.09	3600.27	3600.16

Table A.10: Position-indexed formulation experiments for $|P| = 500$, $|A| = 125$

Method		$K = 2$	$K = 3$	$K = 4$	$K = 5$	$K = 6$	$K = 7$	$K = 8$	$K = 9$	$K = 10$	$K = 11$	$K = 12$
PICEF	Mean	7.08	9.33	13.04	17.54	22.90	28.08	36.50	43.25	52.92	64.75	75.29
	Stdev	1.56	1.61	2.52	3.43	5.08	5.70	5.79	7.13	10.70	11.86	20.55
	Min	4.53	6.40	9.09	11.87	16.00	19.19	25.22	27.78	34.55	49.41	48.34
	Max	11.31	13.87	18.28	24.26	38.67	38.91	44.74	59.57	76.93	91.35	107.55
CG-TSP	Mean	3600.55	3600.76	3600.71	3600.81	3600.80	3600.69	3600.86	3467.07	3328.12	3206.21	2258.82
	Stdev	0.34	0.46	0.41	0.36	0.45	0.38	0.54	454.31	625.70	708.71	1165.52
	Min	3600.15	3600.26	3600.27	3600.28	3600.21	3600.21	3600.24	1838.14	1825.68	1847.26	102.74
	Max	3601.35	3601.98	3601.80	3601.61	3601.98	3601.54	3602.45	3602.05	3601.67	3601.78	3601.04
BNP-PICEF	Mean	78.89	8.18	13.75	26.74	37.15	48.15	80.05	100.20	140.66	187.31	238.27
	Stdev	351.95	1.51	5.48	7.94	9.18	11.02	19.44	22.66	34.57	37.81	46.79
	Min	5.68	6.31	6.76	10.52	20.57	22.27	53.48	56.36	76.52	122.53	145.24
	Max	1803.09	11.65	26.37	39.92	54.66	70.94	126.77	140.54	219.19	254.26	332.33
BNP-DFS	Mean	125.75	2029.37	3600.01	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
	Stdev	346.25	466.77	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Min	32.45	1048.41	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
	Max	1821.02	2831.76	3600.02	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
HPIEF	Mean	14.00	17.78	23.48	32.89	44.44	61.14	79.91	103.76	139.35	174.60	193.66
	Stdev	3.08	5.18	5.83	11.61	13.89	23.56	27.10	37.86	53.76	64.40	72.27
	Min	10.83	13.29	16.17	19.70	22.02	25.65	30.00	35.09	39.42	46.47	59.89
	Max	26.49	39.65	42.38	73.79	68.14	126.82	117.08	163.35	230.51	255.93	291.70
BNP-POLY	Mean	87.19	18.76	97.20	28.25	38.46	100.95	205.02	148.89	661.64	908.36	1020.85
	Stdev	350.96	1.91	350.66	3.94	27.81	188.11	491.05	201.06	767.87	852.81	752.28
	Min	11.04	14.57	19.19	22.15	24.80	25.13	26.05	32.74	35.45	38.06	133.57
	Max	1805.96	22.00	1815.00	35.22	171.97	928.79	1897.63	847.83	1892.30	2474.09	2765.64

Table A.11: Position-indexed formulation experiments for $|P| = 700$, $|A| = 7$

Method		$K = 2$	$K = 3$	$K = 4$	$K = 5$	$K = 6$	$K = 7$	$K = 8$	$K = 9$	$K = 10$	$K = 11$	$K = 12$
PICEF	Mean	9.81	16.64	27.03	45.15	85.77	132.08	221.64	476.49	746.67	1060.93	1391.81
	Stdev	3.96	8.86	14.48	15.45	36.51	54.10	143.49	473.81	674.82	624.99	953.63
	Min	6.10	10.23	16.92	28.10	46.56	64.96	83.36	123.93	153.49	229.89	310.20
	Max	22.29	48.46	79.24	96.61	170.04	236.84	752.40	2045.09	2417.46	2157.87	3600.01
BNP-PICEF	Mean	1597.19	1239.85	1970.94	2557.71	2661.05	3094.01	3426.76	3436.00	3551.63	3566.88	3600.04
	Stdev	1054.23	1099.56	1112.42	1099.13	1262.80	738.47	469.54	444.97	237.36	162.52	0.01
	Min	10.47	18.77	41.81	114.79	183.42	1959.23	2143.36	2172.23	2388.82	2770.72	3600.01
	Max	3601.19	3601.02	3600.93	3601.05	3600.62	3600.34	3600.21	3600.14	3600.10	3600.10	3600.06
HPIEF	Mean	47.46	60.30	71.00	93.18	141.93	194.25	301.22	851.62	874.49	1214.84	1688.33
	Stdev	20.40	31.39	35.59	51.30	125.99	128.11	286.29	784.70	735.08	978.75	1196.99
	Min	24.81	28.72	34.29	42.88	52.14	77.45	86.79	89.34	101.09	175.17	149.92
	Max	89.99	130.90	142.10	189.76	649.25	583.77	1232.76	2557.22	3057.21	3600.02	3600.02
BNP-POLY	Mean	1521.97	1159.34	1734.26	2239.58	2454.59	2979.44	3269.40	3274.71	3518.71	3459.88	3420.45
	Stdev	976.24	1121.69	1074.63	1364.66	1328.86	834.51	809.68	656.30	316.35	475.76	496.20
	Min	6.95	7.16	8.84	11.51	12.04	1807.77	323.14	1812.67	2037.16	1826.98	1819.21
	Max	3600.07	3600.06	3600.08	3600.23	3600.37	3600.37	3600.38	3600.46	3600.32	3600.55	3600.81

Table A.12: Position-indexed formulation experiments for $|P| = 700$, $|A| = 14$

Method		$K = 2$	$K = 3$	$K = 4$	$K = 5$	$K = 6$	$K = 7$	$K = 8$	$K = 9$	$K = 10$	$K = 11$	$K = 12$
PICEF	Mean	14.57	23.52	38.46	65.49	143.29	340.24	694.99	1085.51	1036.88	1016.48	1027.84
	Stdev	5.69	9.43	15.64	28.21	119.45	267.07	540.55	756.88	521.12	390.71	412.54
	Min	7.45	12.76	20.75	28.59	46.11	53.92	65.29	170.95	354.81	290.44	385.99
	Max	30.89	55.16	82.82	128.23	656.89	1078.52	2192.32	2548.76	2269.19	1769.58	1887.18
BNP-PICEF	Mean	1264.28	1612.64	2407.02	2921.08	3412.63	3390.81	3346.30	3378.98	3478.75	3586.62	3600.02
	Stdev	1295.03	1272.61	1189.48	960.60	509.38	436.66	396.00	313.42	305.45	58.48	0.01
	Min	9.71	18.89	76.04	208.48	1957.17	2259.70	2220.26	2502.96	2166.24	3302.37	3600.01
	Max	3600.80	3600.83	3600.68	3600.62	3600.37	3600.24	3600.15	3600.08	3600.07	3600.05	3600.05
HPIEF	Mean	39.05	51.21	62.70	93.00	240.20	564.15	1038.49	1413.02	1444.70	1473.19	1283.89
	Stdev	9.05	23.86	24.46	43.04	249.84	484.44	719.21	952.48	816.46	716.48	527.02
	Min	30.31	32.43	40.51	45.95	62.73	102.38	105.83	232.39	281.55	335.61	516.76
	Max	63.12	134.94	129.85	208.47	1241.43	1786.57	2751.71	3600.02	2877.35	2757.60	2371.12
BNP-POLY	Mean	1017.01	1519.12	2381.49	2884.76	3315.11	3266.22	3348.78	3119.47	3418.67	3600.08	3600.08
	Stdev	1250.38	1406.13	977.20	1011.44	653.27	670.13	578.91	723.61	447.02	0.05	0.04
	Min	6.83	8.59	22.14	28.53	1814.72	1822.70	1894.07	1842.47	2084.25	3600.02	3600.02
	Max	3600.06	3600.13	3600.21	3600.34	3600.43	3600.38	3600.39	3600.30	3600.26	3600.21	3600.17

Table A.13: Position-indexed formulation experiments for $|P| = 700$, $|A| = 35$

Method		$K = 2$	$K = 3$	$K = 4$	$K = 5$	$K = 6$	$K = 7$	$K = 8$	$K = 9$	$K = 10$	$K = 11$	$K = 12$
PICEF	Mean	13.13	23.38	55.18	99.27	203.74	317.91	547.05	797.67	1116.34	1293.25	1282.57
	Stdev	7.45	11.99	34.73	34.95	49.29	70.34	181.52	242.31	516.93	432.45	501.84
	Min	8.25	14.21	24.31	42.27	72.62	113.05	131.92	179.43	239.43	264.00	315.97
	Max	47.76	75.74	194.49	208.56	276.76	462.41	940.13	1179.01	2736.66	1896.02	2673.03
BNP-PICEF	Mean	1020.98	969.54	1190.59	833.25	1222.95	1820.27	2533.52	3101.86	3430.50	3472.62	3294.55
	Stdev	1249.82	1137.59	975.36	712.50	476.76	701.44	481.35	412.11	251.68	241.88	380.11
	Min	13.18	25.06	98.13	260.07	587.42	815.88	1277.61	2278.87	2662.25	2886.00	2357.03
	Max	3600.62	3600.69	3600.35	2321.96	2240.81	3600.02	3600.02	3600.03	3600.07	3600.04	3600.05
HPIEF	Mean	34.37	48.26	110.16	186.29	305.20	507.13	826.60	1171.59	1566.82	1763.52	1506.63
	Stdev	3.00	11.15	77.52	45.19	60.17	108.26	238.30	248.72	313.97	509.08	605.57
	Min	29.09	36.88	54.65	95.82	220.89	352.13	449.15	825.67	1012.74	1000.12	502.45
	Max	39.79	87.52	420.49	353.66	499.14	770.13	1652.82	1810.44	2481.68	2796.51	2489.86
BNP-POLY	Mean	946.39	968.40	1175.60	489.31	408.89	987.92	1204.20	1807.54	1940.38	2415.57	3181.40
	Stdev	1147.60	1134.31	1114.13	758.29	656.11	843.59	947.76	931.08	1101.74	903.48	679.19
	Min	10.57	14.50	28.51	33.90	37.49	43.77	86.42	101.46	139.30	244.96	1887.27
	Max	3600.14	3600.09	3600.35	1894.51	1866.21	1977.77	3600.26	3600.20	3600.18	3600.13	3600.15

Table A.14: Position-indexed formulation experiments for $|P| = 700$, $|A| = 175$

Method		$K = 2$	$K = 3$	$K = 4$	$K = 5$	$K = 6$	$K = 7$	$K = 8$	$K = 9$	$K = 10$	$K = 11$	$K = 12$
PICEF	Mean	29.54	43.30	71.41	105.27	151.55	203.83	286.35	374.32	467.57	560.50	681.95
	Stdev	13.36	16.27	29.08	44.51	66.12	87.99	125.80	166.32	215.48	271.71	370.30
	Min	14.44	19.95	28.55	33.10	44.67	58.77	82.04	104.68	119.57	156.31	137.36
	Max	63.37	72.10	112.54	161.79	261.19	336.25	493.77	644.54	807.41	990.81	1278.76
BNP-PICEF	Mean	18.37	25.95	58.41	101.92	133.68	200.45	298.62	396.98	625.76	963.66	1079.16
	Stdev	2.09	6.89	19.69	25.51	28.12	46.29	57.63	99.96	274.92	353.71	359.13
	Min	14.32	17.54	25.16	43.12	69.15	121.50	199.20	255.27	360.26	513.35	569.38
	Max	23.72	38.25	100.01	146.01	205.94	278.30	477.24	656.46	1539.05	2064.30	2096.84
HPIEF	Mean	39.67	55.28	93.24	137.82	200.39	277.33	378.82	533.55	642.43	874.87	908.95
	Stdev	5.51	13.00	34.40	43.98	73.27	105.09	146.86	233.83	266.91	385.84	363.31
	Min	31.07	37.56	45.29	55.53	69.20	84.06	103.58	114.62	130.30	165.84	169.59
	Max	57.05	94.99	182.11	211.11	331.98	512.84	602.28	870.66	1023.77	1297.35	1350.16
BNP-POLY	Mean	50.03	77.46	96.51	102.51	146.29	294.77	712.95	936.95	1281.41	1608.44	2130.64
	Stdev	6.49	12.00	13.69	16.35	95.95	464.74	769.68	807.90	871.01	1016.13	1106.75
	Min	36.48	55.54	67.21	80.60	92.13	102.89	126.45	127.95	153.58	293.59	257.74
	Max	61.19	100.42	124.90	130.75	603.92	1880.82	2189.93	2001.10	3600.04	3600.05	3600.04

B

Additional theoretical results for a new kidney exchange model

In this appendix, we provide the full proof of Theorem 12 from Chapter 4. Recall the (k,t)-REPRESENTATION WITH IGNORED EDGES: given an input of a directed graph $G = (V, E)$, a subset F of E, and integers $k \geq 1$ and $t \geq 0$, this problem asks whether there exist bit vectors $\mathbf{d}_i$ and $\mathbf{p}_i$ of length k for each $i \in V$ such that the $\{i, j\} \in F$ if and only if $\langle \mathbf{d}_i, \mathbf{p}_j \rangle \leq t$.

Consider the gadget G_k defined as follows on a graph on $\binom{k}{2} + k$ vertices. Let G_k^1 be the graph defined in Theorem 11 on $\binom{k}{2}$ vertices, i.e., the complement of a directed cycle on this many vertices. Associate with each vertex $u \in G_k^1$ a unique element from $\binom{[k]}{2}$ (all subsets of $[k]$ of size 2). Let G_k^2 be an independent set of k vertices. For each vertex $i \in G_k^2$, $i \in [k]$, add an incoming edge into i from $u \in G_k^2$ if and only if $i \in S_u$. Figure B.1 shows G_4.

Denote the *donor neighborhood* of $i \in V$ by $N_d(i) = \{j \in V : (i, j) \in E, i \neq j\}$, i.e., the set of patients compatible with donor i. Similarly, the *patient neighborhood* of $j \in V$ is $N_p(j) = \{i \in V : \{i, j\} \in E, i \neq j\}$.

Lemma 12. *There is a unique (up to permutations)* $(k, 1)$*-representation of* G_k.

Proof. First consider G_k^1. For all $u \in V(G_k^1)$, since $\{u, u-1\} \notin E(G_k^1)$, and the compatibility function is f_{thresh}^1, there exist two distinct conflict bits q_1^u and q_2^u in $Q_d(u) \cap Q_p(u-1)$. Moreover, for any u, v distinct, $\{q_1^u, q_2^u\} \neq \{q_1^v, q_2^v\}$. Otherwise, $\{q_1^u, q_2^u\} \subseteq Q_p(v-1)$ and $\{q_1^v, q_2^v\} \subseteq Q_p(u-1)$, but at least one of the edges $\{u, v-1\}$ or $\{v, u-1\}$ exists in G_k^1.

In addition, $|Q_d(u)| = 2$ for all $u \in V(G_k^1)$. Suppose not, and there exists a third distinct (from q_1^u and q_2^u) conflict bit q_3^u in $Q_d(u)$. As the number of vertices is $\binom{k}{2}$, there exists a vertex v_1 with $\{q_1^{v_1}, q_2^{v_1}\} = \{q_1^u, q_3^u\}$, and a (different) vertex v_2 with $\{q_1^{v_2}, q_2^{v_2}\} = \{q_2^u, q_3^u\}$. Then $\{u, v_1 - 1\}$ and $\{u, v_2 - 1\}$ are both not in $E(G_k^1)$. However, u has edges to all vertices except itself and $u-1$, which is a contradiction, as u, v_1, and v_2 are all distinct. From this, it also follows that $|Q_p(u)| = 2$.

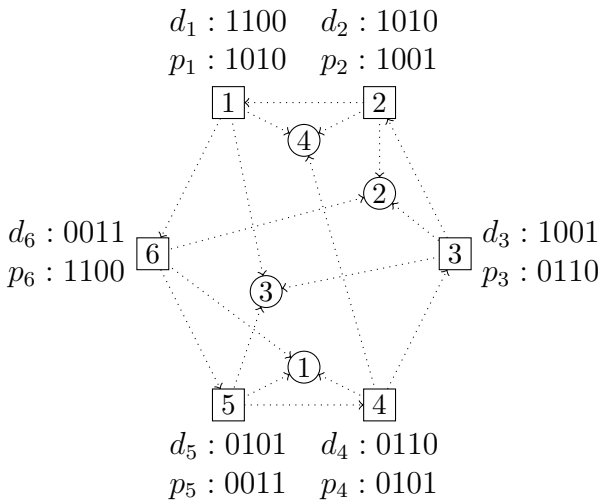


Figure B.1: Gadget G_4 with a subset of *non*-edges shown; all edges between circle vertices (those in G_4^2) are also not in E.

We have thus shown that every vertex $u \in G_k^1$ has exactly two bits set to one in its donor attribute vector, with a unique pair of bits per vertex, and $Q_d(u) = Q_p(u-1)$. However, without more structure, it is not possible to tell in which donor vectors a particular conflict bit appears. The additional graph G_k^2 allows us to identify this, up to permutations.

Since there are no outgoing edges from any of the vertices in G_k^2, and every pair of bits in $\binom{[k]}{2}$ appears in exactly one patient vector of a vertex in G_k^1, each donor vector in G_k^2 must be the all-ones vector of length k.

Consider vertex $i \in [k]$ in G_k^2. It has an incoming edge from each vertex $u \in V(G_k^1)$ such that $i \in S_u$ and it is missing the $\binom{k-1}{2}$ other possible incoming edges from G_k^1 (note that the labeling of the vertices, as well as the choices of the sets S_u, are made without any knowledge of the bit-vectors associated with the vertices). We next show that $\left|\cap_{u \in N_p(i)} Q_d(u)\right| = 1$. That this quantity is at most 1 is clear, as $Q_d(u)$ and $Q_d(v)$ intersect in at most one conflict bit for all $u, v \in V(G_k^1)$, $u \neq v$. If this quantity were 0, then for some $u, v \in N_p(i)$, $Q_d(u) \cap Q_d(v) = \emptyset$. But then at least two zeroes would appear in $Q_p(i)$, which is a contradiction as it implies that i would have more than k incoming edges. Thus, the patient vector $\mathbf{p}_i$ for $i \in V(G_k^2)$ has exactly one zero and ones elsewhere. Moreover, since $N_p(i) \neq N_p(j)$ for any distinct $i, j \in [k]$, it follows that $\mathbf{p}_i \neq \mathbf{p}_j$, so each patient vector is distinct and the position of its only zero is unique. □

Lemma 13. *Consider a digraph G having G_k as a subgraph and an additional vertex $x \notin V(G_k)$. We use the compatibility function $f^1_{\textsf{thresh}}$ and seek to find a $(k,1)$-representation for the induced subgraph $G[V(G_k) \cup \{x\}]$. Let $U \subseteq V(G_k^1)$ having that property that if $v \in V(G_k^1)$ with $Q_d(v) \subseteq \cup_{u \in U} Q_d(u)$, then $v \in U$. Let $U' = \{u \in V(G_k^1) : u + 1 \in U\}$. Let $Q = \cup_{u \in U} Q_d(u)$.*

If $N_p(x) = V(G_1^k) \setminus U$, then $Q_p(x) = Q$. If $N_d(x) = V(G_1^k) \setminus U'$, then $Q_d(x) = Q$.

Proof. We use the fact that there are exactly two bits set to one in the donor and patient vectors of each vertex in G_k in any $(k,1)$-representation. For the first statement, since x has no edge from

$u \in U$, $Q_p(x) \supseteq Q_d(u)$. Thus $Q_p(x) \supseteq Q$. Now let $v \in V(G_k^1) \setminus U$ and $q_v \in Q_d(v) \setminus Q$. If $q_v \in Q_p(x)$, then for each $q \in Q$, there exists a vertex w in G_k^1 with $Q_d(w) = \{q, q_v\}$, so that $\{w, x\}$ would also not be an edge of G, a contradiction. Hence, $Q_p(x) = Q$. The second statement follows analogously. □

Theorem 12. *The* (k,t)-REPRESENTATION WITH IGNORED EDGES *problem is* $\mathcal{NP}$*-complete.*

Proof. Consider a 3SAT formula on n variables and with m clauses. Set $k = 2n+2$, and build the following graph on $2 + n + m + \binom{k}{2} + k$ vertices. The first two vertices are labeled v and u. Then there is a vertex v_i for each variable $i \in [n]$, a vertex c for each clause $c \in [m]$. Call the subgraph induced by these $2 + n + m$ vertices G'. The last vertices come from the gadget G_k.

The vertices in G_k^2 ground the k bits used in each donor and patient vector.. We think of the k bits, in order, as corresponding to the n positive literals, then their n negations, followed by two "extra" bits. Then the index of literal x_i will be i, and the index of literal $\bar{x}_i$ will be $n + i$. For i and j distinct in $V(G_k^2)$, $|N_p(i) \cap N_p(j)| = 1$ within G_k. Denote this vertex of G_k^1 by $v(i,j)$, and without loss of generality we can assume that $Q_d(v(i,j)) = \{i, j\}$.

The edges among vertices in the induced subgraph G_k are already defined; we define (a subset) of the rest of the edges. Together, these comprise precisely the subset F of the edges and non-edges specified as an input the instance we are creating of (k,t)-REPRESENTATION WITH IGNORED EDGES.

Vertex v has no incoming edges, and the only outgoing edges from v to $V(G')$ are to every variable vertex v_i, $i \in [n]$. The rest of the vertices that are not in G_k have no outgoing edges at all, to either $V(G')$ or $V(G_k)$, and the only incoming edges are from vertices of G_k^1. Vertex u has an incoming edge from every vertex of G_k^1 except $v(2n+1, 2n+2)$. For each variable vertex v_i, $i \in [n]$, it has an incoming edge from every vertex in $V(G_k^1)$ except $v(i, n+i)$. For each clause $c \in [m]$, let $\{c_1, c_2, c_3\}$ be the indices of the three literals that appear in c. Let $C \subset V(G_k^1)$ be $\{v(c_1,c_2), v(c_1,c_3), v(c_2,c_3), v(c_1,k), v(c_2,k), v(c_3,k)\}$. Then the vertex corresponding to c has an incoming edge from every vertex in $V(G_k^1) \setminus C$.

Every vertex of $V(G')$ except for v will have a donor vector with every bit set to one because there are no outgoing edges to any vertex of G_k^1, and v will have an all-ones patient vector because it has no incoming edges from G_k^1. By Lemma 13, in any $(k,1)$-representation of G, vertex u will have $Q_p(u) = \{2n+1, 2n+2\}$. Variable vertex v_i, $i \in [n]$, will have $Q_p(v_i) = \{i, n+i\}$. Clause vertex $c \in [m]$ will have $Q_p(c) = \{c_1, c_2, c_3, 2n+2\}$.

Since the graph does not have an edge from v to u, $\{2n+1, 2n+2\} \subseteq Q_d(v)$ (these are the only two conflict bits in $Q_p(u)$ and the threshold is 1). Since the graph has an edge from v to each variable vertex v_i, $i \in [n]$, $Q_d(v)$ must contain at most one of the indices corresponding to the variable or its negation (there are no conflicts from the extra bits, which are set to 0 in the patient vector of v_i). Since the graph does not have an edge from v to any of the clause vertices, it has to have at least one conflict bit in a position corresponding to one of the three literals in the clause (the other conflict comes from the extra bit $2n+2$).

Thus, finding a suitable $(k,1)$-representation that satisfies the adjacencies of edges that appear in F would involve finding an appropriate set $Q_d(v)$, which we have shown corresponds to choosing at most one value for each x_i, as well as choosing at least one literal that appears in each clause.

This is the same as the problem of finding a satisfying formula for the initial instance of 3SAT.

As an example, consider the 3SAT formula $x_1 \vee \bar{x}_2 \vee x_3$. Figure B.2 shows the most relevant part of the graph used in the reduction. One possible $(k, 1)$-representation may have $Q_d(v) = \{1, 7, 8\}$, indicating $x_1 = 1$ and the rest of the variables are arbitrary. Another example of a possible representation is $Q_d(v) = \{1, 3, 5, 7, 8\}$, meaning $x_1 = 1$, $x_2 = 0$ (index 5 appears), and $x_3 = 1$. □

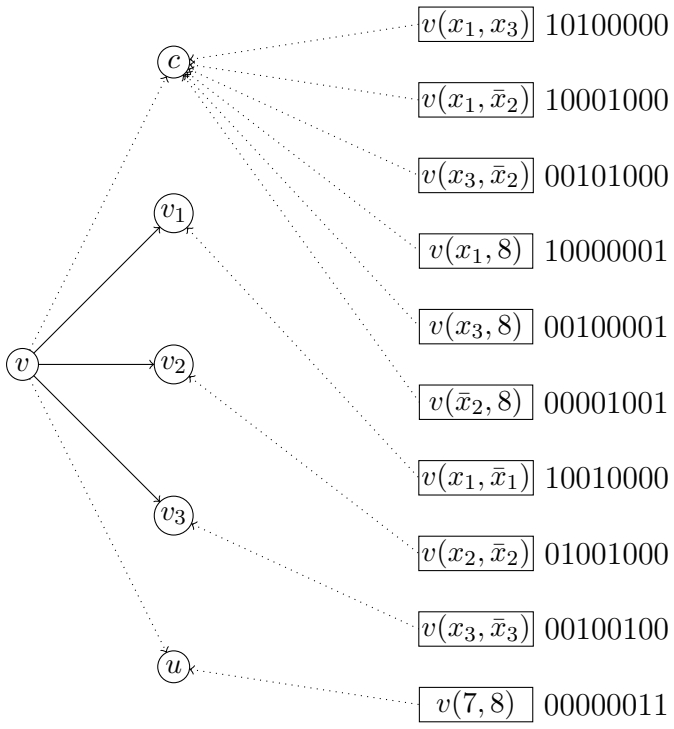


Figure B.2: Example of 3SAT reduction to (k, t)-representation.

C

Additional experimental results for pre-match edge testing

C.1 Additional experimental results on UNOS graphs

In this appendix, we include additional experimental results on the same 169 compatibility graphs drawn from the real UNOS kidney exchange used in the adaptive and non-adaptive edge testing experiments of Section 5.10. These experiments mimic those of Section 5.10.2, only this time including in the analysis empty omniscient matchings. If an omniscient matching is empty, then our algorithm will achieve at most zero matches as well. In Section 5.10, we removed these cases from the experimental analysis because achieving zero matches (using any method) out of zero possible matches trivially achieves 100% of the omniscient matching; by not including those cases, we provided a more conservative experimental analysis. In this section, we include those cases and rerun the analysis.

Figure C.1 mimics Figure 5.15 from Section 5.10 in the body of this thesis. It shows results for 2-cycle matching on the UNOS compatibility graphs, without chains (left) and with chains (right), for $R \in \{0, 1, \ldots, 5\}$ and varying levels of $f \in \{0, 0.1, \ldots, 0.9\}$. We witness a marked increase in the fraction of omniscient matching achieved as f gets close to 0.9; this is due to the relatively sparse UNOS graphs admitting no matchings for high failure rates.

Figure C.2 shows the same experiments as Figure C.1, only this time allowing both 2- and 3-cycles, without (left) and with (right) chains. It corresponds to Figure 5.16 in Section 5.10, and exhibits similar but weaker behavior to Figure C.1 for high failure rates. This demonstrates the power of including 3-cycles in the matching algorithm—we see that far fewer compatibility graphs admit no matchings under this less-restrictive matching policy.

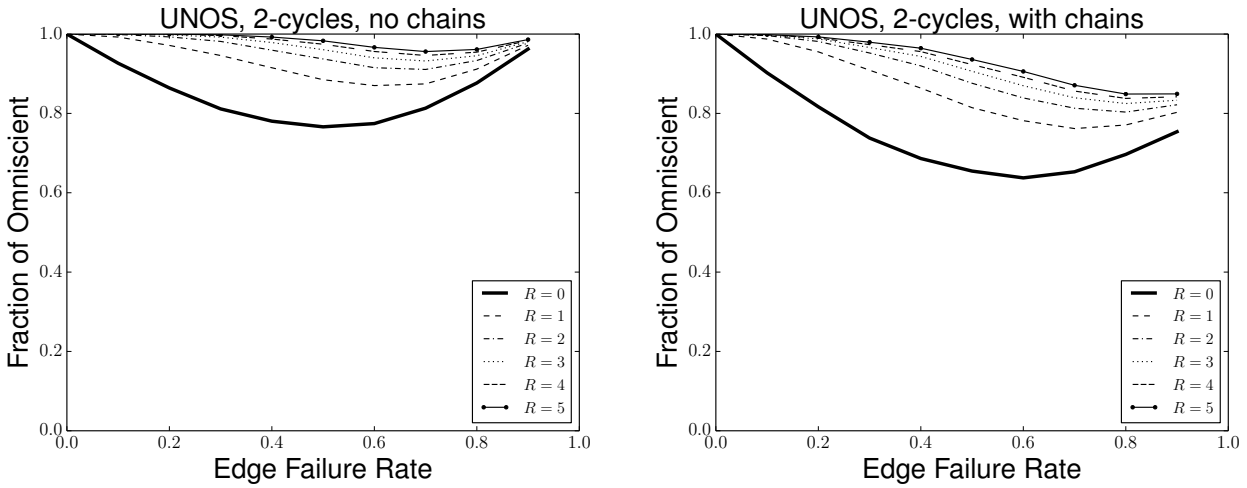


Figure C.1: Real UNOS match runs, restricted matching of 2-cycles only, without chains (left) and with chains (right), including zero-sized omniscient matchings.

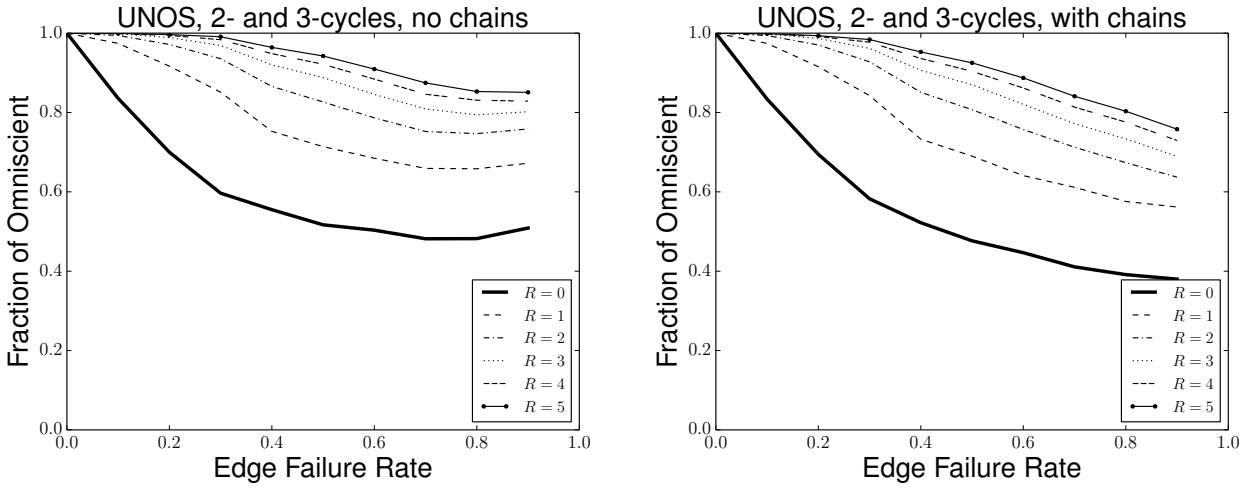


Figure C.2: Real UNOS match runs, matching with 2- and 3-cycles, without chains (left) and with chains (right), including zero-sized 0 matchings.

C.2 Myopic incremental edge testing

In this section, we present a basic algorithm for incrementally testing edges in a non-batch setting; that is, we work under the assumption that testing can be performed and that information digested on an edge-by-edge basis. We compare against the batch adaptive edge testing algorithm from Section 5.10. By allowing the algorithm to receive and react to new information at a faster pace, this comparison allows us to measure, in some sense, the "cost of batching" in our initial adaptive algorithm. Indeed, while it may not be the case that immediate feedback without batching is realistic in the kidney exchange context, research in this direction may allow for even sparser

batches of edge testing in the context of the original batch algorithms.

Algorithm 5 implements a basic "expected improvement"-style algorithm (see, e.g., work by Jones et al. [122] for foundational work on general expected improvement algorithms) for adaptive edge testing of barter exchange graphs. It is initialized with a directed graph with edges that may or may not exist; it then tests edges one by one until an improvement-based stopping condition is met, or it runs out of edges to test. Iteratively, it myopically tests for existence the edge that improves the expected utility of the current optimal failure-aware matching—taking all previous edge tests into account—the most. This is determined by iterating over every edge e, with probability of existence p_e, calculating the expected utility of matching with that edge existing (denoted o_e^+) and without it existing (denoted o_e^-), and finding the edge e^* with highest weighted average $o^* = p_{e^*} o_{e^*}^+ + (1 - p_{e^*}) o_{e^*}^-$. That edge is then tested for existence and then set to exist or not exist (depending on the test) for all remaining iterations.

Code for Algorithm 5, as well as for the succeeding experiments, can be found at `https://github.com/JohnDickerson/KidneyExchange` in the `kpd.rematch` package.

We now perform an experimental comparison of Algorithm 5 and the batch adaptive algorithm of Section 5.10. We operate in essentially the same environment as the experiments of Sections 5.10.2 and C.1—drawing generated graphs that mimic the UNOS exchange by way of seeding a generator with the first 314 real UNOS match runs. The experiments are performed as follows. Draw a graph from the generator; in our experiments, we draw graphs such that $|V| = 100$. Determine probabilistically which edges exist and do not exist. First, determine the optimal matching given omniscience, that is, complete knowledge of edges' existence. Next, run the adaptive batch algorithm for 5 batch matches (as in the experiments in of Sections 5.10.2 and C.1), recording the match utilities realized for each number of batches and the number of edges tested. Then, run Algorithm 5—which tests edges one by one—and record the match utility at every step. Under all algorithms, simulated edge failures are exactly the same; that is, the underlying graph and its dynamics are identical.

Figure C.3 shows experimental results on the same set of generated graphs both allowing chains and not allowing chains. (In both sets of experiments, we allow 2- and 3-cycles, as is the case in most experiments in this thesis, and in most kidney exchange clearinghouses in practice.) The y-axis shows, for each number of rematches $R \in \{1, 2, 3, 4, 5\}$, the number of edges used by Algorithm 5 to achieve at least the utility of the batch algorithm, as a fraction of the number of edges used by the batch adaptive algorithm. For example, if the batch algorithm for $R = 2$ used 20 edges to achieve objective $o_B^{R=2}$, and Algorithm 5 achieved objective $o_A \geq o_B^{R=2}$ using just 10 edges, then this would appear on Figure C.3 as a point $10/20 = 0.5$ for the $R = 2$ line, for a given underlying failure rate. We vary on the x-axis uniform failure rates in $\{0.0, 0.1, \ldots, 1.0\}$ as in previous experiments. Thus, "lower is better" in Figure C.3 in the sense that lower values show Algorithm 5 performing substantially fewer edge tests to achieve at least the same utility as the batch adaptive algorithm.

Figure C.3 shows that Algorithm 5 results in the greatest gains when the failure rates for edges are very low or very high. This is intuitive: when failure rates are low, most edges exist, and the batch adaptive algorithm will be too conservative, needlessly testing many edges that will likely exist in reality. When the failure rate is very high, the amount of information gained from testing a

Algorithm 5 Myopic edge test policy

function DOMYOPICMATCHING($G = (V, E)$)
    $E_+ = \emptyset$; $E_- = \emptyset$
    while $E_+ \cup E_- \neq E$ **do**
        $o^* = \text{MATCH}(G, E_+, E_-)$
        $(o, e) = \text{GETBESTEDGE}(G, E_+, E_-)$
        if $o > o^* \wedge e \neq \emptyset$ **then**
            if EDGEEXISTS(e) **then**
                $E_+ = E_+ \cup \{e\}$
            else
                $E_- = E_- \cup \{e\}$
        else
            break
    return (E_+, E_-)

function GETBESTEDGE($G = (V, E), E_+, E_-$)
    $e^* = \emptyset$; $o^* = 0$
    for each $e \in E \setminus (E_+ \cup E_-)$ **do**
        $o_e^+ = \text{MATCH}(G, E_+ \cup \{e\}, E_-)$
        $o_e^- = \text{MATCH}(G, E_+, E_- \cup \{e\})$
        $o_e = p_e o_e^+ + (1 - p_e) o_e^-$
        if $o_e > o^*$ **then**
            $o^* = o_e$; $e^* = e$
    return (o^*, e^*)

function MATCH($G = (V, E), E_+, E_-$)
    Performs optimal matching using the failure-aware algorithm from Section 5.5, with failure probabilities for edges in E_+ set to 0.0 and all edges in E_- removed (or set to fail with probability 1.0), and all other as-of-now untested edges existing probabilistically.
    return the utility of the matching minus $\sum_{e \in E_+ \cup E_-} r_e$, the amount we have spent testing edges so far. (For our experiments, we ignore edge testing costs.)

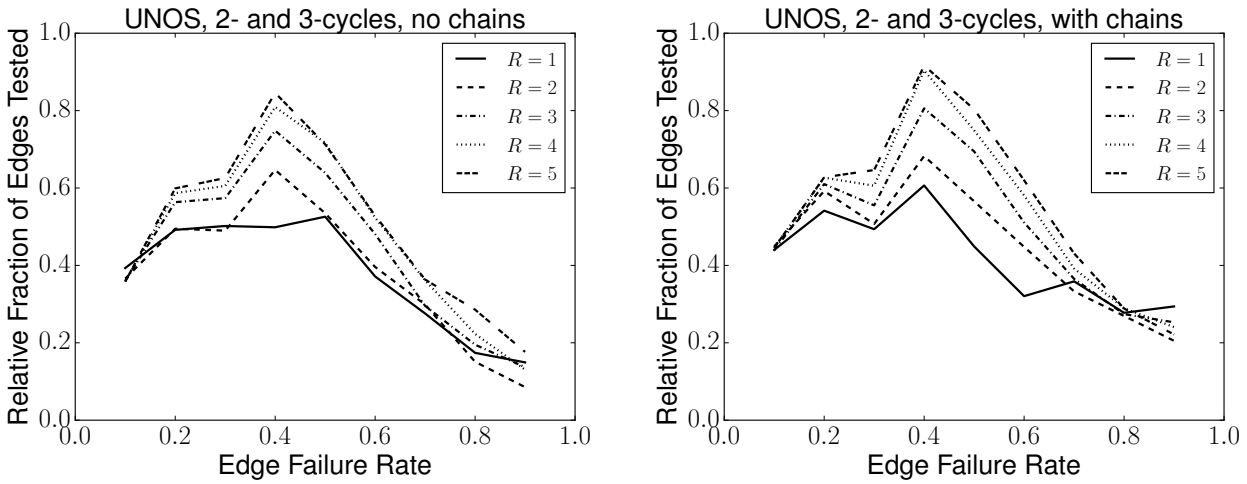


Figure C.3: Real UNOS match runs, matching with 2- and 3-cycles, without chains (left) and with chains (right), not including zero-sized 0 matchings.

single edge and finding out that it exists is extremely high, and can be used to immediately guide the search toward other parts of a potential matching.

For realistic edge failure rates around 0.5, Algorithm 5 still provides a large gain (saving roughly 50% of edge tests) for $R = 1$; however, as we increase the number of rematches R, that gain tapers off. This is due in part to the number of edges tested by the batch matching algorithm in each round *also* tapering off as the number of rounds increases. As that algorithm tests and finds that parts of a matching exist, it will potentially include those tested edges (and cycles and chains) in future batch matchings; we do not count edge tests in a batch matching after the initial test, so the batch matching algorithm in some sense approaches the incremental edge testing strategy of Algorithm 5 as the number of batch rematches increases.

D

Additional liver and multi-organ exchange methodology and experimental results

D.1 A parameterized, realistic compatibility graph generator

In this section, we provide a more in-depth enumeration of the steps taken to generate realistic liver or multi-organ exchange compatibility graphs; a shorter explanation was given in Section 9.3.1. Section D.1.1 describes the process of drawing data from reliable sources (here, specific to the US), while Section D.1.2 shows how we feed this generated data into a graph creation algorithm that probabilistically determines the existence of compatible and incompatible candidate-donor pairs, as well as compatibility constraints between different candidate-donor pairs. As noted in Chapter 9, in the large and with high probability, graphs generated by this algorithm will mimic the demographics that would prevail in a large-scale fielded exchange in the US. (Plugging different raw data (e.g., gender, age, weight, blood type distributions) into the generator algorithm would provide realistic generation of non-US compatibility graphs.) Our generator is a generalization of (i.e., more powerful than) the current standard generator proposed by Saidman et al. [191].

D.1.1 Sampling from real-world data

Current medical knowledge is incapable of exactly predicting the compatibility of a particular donor and candidate. However, many attributes are known that can guide doctors—and algorithms—toward a realistic quantification of the chance of organ rejection. In this section, we describe these factors and the open source data sets that our algorithm uses to realistically sample the US population. In the discussions ahead, we use "OPTN" to refer to the data available from the Organ

Procurement and Transplantation Network.[1] All OPTN data is current as of November 11, 2011.

Gender

While a donor of one gender can donate an organ to a candidate of another gender, we must take gender into account during graph generation. This is because other traits that affect the probability of a transplant's success (e.g., weight or age) depend on a person's gender. We draw candidate genders from the OPTN data set, and donor genders from the greater US population through the 2010 US Census report.[2] Table D.1 shows the distributions of liver-needing candidates and the natural US population as donors. Men are very over-represented in the candidate pool. (Note that similar distributions can be obtained for kidney-needing candidates, and used in a multi-organ generator.)

	Male	**Female**
Candidate	61.71	38.29
Donor	48.53	51.47

Table D.1: Distribution of (liver) candidate and donor genders, drawn from OPTN and 2010 US Census data, respectively.

Blood type

A candidate and donor must be ABO blood type compatible (e.g., an A-type donor is compatible with A- and AB-type candidates), although blood type suppression through drugs is a recent advance that has the potential to remove this constraint [204]. We draw candidate blood types from the OPTN distribution (dependent on gender), and donor blood types from the overall US.[3] The OPTN distribution is roughly equal across genders, and both distributions are roughly equal to each other. Nevertheless, it is important to have this parameterized capability in the generator in the event that, for instance, some "harder" blood type (e.g., AB) gets over-represented in the candidate pool. Table D.2 shows the exact distribution and the ABO-compatibility matrix, with percentages shown for liver-needing candidates.

Age

Age plays a role in transplantation, but we were unable to find any specific quantification of the amount by which increased donor or candidate age (or, in the case of children, decreased candidate age) affects this success rate. Even without this information, age is important to model because it will allow us to generate a realistic distribution of candidate and donor *weights*, a trait whose

[1] http://optn.transplant.hrsa.gov/data/
[2] http://www.census.gov/compendia/statab/cats/population.html
[3] http://bloodcenter.stanford.edu/about_blood/blood_types.html

Donor	**Candidate**			
ABO	O	A	B	AB
O	✓	✓	✓	✓
A	✗	✓	✗	✓
B	✗	✗	✓	✓
AB	✗	✗	✗	✓

	Male		**Female**	
ABO	Cand.	Donor	Cand.	Donor
O	47.83	44	48.91	44
A	38.39	42	37.08	42
B	11.37	10	11.41	10
AB	2.40	4	2.58	4

Table D.2: Left: ABO blood type compatibility matrix. Marks indicate a donor (row) as ABO-compatible with a candidate (column). Right: ABO percentages for candidates and donors.

effect is easily quantified. We sample ages (dependent on gender) for candidates from the OPTN pool and for the donors from the 2010 US Census at a granularity level of one year. To save space, Table D.3 does not separate the population into one-year segments as rows, while our generator does. In our generator we also take into account the constraint that organ donors must be 18 years old, and we normalize the distributions accordingly.

	Male		**Female**	
Age	Candidate	Donor	Candidate	Donor
< 1	0.259	–	0.465	–
1–5	0.837	–	1.220	–
5–10	0.568	–	1.075	–
11–17	0.717	–	1.444	–
18–34	4.193	31.883	5.554	29.357
35–49	14.851	27.798	14.976	26.617
50–64	64.851	25.066	57.079	25.053
≥ 65	13.725	15.252	18.186	18.972

Table D.3: Probability distribution of ages, respective of candidate and donor gender.

Weight

Unlike in kidney exchange, the physical weight of both the candidate and donor play an enormous role in the feasibility of liver transplantation.[4] Intuitively, the size of a liver is generally proportional to the size of the person who grew it. In live liver donation, the donor's liver is cut in two (one lobe is removed). For both donor and candidate to remain healthy, the slice of liver left in the donor must be large enough to maintain her life, and the slice of liver given to the candidate must be large enough to maintain his. Thus, a general rule of thumb that the donor must weigh as much as (or more than) the candidate is in place in live liver donation. We adopt that convention for liver exchange.

Given the age and gender (generated separately from OPTN data for candidate and US Census data for donors, as described earlier), we sample from a fine-grained table of weights recently released by the Center for Disease Control [155]. This data, given on a by-year basis until age 20 and in increments of 5 years thereafter, includes mean weights, sample errors, and sample sizes. From this, we calculate a standard deviation and sample from a normal distribution with this mean and standard deviation. While there are issues with this method—most notably that the candidate weights may be drawn from a different distribution than the general US public, and that human weights are not distributed normally but are skewed toward weighing more—we feel that this sampling approach provides a reasonable starting point for future generation techniques. The full table of weights is omitted due to space.

HLA antibodies and antigens

In kidney exchange, tissue type (HLA antibodies and antigens) are another very important determinant of compatibility. A candidate and donor sharing antigen encoding on the same locus possibly results in a positive *virtual crossmatch* across antigens. A positive virtual crossmatch means that the system can detect incompatibility. In kidney exchange graph generation, this is quantified by the probability that the candidate is not tissue-type compatible with a randomly drawn donor. This probability is called %PRA for panel reactivity antibody [191]. Furthermore, tissue type can change over time, resulting in the need for contingency plans after the time of algorithmic matching but before the surgery. For example, if the candidate comes down with a cold or flu days before surgery, the surgery may need to be rescheduled or permanently canceled.

In liver exchange, %PRA plays less of a role due to the use of suppressant drugs. As such, while the generator supports %PRA (and can use sampled data from the OPTN databases[5]), we exclude %PRA in our liver experiments. However, %PRA is included in our multi-organ experiments for kidney candidates.

[4]Large weight differences between donor and candidate can factor into kidney exchange as well, but this has not been taken into account in either the current state of the art generator or the weighting algorithms used in the fielded US-wide kidney exchange.

[5]The relationship (e.g., parent-child, spousal) between candidate and donor can yield information on HLA compatibility (e.g., due to inheritance of HLA from each parent or changes in HLA antibodies due to pregnancy; see Saidman et al. [191] for details), and is supported by the generator of Saidman et al. and our generator.

D.1.2 Generator algorithm

We now give the method for generating the compatibility graph from data sampled from the sources given in the previous section. Note that the probability distributions from the previous section (and the organs to which they pertain) can be swapped without affecting the correctness of the algorithm beyond the "is compatible" checks described below.

Algorithm 6 Compatibility graph generator

$G := (V = \emptyset, E = \emptyset)$
while $|V| < n$ **do**
    $c =$ candidate, $d =$ donor
    $c.drawOrganType()$
    $\{c, d\}.drawGender()$
    $\{c, d\}.drawBlood(gender)$
    $\{c, d\}.drawAge(gender)$
    $\{c, d\}.drawTissueType(gender)$
    $\{c, d\}.drawWeight(gender, age)$
    if $\neg isCompatible(c, d)$ **then**
        $V = V \cup \{v^{c,d}\}$
for $v_i, v_j \in V$ s.t. $V_i \neq V_j$ **do**
    if $isCompatible(v_j^c, v_i^d)$ and $x \in U[0, 1] > f$ **then**
        if $isWilling(v_i, v_j)$ **then**
            $E = E \cup \{(v_i, v_j)\}$
return directed compatibility graph G

Algorithm 6 gives a two-step process for generating a compatibility graph $G = (V, E)$, given a number n, such that $|V| = n$. First, sample from real-world data until n incompatible candidate-donor pairs are generated. When generating a liver exchange, one would set the algorithm to sample from the liver data given above; however, when generating a multi-organ exchange consisting of livers and kidneys, one would include the proper proportions of kidney and liver candidates and sample from the appropriate real-world data per organ. When we ran the liver and multi-organ experiments in Section 9.4, the kidney waitlist was 5.84 times longer than the liver waitlist, which was reflected in this algorithm. (When the experiments for Chapter 9 were run, the kidney waitlist was 6.50 times longer than the liver waitlist.)

If needed, the algorithm can easily be augmented to keep track of any compatible candidate-donor pairs generated. As is common practice in kidney exchange, these pairs are assumed to match on their own, and do not enter the pool. Recent kidney exchange research suggests that incentivizing even compatible pairs to join a nationwide exchange could result in better matchings [19, 178]. Other additions could be made to the algorithm as data becomes available (e.g., correlating donor and candidate characteristics under the assumption that a donor may likely come from the candidate's family).

After n incompatible candidate-donor pairs are generated, the algorithm steps through each pair v_i, v_j of candidate-donor pairs and, if the latter's candidate v_j^c is compatible with the former's

donor v_i^d, then a directed edge is added from v_i to v_j. Note the inclusion of an exogenous "incompatibility factor" $f \in [0, 1]$ that, if prescribed, randomly determines an edge failure even in the case of a compatibility success. This factor is common in the kidney literature [22], and is used to account for incompleteness of medical knowledge and, during simulation, temporal fluctuations in candidate-donor compatibility.

Algorithm 6 calls a function *isCompatible(c,d)*. In the liver case, this checks whether two patients are ABO-compatible and whether the donor's weight is greater than or equal to the candidate's weight. In the kidney case, this checks whether two patients are ABO-compatible and whether a virtual crossmatch based on tissue type returns negative. As better medical knowledge and data become available, this function can be generalized to take new compatibility aspects into account. The algorithm also calls a function *isWilling(v_i, v_j)*, which returns true if the donor at v_i is willing to give an organ of the type needed by the patient in v_j. This corresponds to, e.g., the probabilities $p_{K\to L}$ and $p_{L\to K}$ used in Chapter 9 theoretical and experimental sections.

D.2 Additional experimental results

In this section, we provide statistical significance testing for the dynamic bi-organ experiments of Section 9.4.3. The tables are organized as follows. Each table corresponds to a different distribution of compatibility graphs. Tables D.4, D.5, and D.6 give results for DENSE graphs with exogeneous incompatibility rates $f = 0.5$, $f = 0.7$, and $f = 0.9$, respectively. These tables support Figures 9.10 and 9.11 in the body of the paper. Table D.7 gives results for the UNOS family of graphs. This table supports Figure 9.12 in the body of the paper.

Each row in a table corresponds to a different value of $p_{K\to L}$; the value of $p_{K\to L}$ is specified in the first column of the table. From left to right, the columns represent: n, the number of independent runs used to support the results in this row; the average number of patients matched in total for an independent liver and independent kidney exchange; the standard deviation of the previous; the average number of patients matched in total for a combined bi-organ exchange; the standard deviation of the previous; the percentage gain in number of matched patients achieved by combining exchanges; t-statistic from an independent samples t-test; the associated two-tailed p-value for the previous; U-statistic from a Mann-Whitney U test (roughly, a non-parametric version of the independent samples t-test); and the associated one-tailed p-value for the previous.

$f = 0.5$		*Independent*		*Combined*			t-test		Mann-Whitney	
$p_{K \to L}$	n	Avg. #	Stdev	Avg. #	Stdev	% Gain	t	p	U	p
0.0	90	5143.7	(129.9)	–	–	–	–	–		
0.1	41	–	–	6059.4	(155.0)	17.80%	34.884	$\ll 0.001$	0.0	$\ll 0.001$
0.2	53	–	–	6109.3	(153.3)	18.77%	39.825	$\ll 0.001$	0.0	$\ll 0.001$
0.3	63	–	–	6110.4	(149.0)	18.79%	42.332	$\ll 0.001$	0.0	$\ll 0.001$
0.4	79	–	–	6102.4	(143.9)	18.64%	45.240	$\ll 0.001$	0.0	$\ll 0.001$
0.5	73	–	–	6137.5	(155.7)	19.32%	44.141	$\ll 0.001$	0.0	$\ll 0.001$
0.6	83	–	–	6114.5	(126.2)	18.87%	49.491	$\ll 0.001$	0.0	$\ll 0.001$
0.7	81	–	–	6156.4	(153.2)	19.69%	46.472	$\ll 0.001$	0.0	$\ll 0.001$
0.8	77	–	–	6140.8	(140.3)	19.38%	47.364	$\ll 0.001$	0.0	$\ll 0.001$
0.9	79	–	–	6182.7	(143.7)	20.20%	49.060	$\ll 0.001$	0.0	$\ll 0.001$
1.0	81	–	–	6135.3	(133.2)	19.28%	48.953	$\ll 0.001$	0.0	$\ll 0.001$

Table D.4: Statistical significance testing for DENSE distribution graphs with $f = 0.5$.

$f = 0.7$		*Independent*		*Combined*			t-test		Mann-Whitney	
$p_{K \to L}$	n	Avg. #	Stdev	Avg. #	Stdev	% Gain	t	p	U	p
0.0	99	4979.6	(127.6)	–	–	–	–	–		
0.1	90	–	–	5655.3	(146.6)	13.57%	33.685	$\ll 0.001$	0.0	$\ll 0.001$
0.2	69	–	–	5819.3	(116.5)	16.86%	43.215	$\ll 0.001$	0.0	$\ll 0.001$
0.3	50	–	–	5838.9	(134.3)	17.25%	37.874	$\ll 0.001$	0.0	$\ll 0.001$
0.4	35	–	–	5898.4	(143.0)	18.45%	35.188	$\ll 0.001$	0.0	$\ll 0.001$
0.5	42	–	–	5914.6	(134.5)	18.78%	38.872	$\ll 0.001$	0.0	$\ll 0.001$
0.6	32	–	–	5964.0	(119.9)	19.77%	38.196	$\ll 0.001$	0.0	$\ll 0.001$
0.7	33	–	–	6011.4	(175.6)	20.72%	36.094	$\ll 0.001$	0.0	$\ll 0.001$
0.8	36	–	–	6006.9	(134.4)	20.63%	40.476	$\ll 0.001$	0.0	$\ll 0.001$
0.9	40	–	–	6010.4	(152.2)	20.70%	40.419	$\ll 0.001$	0.0	$\ll 0.001$
1.0	36	–	–	6051.4	(156.8)	21.52%	40.192	$\ll 0.001$	0.0	$\ll 0.001$

Table D.5: Statistical significance testing for DENSE distribution graphs with $f = 0.7$.

$f = 0.9$		*Independent*		*Combined*			t-test		Mann-Whitney	
$p_{K \to L}$	n	Avg. #	Stdev	Avg. #	Stdev	% Gain	t	p	U	p
0.0	79	3708.4	(113.2)	–	–	–	–	–		
0.1	79	–	–	4089.6	(113.3)	10.28%	21.018	$\ll 0.001$	19.5	$\ll 0.001$
0.2	81	–	–	4298.0	(108.0)	15.90%	33.501	$\ll 0.001$	0.0	$\ll 0.001$
0.3	82	–	–	4396.4	(121.8)	18.55%	36.866	$\ll 0.001$	0.0	$\ll 0.001$
0.4	81	–	–	4430.6	(133.8)	19.47%	36.577	$\ll 0.001$	0.0	$\ll 0.001$
0.5	81	–	–	4514.9	(129.4)	21.75%	41.651	$\ll 0.001$	0.0	$\ll 0.001$
0.6	81	–	–	4591.5	(139.0)	23.81%	43.721	$\ll 0.001$	0.0	$\ll 0.001$
0.7	78	–	–	4603.2	(133.8)	24.13%	44.977	$\ll 0.001$	0.0	$\ll 0.001$
0.8	78	–	–	4641.6	(153.0)	25.16%	43.200	$\ll 0.001$	0.0	$\ll 0.001$
0.9	78	–	–	4675.2	(112.6)	26.07%	53.307	$\ll 0.001$	0.0	$\ll 0.001$
1.0	79	–	–	4695.1	(121.2)	26.61%	52.553	$\ll 0.001$	0.0	$\ll 0.001$

Table D.6: Statistical significance testing for DENSE distribution graphs with $f = 0.9$.

UNOS		*Independent*		*Combined*			t-test		Mann-Whitney	
$p_{K \to L}$	n	Avg. #	Stdev	Avg. #	Stdev	% Gain	t	p	U	p
0.0	82	4003.8	(108.3)	–	–	–	–	–		
0.1	86	–	–	4099.5	(108.3)	2.39%	5.689	$\ll 0.001$	1959.5	$\ll 0.001$
0.2	84	–	–	4162.9	(124.6)	3.97%	8.719	$\ll 0.001$	1154.5	$\ll 0.001$
0.3	80	–	–	4211.5	(119.3)	5.19%	11.538	$\ll 0.001$	617.5	$\ll 0.001$
0.4	79	–	–	4210.4	(109.2)	5.16%	11.978	$\ll 0.001$	561.0	$\ll 0.001$
0.5	74	–	–	4252.8	(103.6)	6.22%	14.539	$\ll 0.001$	306.5	$\ll 0.001$
0.6	76	–	–	4263.4	(115.3)	6.48%	14.501	$\ll 0.001$	303.0	$\ll 0.001$
0.7	76	–	–	4304.1	(112.8)	7.50%	16.961	$\ll 0.001$	185.5	$\ll 0.001$
0.8	62	–	–	4313.6	(124.6)	7.74%	15.813	$\ll 0.001$	143.0	$\ll 0.001$
0.9	67	–	–	4298.3	(121.2)	7.36%	15.544	$\ll 0.001$	181.0	$\ll 0.001$
1.0	68	–	–	4304.3	(120.7)	7.51%	15.949	$\ll 0.001$	163.0	$\ll 0.001$

Table D.7: Statistical significance testing for UNOS distribution graphs.

"But there's a lot of strategy involved as well—I use both apps, I go back and forth ..."

– YouTube user "Anthony D", driver for both Uber and Lyft

E

Competing dynamic matching markets

E.1 Introduction

In matching problems, a central clearinghouse pairs agents with other agents, transactions, or contracts. Most classical matching problems—matching medical residents to hospitals, matching students to schools—are static, where agents and items exist at the same time, are matched, and then the market disappears; however, many real-world matching problems are dynamic, with agents arriving and departing over time in a persistent market. In Chapters 6 and 8, we explored optimization-based approaches to learning matching policies for a single dynamic matching market; this appendix presents ongoing work that addresses a generalization of that setting.

Many dynamic matching applications involve *multiple* competing clearinghouses with overlapping sets of participants. For example, a lonely graduate student may register on two dating websites (e.g., Match.com and OkCupid), or choose to only register on one. Thus, a member of both sites can be matched to any member of either site, while single-site members can only be matched to members of their specific dating market. The clearinghouses then compete on a metric like total number of matches. It is also common for patient-donor pairs in kidney exchange to register on multiple exchanges, an application we explore in detail later.

In this chapter, we explore, in a dynamic matching setting, how rival clearinghouses affect global social welfare in terms of total agents matched relative to a world in which all agents enter exactly one market, which can optimize for how to match them independently. This is early-stage work with Sanmay Das and Zhuoshu Li at Washington University in St. Louis; a preliminary version of this research appeared at AMMA-2015 [65].

E.1.1 Our contribution

This chapter's major contribution is the extension of a recent framework of dynamic matching due to Akbarpour et al. [9] to two rival matching markets with overlapping pools. Specifically, we formalize a two-market model where agents enter one market or both markets; they can then be matched to other agents who have joined the same market or both markets. The markets adhere to different matching policies, with one matching greedily and the other building market thickness through a patient policy. We provide an analytic lower bound on the *loss*, or the expected fraction of vertices who enter and leave the pool without finding a match, of the two-market model and show that it is higher than running a single "patient" market. We also provide a quantitative method for determining the loss of the two-market model.

Our work draws motivation from kidney exchange, an instantiation of barter exchange where patients paired with willing but medically incompatible donors swap those donors with other patients. In the United States, multiple fielded kidney exchanges exist, and patient-donor pairs are entered simultaneously into one or more of these markets, based on geographical location, travel preferences, home transplant center preferences, or other logistical reasons. Individual kidney exchange clearinghouses have incentive to compete on number of matches performed within their specific pools; yet, fragmenting the market across multiple exchanges operating under different matching policies may lower global welfare. We provide the first experimental evidence on dynamic kidney exchange graphs showing that this may indeed be the case.

E.1.2 Related work

Most related to our work is a recent paper by Akbarpour et al. [9], which presents a general framework for bilateral dynamic matching in a single market and analyzes the efficacy of a variety of matching policies over time. We build directly on that framework and delay a more in-depth review of that work until Section E.2.

Dynamic matching in a *single* market has been explored in many domain-specific applications. Some examples are given below for both one-sided and two-sided traditional matching markets, as well as for barter exchanges; this list is not exhaustive.

One-sided markets. In these settings, only one side (the *agents*) has preferences over the other (the *items*). Waiting lists are used in many applications as a mechanism for allocating the items, which are scarce resources, to agents. Both agents and items arrive over time, and an agents' priority for an arriving item can be set by a variety of factors. Examples of waiting list applications include public housing assignment [127, 128, 145] and cadaveric organ allocation [37, 202, 223]. In a two-period dynamic housing allocation problem, agents can either apply for a public good (e.g., a house) in the first stage and receive priority in that stage, or opt out in the first stage and receive priority in the second stage [3]. Other variants of the dynamic housing allocation problem have also been addressed where, e.g., agents arrive and depart and, upon departure, an agent's allocated item is then given to an existing agent in the waiting pool [42, 139].

Two-sided markets. In two-sided markets, participating agents belong to one of two disjoint sets (e.g., "firms" or "workers"), but an agent on either side will have preferences over those on the other. In online labor marketplaces like Upwork, employers and applicants arrive and depart over

time and are interested in finding an acceptable match [18, 123]. In the dynamic school choice problem, schools exist permanently and indefinitely, but students arrive and depart periodically in a discrete time model [130]. Students matched to a school at one time period may be matched to a new school at a different time period. Schools and students have preference orderings over each other, based on the utility provided to one side by being allocated an element or elements of the other. Finally, generalizations of the online bipartite matching problem as originally introduced by Karp et al. [129] have recently seen great real-world impact in Internet ad allocation [158, 159].

We note that our work does not assume a bipartite structure in the matching graph—as in [3, 18, 42, 111, 123, 127, 128, 130, 139, 145, 202, 213, 223] and much of the static matching mechanism design literature—and involves more than a single market.

Barter exchange. In barter exchange, agents can directly swap goods with other agents in cycles of length greater than or equal to two. One fielded example is kidney exchange [184], where patients with end-stage renal failure and willing but incompatible paired donors swap those donors with other patients. Ünver [213] was the first to address dynamic kidney exchange, where patient-donor pairs arrive and depart over time, with recent follow-up work by Ashlagi et al. [24] and Anderson et al. [15]. All three papers look at matching policies that aim to maximize (discounted) social welfare. Particularly relevant to real-world kidney exchanges are *batching* policies, where a market clearing occurs at a fixed interval; some theoretical and empirical explorations of this class of policy has been performed [15, 17, 24, 27]. Learning approaches have also been used to determine more complex matching policies that adhere to specific data distributions [70, 73]; some of these approaches were discussed in Chapters 6 and 8.

To our knowledge, no work in the general barter exchange or kidney exchange literature has addressed multiple competing exchanges, a problem that is especially relevant in the US now, and, as kidney exchanges move to international swapping, will soon become relevant worldwide.

Interacting mechanisms have been studied in a variety of domains like auctions [52, 162], adaptations of settings from the classical multi-agent systems literature [196], and in two-sided networks that typically exhibit winner-take-all dynamics, where only one or a few large players (e.g., credit card companies, computer operating systems, HMOs) prevail due to network effects [87]. Recent work by Ostrovsky [171] generalizes traditional two-sided matching to a supply chain model with interconnected markets represented as nodes in a path, such that an "upstream" neighbor's supply overlaps with its "downstream" neighbor's demand. Relatively little work focuses on markets competing based on variable scheduling or clearing policies, with notable exceptions in cloud or grid computing [21] and in financial markets [51]. To the best of our knowledge, no work looks at competition between two markets in a general framework of dynamic bilateral matching, as this chapter does.

E.2 Greedy and patient exchanges

We begin by restating some of the most important results of Akbarpour et al. [9], which will serve as the foundation for our model of competing exchanges. Akbarpour et al. analyze "greedy" and "patient" matching policies—and interpolations between the two—by building stochastic continuous-time bilateral matching models of exchanges running these policies, then measuring the efficacy

of the policies in terms of discounted social welfare.

More specifically, an exchange is running in the continuous-time interval $[0, T]$, with agents arriving according to a Poisson process with rate parameter $m \geq 1$. The exchange determines whether potential bilateral transactions between agents are either acceptable or unacceptable. The probability of an *acceptable* transaction existing between any pair of distinct agents is defined as d/m, $0 \leq d \leq m$, and is independent of any other pair of agents in the market. Each agent a remains in the market for a *sojourn* $s(a)$ drawn independently from an exponential distribution with rate parameter $\lambda = 1$; the agent becomes *critical* immediately before her sojourn ends, and this criticality is known to the exchange. An agent leaves either upon being matched successfully by the exchange or upon becoming critical and remaining unmatched, at which point she perishes.

At any time $t \geq 0$, the network of acceptable transactions among agents forms a random graph $G_t = (A_t, E_t)$, where the agents in the exchange at time t form the vertex set A_t, and the acceptable transactions between agents forms the edge set E_t. We assume $A_0 = \emptyset$. Let A_t^n denote the set of agents who enter the exchange at time t, such that with probability 1, $|A_t^n| \leq 1$ for any $t \geq 0$. Finally, let $A = \cup_{t \leq T} A_t^n$.

Akbarpour et al. [9] present a parameterized space of online *matching policies*, with a focus specifically on two: Patient and Greedy. (In the next section, we will present a novel model of two overlapping exchanges, one running the Patient policy and the other running the Greedy policy.) As described above, vertex arrivals are treated as a continuous-time stochastic process. These policies behave as follows.

Greedy. The Greedy matching algorithm attempts to match each entering agent immediately by selecting one of its neighbors (if a neighbor exists at the time of entry) uniformly at random. One obvious consequence of this is that the remaining graph of unmatched agents at any instant is always empty. We refer to a market running this policy as the *Greedy market* or simply *Greedy* for the rest of the chapter.

Patient. The Patient matching algorithm attempts to match each agent only at the instant she becomes critical. As with Greedy, if a critical agent has multiple neighbors, only one is selected uniformly at random. We refer to a market running the Patient policy as a *Patient market* or simply *Patient* when appropriate.

If the random graph model is Erdős-Rényi [88] when not considering arrivals, departures, and matching, then the remaining graph at any instant is also Erdős-Rényi with parameter d/m; furthermore, d is the average degree of the agents. Both the Patient and Greedy policies maintain this observation.

The main result of Akbarpour et al. [9] is that waiting to thicken the market can be substantially more important than increasing the speed of transactions. Formally, the Patient exchange dramatically reduces the number of agents who perish (and thus leave the exchange without finding a match) compared to the Greedy exchange.

In the Akbarpour et al. [9] paper, an agent a receives zero utility if she perishes, or $u(a) = 0$. If she is matched, she receives a utility of 1 discounted at rate δ, or $u(a) = e^{-\delta s(a)}$. In this work, we focus on the special case of $\delta = 0$ (i.e., we only consider whether or not an agent is matched), and leave the $\delta \neq 0$ case for future research. Let $\text{ALG}(T) := \{a \in A : a \text{ is matched by ALG by time } T\}$. Then, in this model, the loss of an algorithm ALG is defined

as the ratio of the expected number of perished agents to the expected size of A, as shown in Equation E.1.

$$\mathbf{L}(\text{ALG}) = \frac{E[|A - \text{ALG}(T) - A_T|]}{E(|A|)} = \frac{E[|A - \text{ALG}(T) - A_T|]}{mT} \tag{E.1}$$

At any time $t \in [0, T]$, let $Z_{g,t}, Z_{p,t}$ represent the size of the pools under the Greedy and Patient matcing policies, respectively. Then, Akbarpour et al. [9] proved that the Markov chain on $Z_{\cdot,t}$ has a unique stationary distribution under either of those policies. Furthermore, let $\pi_g, \pi_p : \mathbb{N} \to \mathbb{R}^+$ be the unique stationary distribution of the Markov chain on $Z_{g,t}, Z_{p,t}$, respectively, and let $\xi_g := \mathbb{E}_{Z_g \sim \pi_g}[Z_g], \xi_p := \mathbb{E}_{Z_p \sim \pi_p}[Z_p]$ be the expected size of the pool under the stationary distribution under Greedy and Patient. Then, the following observations can be made.

Loss of Greedy. If a Greedy exchange is run for a sufficiently long time, then $\mathbf{L}(\text{Greedy}) \approx \frac{\xi_g}{m}$. The intuition here is that the Greedy pool is (almost) always an empty graph. Equation E.2 formalizes the loss.

$$\mathbf{L}(\text{Greedy}) = \frac{1}{mT}\mathbb{E}\left[\int_0^T Z_{g,t} dt\right] = \frac{1}{mT}\int_0^T \mathbb{E}\left[Z_{g,t}\right] dt \tag{E.2}$$

Loss of Patient. If a Patient exchange is run for a sufficiently long time, at any point in time it is an Erdős-Rényi random graph. So once an agent becomes critical, she has no acceptable transaction with probability $(1 - d/m)^{Z_{p,t}-1}$. Thus, $\mathbf{L}(\text{Patient}) \approx \frac{\xi_p(1-d/m)^{\xi_p - 1}}{m}$. Equation E.3 formalizes the loss of a Patient market.

$$\begin{aligned} \mathbf{L}(\text{Patient}) &= \frac{1}{mT}\mathbb{E}\left[\int_0^T Z_{p,t}(1 - d/m)^{Z_{p,t}-1} dt\right] \\ &= \frac{1}{mT}\int_0^T \mathbb{E}\left[Z_{p,t}(1 - d/m)^{Z_{p,t}-1}\right] dt \end{aligned} \tag{E.3}$$

E.3 Overlapping exchanges

The key result of Akbarpour et al. [9] is that a greedy dynamic matching market leads to significantly lower global social welfare than a patient matching market with full knowledge of criticality. The central question of this chapter is what happens in a situation where a greedy exchange and a patient exchange exist *simultaneously* and compete with each other to match some shared portion of the population. Agents in this overlapping subset of the population join both exchanges simultaneously and accept the first match offer from either of the constituent exchanges.

Drawing on Section E.2, we model this in a similar stochastic, continuous-time framework as follows. Agents arrive at the Competing market (a model for the whole system, incorporating both the Greedy and Patient exchanges) at some rate m according to a Poisson process. For each agent, the probability of entering *both* the Greedy exchange and the Patient exchange is γ, the probability of entering the Greedy exchange alone is $(1 - \gamma)\alpha$, and the probability of entering the Patient exchange alone is $(1 - \gamma)(1 - \alpha)$, where $\gamma, \alpha \in [0, 1]$. The probability that a bilateral transaction

between each pair of agents is acceptable remains d/m, *conditioned on both agents being mutually "visible" to an exchange.* The agents' rates of perishing, received utility for being (un)matched, and other settings are otherwise the same as in Section E.2.

We analyze the Competing market as three separate evolving pools:

Greedy$_c$ is the pool consisting of agents who enter the Greedy exchange only (with probability $\alpha(1-\gamma)$).

Patient$_c$ is the pool consisting of agents who enter the Patient exchange only (with probability $(1-\alpha)(1-\gamma)$).

Both$_c$ is the pool consisting of agents who enter both exchanges (with probability γ).

We use $\hat{Z}_{g,t}$, $\hat{Z}_{p,t}$ and $\hat{Z}_{b,t}$ to denote the size of Greedy$_c$, Patient$_c$ and Both$_c$, respectively, at any time t. Similar to an exchange running a single Greedy or Patient matching policy, the Markov chain on $\hat{Z}_{\cdot,t}$ also has a unique stationary distribution. Let $\hat{\pi}_\cdot : \mathbb{N} \to \mathbb{R}^+$ be the unique stationary distribution of the Markov chain on $\hat{Z}_{\cdot,t}$, and let $\hat{\xi}_\cdot := \mathbb{E}_{\hat{Z}_\cdot \sim \hat{\pi}_\cdot}[\hat{Z}_\cdot]$ be the expected size of the pool under the stationary distribution. Using this, we will define the loss of Greedy$_c$, $\hat{\mathbf{L}}$(Greedy$_c$), the loss of Patient$_c$, $\hat{\mathbf{L}}$(Patient$_c$), and the loss of Both$_c$, $\hat{\mathbf{L}}$(Both$_c$).

First, note that the graph formed by the agents in Greedy$_c$ is empty, so the loss—as in Equation E.2—can be approximated by $\hat{\mathbf{L}}(\text{Greedy}_c) \approx \frac{\hat{\xi}_g}{m}$.

Next, we consider the agents in Both$_c$. If an edge exists between an agent in Both$_c$ and an existing agent in Greedy$_c$ or another agent in Both$_c$, she will be matched immediately by the Greedy exchange (and thus does not contribute to the loss). Similar to the Greedy$_c$ case, at any point in time t, the Both$_c$ pool is an empty graph; thus, any unmatched agents who become critical in Both$_c$ will only be matched to agents in Patient$_c$. Thus, these leftover agents in Both$_c$ have no acceptable transactions with probability $(1-d/m)^{\hat{Z}_{p,t}}$. Since each agent becomes critical with rate 1, letting Competing market run for a sufficiently long time results in $\hat{\mathbf{L}}(\text{Both}_c) \approx \frac{\hat{\xi}_b(1-d/m)^{\hat{\xi}_p}}{m}$, where $\hat{\xi}_b, \hat{\xi}_p$ are the previously defined expected sizes of Both$_c$ and Patient$_c$.

Finally, we consider the Patient$_c$ pool. At any time t, the agents who remain in Patient$_c$ potentially have acceptable transactions with only the agents in Both$_c$ and the agents in Patient$_c$. Hence, in $\hat{Z}_{p,t}$, once an agent is critical, she has no acceptable transactions with probability $(1-d/m)^{\hat{Z}_{p,t}+\hat{Z}_{b,t}-1}$. Similarly, each agent becomes critical with rate 1; thus, if we allow the Competing market a sufficiently long execution window, $\hat{\mathbf{L}}(\text{Patient}_c) \approx \frac{\hat{\xi}_p(1-d/m)^{\hat{\xi}_p+\hat{\xi}_b-1}}{m}$.

Because the three pools of agents—Greedy$_c$, Patient$_c$, and Both$_c$—are disjoint (although they may be connected via possible transactions in the ways listed above), we can define the total loss of the Competing market as follows.

$$\mathbf{L}(\text{Competing}) \approx \frac{\hat{\xi}_g + \hat{\xi}_p(1-d/m)^{\hat{\xi}_p+\hat{\xi}_b-1} + \hat{\xi}_b(1-d/m)^{\hat{\xi}_p}}{m}. \tag{E.4}$$

A more precise version of Equation E.4 follow as Equation E.5; we will make use of this form

in Section E.5.

$$\begin{aligned}\mathbf{L}(\text{Competing}) &= \frac{1}{mT}\mathbb{E}\Bigg[\int_0^T \hat{Z}_{p,t}(1-d/m)^{\hat{Z}_{p,t}+\hat{Z}_{b,t}-1} \\ &\quad + \hat{Z}_{b,t}(1-d/m)^{\hat{Z}_{p,t}} + \hat{Z}_{g,t}dt\Bigg] \\ &= \frac{1}{mT}\int_0^T \mathbb{E}\big[\hat{Z}_{p,t}(1-d/m)^{\hat{Z}_{p,t}+\hat{Z}_{b,t}-1} \\ &\quad + \hat{Z}_{b,t}(1-d/m)^{\hat{Z}_{p,t}} + \hat{Z}_{g,t}\big]dt\end{aligned} \tag{E.5}$$

Unfortunately, we do not have a closed form expression for the stationary distribution or the expected size of the pool under the stationary distribution. We note that each of $\hat{\xi}_g$, $\hat{\xi}_p$, and $\hat{\xi}_b$ can be approximated well using Monte Carlo simulations—thus, Equation E.4 can be solved numerically. We do this in Section E.5.1 for two parameterizations of the rival market setting.

E.4 A bound on total loss

While we do not have a closed form for the exact expected loss of the Competing market as described by Equation E.4, we can provide bounds on the overall loss. In this section, we give one such bound for the global loss under the constraint that Greedy$_c$ is more likely to receive agents than the overlapping Both$_c$ exchange. Formally, this occurs when $\gamma \leq 0.5$ and $\alpha \geq \frac{\gamma}{1-\gamma}$. We also impose some loose requirements on the arrival rate of vertices to the exchange and the probability of an acceptable transaction existing between two agents; intuitively, the exchange cannot be "too small" or "too sparse," which we formalize below. Under these assumptions, we use the bound to prove Theorem 22, which states that a single Patient market outperforms the Competing market.

Theorem 22. *Assume $\gamma \leq 0.5$, $m > 10d$, and $\alpha(1-\gamma) \geq \max\left\{\gamma, \frac{1}{2}e^{-d/2}(1+3d)\right\}$. Then, as $m \to \infty$ and $T \to \infty$, almost surely*

$$\mathbf{L}(\textit{Competing}) > \mathbf{L}(\textit{Patient}).$$

Proof. We prove the theorem by giving a lower bound on $\hat{\mathbf{L}}(\text{Greedy}_c)$, the loss of only the greedy portion of the Competing market. In our model, the fraction of agents entering only the Greedy$_c$ side of the market is $\alpha(1-\gamma)$; for notational simplicity, we use $x := \alpha(1-\gamma)$ in this proof. Similarly, the fraction of agents entering Both$_c$ is γ; again, for notational simplicity, we use $y := \gamma$ throughout this proof.

As before, let $\hat{Z}_{g,t}$ be the size of Greedy$_c$ at any $t \in [0,T]$, and $\hat{\tau}$ the expected size of the Greedy$_c$ pool. Similarly, let $\hat{Z}_{b,t}$ be the size of Both$_c$ at any $t \in [0,T]$, and $\hat{\eta}$ the expected size of the Both$_c$ pool. That is,

$$\hat{\tau} := \mathop{\mathbb{E}}_{t\sim\text{unif}[0,T]}\left[\hat{Z}_{g,t}\right] \quad \text{and} \quad \hat{\eta} := \mathop{\mathbb{E}}_{t\sim\text{unif}[0,T]}\left[\hat{Z}_{b,t}\right].$$

By assumption, $\alpha(1-\gamma) \geq \gamma$; that is, the arrival rate of Greedy$_c$ is greater than or equal to the arrival rate of Both$_c$. In this case, $\hat{\tau} \geq \hat{\eta}$; the Greedy matching policy removes verties from both

Both_c and Greedy_c, while the Patient matching policy removes vertices from only Both_c, which means the matching rate for Both_c is greater than the matching rate for Greedy_c.

From Akbarpour et al. [9], we know the expected rate of perishing of the individual Greedy exchange is equal to the pool size because the Greedy matching policy does not react to the criticality of an agent at any time t in its pool and each critical agent will perish with probability 1. Therefore, we can draw directly on Equation E.2 to write

$$\hat{\mathbf{L}}(\text{Greedy}_c) = \frac{1}{xmT}\mathbb{E}[\int_{t=0}^{T} dt\ \hat{Z}_{g,t}] = \frac{\hat{\tau}}{xm}. \tag{E.6}$$

We know x and m, so lower bounding $\hat{\tau}$ will result in an analytic lower bound on $\hat{\mathbf{L}}(\text{Greedy}_c)$. Following the ideas of Akbarpour et al. [9], we do this by lower bounding the probability that an agent a does not *ever* have an acceptable transaction for the duration of her sojourn $s(a)$. Because these agents cannot be matched by any matching policy, this directly gives a lower bound on $\hat{\mathbf{L}}(\text{Greedy}_c)$. Toward this end, fix an agent $a \in A$ who enters Greedy_c at time $t_0 \in \text{unif}[0, T]$ and draws a sojourn $s(a) = t$. Let $f_{s_a}(t)$ be the probability density function at t of $s(a)$. Then we can write the probability that a will never have a neighbor (i.e., possible match) as

$$\begin{aligned}\mathbb{P}\left[N(a) = \emptyset\right] = \int_{t=0}^{\infty} f_{s_a}(t)\mathbb{E}\left[(1-d/m)^{\hat{Z}_{g,t_0}+\hat{Z}_{b,t_0}}\right] \\ \mathbb{E}\left[(1-d/m)^{|AG^n_{t_0,t_0+t}+AB^n_{t_0,t_0+t}|}\right] dt,\end{aligned}$$

where $AG^n_{t_0,t_0+t}$ (resp. $AB^n_{t_0,t_0+t}$) denotes the set of agents who enter Greedy_c (resp. Both_c) in time interval $[t_0, t_0 + t]$. The first expectation captures the probability that agent a has no matching at the moment of entry and the second expectation considers the probability that no new agents that can match with a arrive during her sojourn.

Using Jensen's inequality, we have

$$\begin{aligned}\mathbb{P}\left[N(a) = \emptyset\right] &\geq \int_{t=0}^{\infty} e^{-t}(1-d/m)^{\mathbb{E}[\hat{Z}_{g,t_0}+\hat{Z}_{b,t_0}]} \\ &\quad (1-d/m)^{\mathbb{E}[|AG^n_{t_0,t+t_0}+AB^n_{t_0,t+t_0}|]}dt \\ &= \int_{t=0}^{\infty} e^{-t}(1-d/m)^{\hat{\tau}+\hat{\eta}}(1-d/m)^{(x+y)mt}dt.\end{aligned}$$

From the assumptions in the theorem statement, $\frac{d}{m} < \frac{1}{10}$, so $1 - d/m \geq e^{-d/m-d^2/m^2}$. Also, as

described earlier, $\hat{\tau} \geq \hat{\eta}$ (when $\gamma \leq 0.5$ and $\alpha \geq \frac{\gamma}{1-\gamma}$, as assumed). Therefore,

$$\begin{aligned}
\hat{\mathbf{L}}(\text{Greedy}_c) &\geq \mathbb{P}\left[N(a) = \emptyset\right] \\
&\geq e^{-(\hat{\tau}+\hat{\eta})(d/m+d^2/m^2)} \\
&\quad \times \int_{t=0}^{\infty} e^{-t-(x+y)td-(x+y)td^2/m} dt \\
&\geq \frac{1-(\hat{\tau}+\hat{\eta})(1+d/m)d/m}{1+(x+y)d+(x+y)d^2/m} \\
&\geq \frac{1-2\hat{\tau}(1+d/m)d/m}{1+(x+y)d+(x+y)d^2/m},
\end{aligned} \tag{E.7}$$

where the third inequality is obtained from the fact that $e^{-z} \geq 1 - z$ when $z \geq 0$, here $z = (\hat{\tau} + \hat{\eta})(d/m + d^2/m^2)$.

Combining Equation E.6 and Equation E.7,

$$\hat{\mathbf{L}}(\text{Greedy}_c) = \frac{\hat{\tau}}{xm} \geq \frac{1-2\hat{\tau}(1+d/m)d/m}{1+(x+y)d+(x+y)d^2/m},$$

which gives us a lower bound for $\hat{\tau}$,

$$\hat{\tau} \geq \frac{xm}{1+(3x+y)d+(3x+y)d^2/m}.$$

Thus, as $m \to \infty$, we get,

$$\begin{aligned}
\hat{\mathbf{L}}(\text{Greedy}_c) &\geq \frac{1}{1+(3x+y)d+(3x+y)d^2/m} \\
&\geq \frac{1}{1+3d}.
\end{aligned}$$

We are interested in bounding the total loss of the Competing market, which is $\mathbf{L}(\text{Competing}) = x\hat{\mathbf{L}}(\text{Greedy}_c) + (1-\alpha)(1-\gamma)\hat{\mathbf{L}}(\text{Patient}_c) + y\hat{\mathbf{L}}(\text{Both}_c)$. By definition, both $\hat{\mathbf{L}}(\text{Patient}_c) \geq 0$ and $\hat{\mathbf{L}}(\text{Both}_c) \geq 0$, and by Equation E.4, $\hat{\mathbf{L}}(\text{Greedy}_c) \geq \frac{1}{1+3d}$. Thus,

$$\mathbf{L}(\text{Competing}) \geq \frac{x}{1+3d}.$$

Akbarpour et al. [9] showed that running an *individual* Patient market results in exponentially small loss $\mathbf{L}(\text{Patient}) < \frac{1}{2}e^{-d/2}$. Thus, as $T, m \to \infty$, we can get,

$$\mathbf{L}(\text{Competing}) > \mathbf{L}(\text{Patient}). \tag{E.8}$$

□

We note that the result of Theorem 22 holds for only a section of the possible parameterizations of a Competing market—specifically, when $\gamma \leq 0.5$ and $\alpha \geq \frac{\gamma}{1-\gamma}$. In the next section, we will give numerical results showing that this result—that the loss of the Competing market is greater than the loss of an individual Patient exchange—appears to hold for a vastly larger space of values of γ and α. Indeed, experimentally, we will see that the loss of the Competing market is sometimes greater than the loss of an individual *Greedy* exchange, which itself is substantially greater than the loss of an individual Patient exchange.

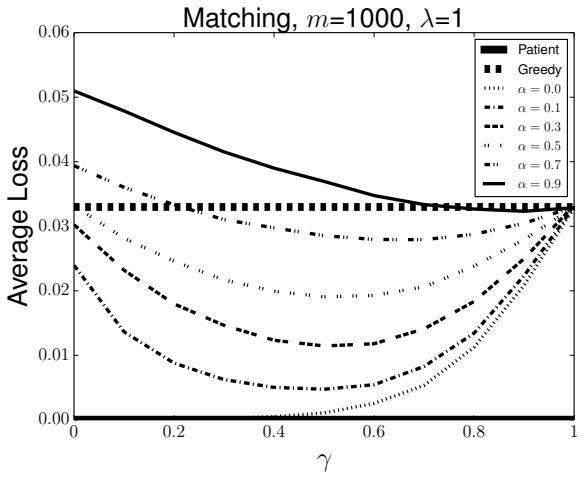
E.5 Experimental validation

In this section, we provide experimental validation of the theoretical results presented in Sections E.3 and E.4. Section E.5.1 quantifies the loss due to competing markets as described by Equation E.4, while Section E.5.2 expands the model to *kidney exchange* and draws from realistic data to quantify the loss of competing kidney exchange clearinghouses.

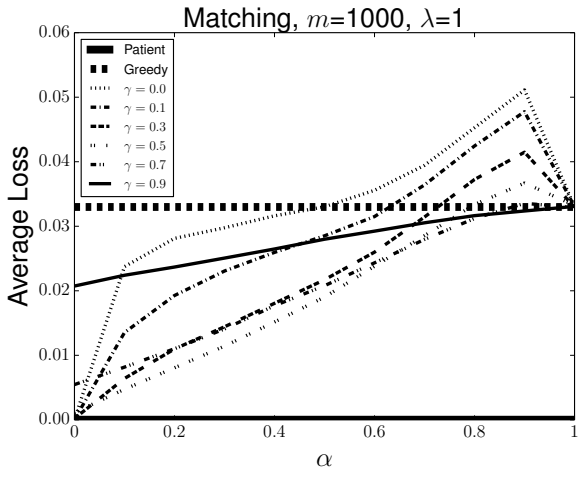
E.5.1 Dynamic matching

In Section E.3, we gave a method for computing the expected loss due to competing markets as Equation E.4; however, we were unable to derive closed forms for the expected size of the competing, patient, and greedy pools ($\hat{\xi}_b$, $\hat{\xi}_p$, and $\hat{\xi}_g$, respectively) under the stationary distribution. These quantities can be estimated using Monte Carlo simulation for different entrance rates m. We do that now.

Figures E.1a and E.1b simulate agents entering the Greedy$_c$, Both$_c$, and Patient$_c$ markets according to a Poisson process with rate parameter $m = 1000$ and remaining for a sojourn drawn from an exponential distribution with rate parameter $\lambda = 1$. An agent chooses to enter Both$_c$ with probability γ, only Greedy$_c$ with probability $\alpha(1-\gamma)$, and only Patient$_c$ with probability $(1-\alpha)(1-\gamma)$, as in the theory above. We vary $\alpha \in \{0, 0.1, \ldots, 1\}$ and $\gamma \in \{0, 0.1, \ldots, 1\}$, and plot the global loss realized for each of these parameter settings.



(a) Average loss (y-axis) as the overlap between markets γ increases (x-axis), for different values of α.



(b) Average loss (y-axis) as the probability α of entering Patient$_c$ or Greedy$_c$ changes (x-axis), for different values of the market overlap γ.

Figure E.1: Average loss as the probability α of entering Patient$_c$ or Greedy$_c$ (left) or the overlap between the two markets γ (right) changes, with entrance rate parameter $m = 1000$ and $d = 2$. The loss of individual Patient and Greedy markets are shown as thick black and thick dashed bars, respectively.

Immediately obvious is that running a single Patient market results in dramatically less loss than competing markets, for all different values of α and γ. Furthermore, we see that the loss of a single Greedy market is also dramatically higher than the loss of a single Patient market, as predicted by Akbarpour et al. [9]. Indeed, from Equation E.3 we would expect the single Patient market to have essentially zero loss, so these experiments show that adding in a rival Greedy$_c$ market increases loss. In fact, as the left side of Figure E.1a and the right side of Figure E.1b show, it is the case that if the markets do not overlap substantially (i.e., γ is low) and agents are more likely to enter the greedy side of the market (i.e., α is near 1), then the loss of the competing market is worse than running a single Greedy market! This is due in part to the decrease in market thickness on the Patient$_c$ side of the market—a behavior we will see exacerbated below and in the kidney exchange experiments of Section E.5.2.

Figure E.2 decreases the rate parameter of the entrance Poisson process to $m = 100$, while holding the probability of an acceptable transaction between two agents at that of Figures E.1a and E.1b (so $d = 2$, leading to $2/100 = 2\%$). With fewer participants in the market overall, all the qualitative results of the $m = 1000$ markets above are amplified. The individual Greedy market's loss is now 5.9% worse than the individual Patient market (as opposed to 3.3% in the $m = 1000$ case); both individual markets' losses are substantially higher as well. Similarly, the parameter settings for which the competing market scenario has higher loss than either individual market are much broader than the $m = 1000$ case, which is a product of market thinness.

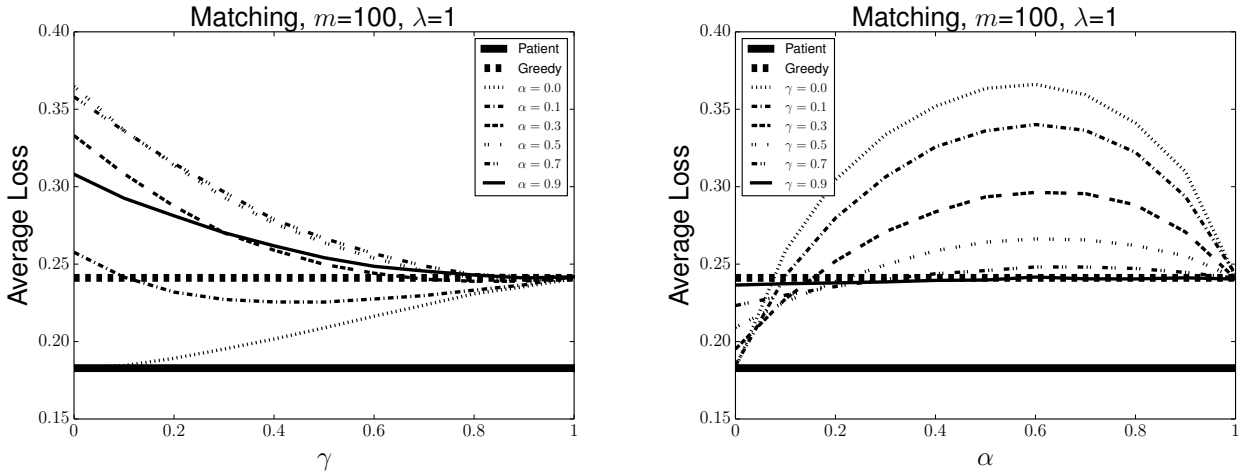


Figure E.2: Average loss as the probability α of entering Patient$_c$ or Greedy$_c$ (left) or the overlap between the two markets γ (right) changes, with entrance rate parameter $m = 100$ and $d = 2$. The loss of individual Patient and Greedy markets are shown as thick black and thick dashed bars, respectively.

E.5.2 Dynamic kidney exchange

In this section, we expand our matching model to one of *barter exchange*, where agents endowed with items participate in directed, cyclic swaps of size greater than or equal to two. One recently-

fielded barter application is *kidney exchange*, where patients with kidney failure swap their willing but incompatible organ donors with other patients. We focus on that application here. Dynamic barter exchange generalizes the matching model presented above, so we would not expect the earlier theoretical results to adhere exactly. Interestingly, as we show in Sections E.5.2 and E.5.2, the qualitative ranking of matching policy loss (with a patient market outperforming a greedy market, both of which outperform two rival markets) remains.

This section's experiments draw from two kidney exchange compatibility graph distributions. One distribution, which we call SAIDMAN (US), was designed to mimic the characteristics of a nationwide exchange in the United States in steady state [191]. Yet, kidney exchange is still a nascent concept in the US, so fielded exchange pools do not adhere to this model. With this in mind, we also include results performed on a dynamic pool generator that mimics the United Network for Organ Sharing (UNOS) nationwide exchange, drawing data from the first 193 match runs of that exchange. We label the distribution derived from this as UNOS.

Formally, we represent a kidney exchange pool with n patient-donor pairs as a directed *compatibility graph* $G = (V, E)$, such that a directed edge exists from patient-donor pair $v_i \in V$ to patient-donor pair $v_j \in V$ if the donor at v_i can give a kidney to the patient at v_j. Edges exist or do not exist due to the medical characteristics (blood type, tissue type, relation, and many others) of the patient and potential donor, as well as a variety of logistical constraints. Our generators take care of these details; for more information on how edge existence checking is done in the SAIDMAN (US) and UNOS distributions, see Saidman et al. [191] or Chapter 8 of this thesis, respectively. Importantly, under either distribution, there is no longer a costant probability "d/m" of an acceptable transaction existing between any two agents.

Vertices arrive via a Poisson process with rate parameter $m = 100$ and depart according to an exponential clock with rate parameter $\lambda = 1$ as before, and choose to enter either exchange or both with the previously-defined probabilities γ and α. However, a "match" now only occurs when a vertex forms either a 2-*cycle* or 3-*cycle* with one or two other vertices, respectively.[1] Section E.5.2 performs experiments on 2-cycles alone, which adheres more closely to the theoretical setting above (2-cycles can be viewed as a single undirected edge between two vertices), while Section E.5.2 expands this to both 2- and 3-cycles.

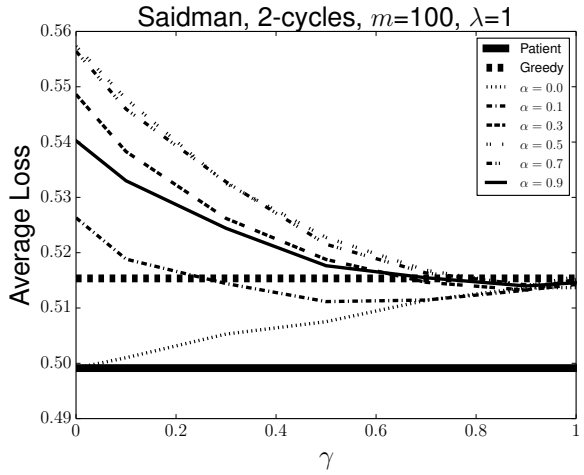
Code to replicate the experiments in this section is available at `github.com/JohnDickerson/KidneyExchange`. This codebase includes our experimental framework, dynamic exchange simulator, and graph generators but, due to privacy concerns, does not include the real match runs from the UNOS kidney exchange.

Kidney exchange with 2-cycles only

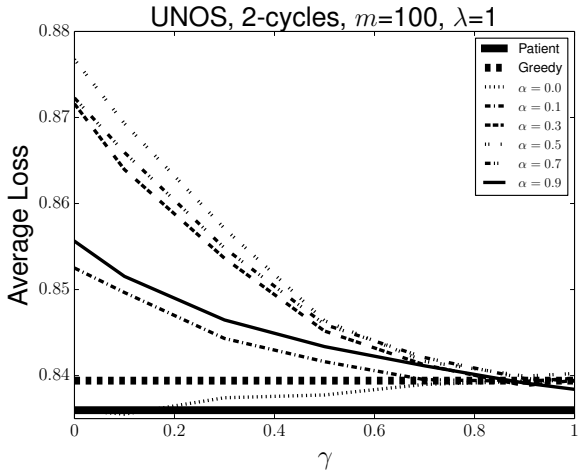
We now present results for dynamic matching under competing Patient$_c$ and Greedy$_c$ kidney exchanges, both of which use only 2-cycles. Figure E.3a and Figure E.3b show losses incurred in

[1] In fielded kidney exchange, cycles longer than some short cap L (e.g., $L = 3$ at the UNOS exchange and many others) are typically infeasible to perform due to logistical constraints, and thus are not allowed. We adhere to that constraint here. Fielded exchanges also realize gains from *chains*, where a donor without a paired patient enters the pool and triggers a directed path of transplants through the compatibility pool. We do not include chains in this work.

our parameterized market when run on SAIDMAN (US)-generated and UNOS-generated pools, respectively.



(a) Average loss under various values of γ and α for the SAIDMAN (US) distribution with 2-cycles only.



(b) Average loss under various values of γ and α for the UNOS distribution with 2-cycles only.

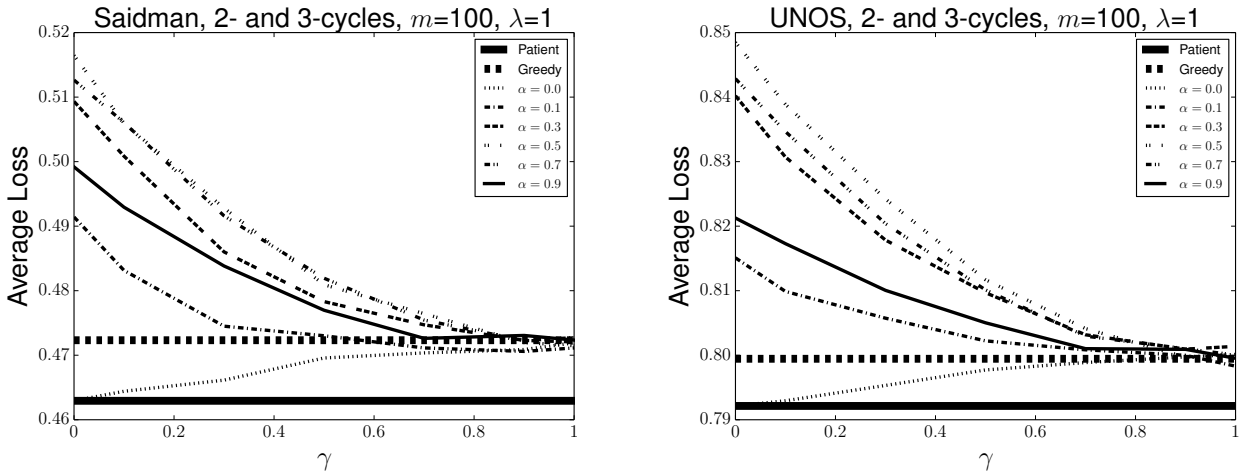
Figure E.3: Average loss under various values of γ and α with 2-cycles only.

While the barter exchange environment under either the SAIDMAN (US) or UNOS distributions clearly breaks the structural properties of the stationary distribution of the underlying Markov process used in our theoretical results, the qualitative results of these experiments align with the traditional dynamic matching results of Section E.5.1. The overall loss realized by UNOS is substantially higher than that realized by SAIDMAN (US) because, in general, UNOS-generated graphs are more sparse than those from the SAIDMAN (US) family. Similarly, in either distribution there exist "highly-sensitized" vertex types that are extremely unlikely to find a match with another randomly selected vertex, and thus almost certainly create loss. Indeed, both Figure E.3a and E.3b exhibit higher loss than the similarly-parameterized Figure E.2 of Section E.5.1.

Kidney exchange with both 2- and 3-cycles

We now extend our experiments to allow for "matches" that include both 2- and 3-cycles. Unlike Section E.5.1 or E.5.2, where a matched edge was chosen uniformly at random from the set of all acceptable transactions between a distinguished vertex and its neighbors, in these results we may wish to distinguish a potential match from others (for example, by choosing a 3-cycle before a 2-cycle, as the former results in a larger myopic decrease in the market's loss). Thus, given a set of possible 2- and 3-cycle matches, we consider two matching policies: UNIFORM selects a cycle at random from the set of possible matches, regardless of cycle cardinality, while UNIFORM3 selects a 3-cycle randomly (if one exists), otherwise a random 2-cycle.

Figures E.4a and E.4b show results for the SAIDMAN (US) and UNOS distributions, respectively, under the UNIFORM match selection policy. Intuitively, one might expect the loss of a matching policy run in the 2- and 3-cycle case to be less than the same policy run in the 2-cycle case alone, as the set of possible matches weakly increases in the former case. We see this behavior when comparing the SAIDMAN (US) results of Figure E.4a to the earlier 2-cycle-only SAIDMAN (US) results of Figure E.3a, witnessing a drop in global loss of around 4% for any parameter setting. We see a similar decrease in loss when comparing the new UNOS results of Figure E.4b to those in the 2-cycle case shown in Figure E.3b.

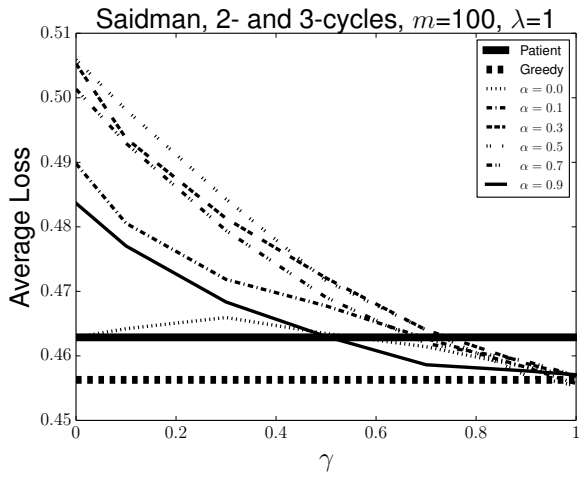


(a) Average loss under various values of γ and α for the SAIDMAN (US) distribution with both 2- and 3-cycles, under the UNIFORM matching policy.

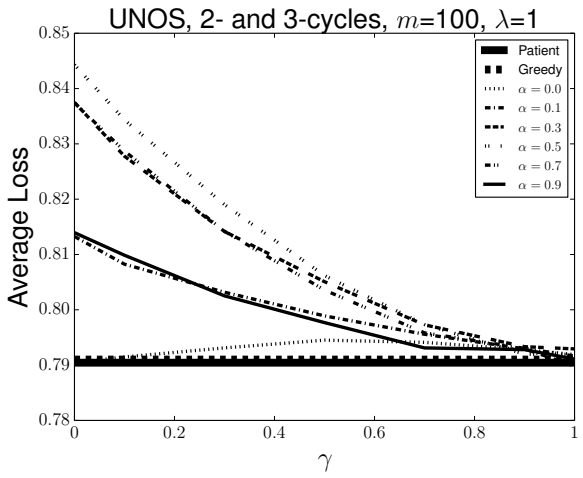
(b) Average loss under various values of γ and α for the UNOS distribution with both 2- and 3-cycles, under the UNIFORM matching policy.

Figure E.4: Average loss under various values of γ and α, with both 2- and 3-cycles, under the UNIFORM matching policy.

We now consider the UNIFORM3 matching policy, which would likely be closer to how a fielded exchange would act. Figures E.5a and E.5b show results for the SAIDMAN (US) and UNOS families of compatibility graphs, respectively. The loss of the individual Patient market does not change in either distribution, which is likely a byproduct of the thicker markets induced by its match cadence. Curiously, the loss of the individual Greedy market drops dramatically—to around the Patient loss in the UNOS case, and *below* Patient in the SAIDMAN (US) case. This large drop in Greedy loss is likely due in part to Greedy now "poaching" larger 3-cycles from the leftover market from which the Patient policy draws. The other qualitative results of earlier sections are repeated, with rival markets hurting global loss relative to either individual market for nearly all settings of γ and α.



(a) Average loss under various values of γ and α for the SAIDMAN (US) distribution with both 2- and 3-cycles, under the UNIFORM3 matching policy.



(b) Average loss under various values of γ and α for the UNOS distribution with both 2- and 3-cycles, under the UNIFORM3 matching policy.

Figure E.5: Average loss under various values of γ and α, with both 2- and 3-cycles, under the UNIFORM3 matching policy.

E.6 Conclusion & future research

Our main goal is to study the impact of competition between exchanges in a dynamic matching setting. In this chapter, we extended the recent dynamic matching model of Akbarpour et al. [9] to two rival matching markets with overlapping pools. Specifically, we formalized a two-market model where agents enter one market or both markets; they can then potentially be matched to other agents who have joined the same market or both markets. The markets, called Greedy and Patient, adhere to different matching policies. We provided an analytic lower bound on the *loss* of the two-market model and showed that it is higher than running a single Patient market. We also provided a quantitative method for determining the loss of the two-market model. We supported these theoretical results with extensive simulation. We also looked at competing *kidney exchanges*, and provided (to our knowledge) the first experimental quantification of the loss in global welfare in a setting with two clearinghouses using realistic kidney exchange data drawn from a generator due to Saidman et al. [191] and another based on the United Network for Organ Sharing (UNOS) program.

We see competing dynamic matching markets as fertile ground for future research, with a trove of both theoretical and practical questions to answer. First, the model of Akbarpour et al. [9] discounts the utility of a match by the time the matching agent has already waited in the pool; this is well motivated in a variety of settings, including kidney exchange. Our results in this chapter assume a discount factor of zero, so it would be valuable to consider the impact on discounted loss for non-zero cases. Second, in our model the choice of market to enter is exogenously determined

for each agent. In reality, agents with different levels of knowledge, wealth, etc. may make strategic decisions on which markets to enter. Thus, one could approach this dynamic matching problem from a game-theoretic point of view. Similarly, taking network effects (where more popular exchanges have an easier time attracting agents, lower operating costs, higher probabilities of two agents forming an acceptable transaction, and other advantages) into account would make these models more applicable to many real-world settings. Finally, we only looked at two overlapping markets; generalizing this to any number of overlapping markets would also be of interest.

In terms of barter exchange and, specifically, kidney exchange, the question of how clearinghouses interact is a timely one. In the United States and, eventually, elsewhere, multi-center and single-center exchange clearinghouses are already competing, each drawing from some (often overlapping) subset of the full set of patient-donor pairs available. Indeed, the dynamic barter exchange problem *in a single market* is still not fully understood (barring very promising recent work due to Anderson et al. [15]). We saw in Section E.5.2 that including 3-cycles in the matching process results in lower loss, even when two markets overlap, compared to including only 2-cycles (a result that has been shown repeatedly in the static [188] and dynamic [15] single clearinghouse setting), so extending the theoretical underpinnings of our framework to a more general setting would be of great value. Finally, it is curious that the UNIFORM3 policy had such a large effect on the loss of the individual Patient and Greedy exchanges compared to the UNIFORM policy; further exploration of different matching policies (including those that use a strong prior to consider possible future states of the pool when matching now) would be helpful in making policy recommendations to fielded exchanges.

Bibliography

[1] A. Abdulkadiroğlu, P. A. Pathak, and A. E. Roth. The New York City high school match. *American Economic Review*, pages 364–367, 2005. 2

[2] A. Abdulkadiroğlu, P. A. Pathak, A. E. Roth, and T. Sönmez. The Boston public school match. *American Economic Review*, pages 368–371, 2005. 2

[3] A. Abdulkadiroğlu and S. Loertscher. Dynamic house allocations, 2007. Working paper. 246, 247

[4] A. Abdulkadiroğlu and T. Sönmez. School choice: A mechanism design approach. *The American Economic Review*, 93(3):729–747, 2003. 2

[5] A. Abdulkadiroğlu and T. Sönmez. Matching markets: Theory and practice. *Advances in Economics and Econometrics*, 1:3–47, 2013. 2

[6] D. Abraham, A. Blum, and T. Sandholm. Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 295–304, 2007. 5, 7, 10, 32, 33, 34, 41, 45, 48, 49, 53, 54, 58, 60, 85, 86, 113, 140, 181, 198

[7] M. Adamczyk. Improved analysis of the greedy algorithm for stochastic matching. *Information Processing Letters*, 111(15):731–737, 2011. 72

[8] M. Akan, O. Alagoz, B. Ata, F. S. Erenay, and A. Said. A broader view of designing the liver allocation system. *Operations Research*, 60(4):757–770, 2012. 113

[9] M. Akbarpour, S. Li, and S. O. Gharan. Dynamic matching market design. In *Proceedings of the ACM Conference on Economics and Computation (EC)*, page 355, 2014. 5, 12, 73, 74, 109, 111, 112, 182, 195, 246, 247, 248, 249, 252, 253, 255, 259

[10] M. Akbarpour, A. Nikzad, and A. Roth. Financing transplants' costs of the poor: A dynamic model of international kidney exchange, 2016. Working paper. 4, 85, 199

[11] O. Alagoz, L. M. Maillart, A. J. Schaefer, and M. S. Roberts. Determining the acceptance of cadaveric livers using an implicit model of the waiting list. *Operations Research*, 55(1): 24–36, 2007. 113

[12] F. Alvelos, X. Klimentova, A. Rais, and A. Viana. A compact formulation for maximizing the expected number of transplants in kidney exchange programs. In *Journal of Physics: Conference Series*, volume 616. IOP Publishing, 2015. 41

[13] R. Amadini. *Portfolio approaches in constraint programming*. PhD thesis, Università di

Bologna, 2015. 198

[14] R. Anderson. *Stochastic models and data driven simulations for healthcare operations*. PhD thesis, Massachusetts Institute of Technology, 2014. 5, 10, 55, 68, 73, 177

[15] R. Anderson, I. Ashlagi, D. Gamarnik, and Y. Kanoria. A dynamic model of barter exchange. In *Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1925–1933, 2015. 5, 15, 73, 74, 109, 112, 182, 195, 247, 260

[16] R. Anderson, I. Ashlagi, D. Gamarnik, and A. E. Roth. Finding long chains in kidney exchange using the traveling salesman problem. *Proceedings of the National Academy of Sciences*, 112(3):663–668, 2015. 5, 10, 32, 41, 45, 46, 49, 53, 73, 107, 193, 194, 198

[17] E. Anshelevich, M. Chhabra, S. Das, and M. Gerrior. On the social welfare of mechanisms for repeated batch matching. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 60–66, 2013. 74, 111, 112, 182, 247

[18] N. Arnosti, R. Johari, and Y. Kanoria. Managing congestion in decentralized matching markets. In *Proceedings of the ACM Conference on Economics and Computation (EC)*, page 451, 2014. 247

[19] I. Ashlagi and A. E. Roth. Free riding and participation in large scale, multi-hospital kidney exchange. *Theoretical Economics*, 9(3):817–863, 2014. xx, 5, 8, 11, 15, 18, 19, 20, 21, 22, 74, 107, 134, 135, 137, 164, 165, 171, 172, 173, 174, 196, 241

[20] I. Ashlagi, F. Fischer, I. Kash, and A. D. Procaccia. Mix and match. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 305–314, 2010. 18

[21] I. Ashlagi, M. Tennenholtz, and A. Zohar. Competing schedulers. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2010. 247

[22] I. Ashlagi, D. S. Gilchrist, A. E. Roth, and M. Rees. Nonsimultaneous chains and dominos in kidney-paired donation—revisited. *American Journal of Transplantation*, 11(5):984–994, 2011. 6, 26, 27, 71, 95, 97, 99, 112, 177, 242

[23] I. Ashlagi, D. Gamarnik, M. Rees, and A. E. Roth. The need for (long) chains in kidney exchange. NBER Working Paper No. 18202, July 2012. 5, 9, 11, 15, 17, 19, 22, 74, 77, 81, 82, 107, 112, 144, 165, 166, 167, 168, 169, 170, 171, 180

[24] I. Ashlagi, P. Jaillet, and V. H. Manshadi. Kidney exchange in dynamic sparse heterogenous pools. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 25–26, 2013. 5, 73, 74, 109, 111, 112, 144, 171, 180, 182, 194, 195, 247

[25] I. Ashlagi, F. Fischer, I. A. Kash, and A. D. Procaccia. Mix and match: A strategyproof mechanism for multi-hospital kidney exchange. *Games and Economic Behavior*, 91:284–296, 2015. 5, 164, 196

[26] I. Ashlagi, M. Burq, P. Jaillet, and V. Manshadi. On matching and thickness in heterogeneous dynamic markets. In *Proceedings of the ACM Conference on Economics and Computation (EC)*, 2016. 73, 74, 112

[27] P. Awasthi and T. Sandholm. Online stochastic optimization in the large: Application to kidney exchange. In *Proceedings of the 21st International Joint Conference on Artificial*

Intelligence (IJCAI), pages 405–411, 2009. 55, 68, 74, 111, 112, 182, 247

[28] E. Balas. The prize collecting traveling salesman problem. *Networks*, 19(6):621–636, 1989. 32

[29] N. Bansal, A. Gupta, J. Li, J. Mestre, V. Nagarajan, and A. Rudra. When LP is the cure for your matching woes: Improved bounds for stochastic matchings. *Algorithmica*, 63(4): 733–762, 2012. 72

[30] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998. 33, 140

[31] BarterQuest. URL http://www.barterquest.com. [Online; accessed 18-June-2016]. 3

[32] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust optimization*. Princeton University Press, 2009. 193

[33] D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004. 194

[34] D. Bertsimas, D. B. Brown, and C. Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011. 193, 194

[35] D. Bertsimas, V. F. Farias, and N. Trichakis. The price of fairness. *Operations Research*, 59 (1):17–31, 2011. 11, 96, 132, 133

[36] D. Bertsimas, V. F. Farias, and N. Trichakis. On the efficiency-fairness trade-off. *Management Science*, 58(12):2234–2250, 2012. 96

[37] D. Bertsimas, V. F. Farias, and N. Trichakis. Fairness, efficiency, and flexibility in organ allocation for kidney transplantation. *Operations Research*, 61(1):73–87, 2013. 3, 73, 96, 113, 137, 198, 246

[38] Best House Swap. URL http://besthouseswap.com. [Online; accessed 18-June-2016]. 3

[39] A. Biere. Yet another local search solver and lingeling and friends entering the SAT Competition 2014. *SAT Competition*, 2014:2, 2014. 65

[40] P. Biró and K. Cechlárová. Inapproximability of the kidney exchange problem. *Information Processing Letters*, 101(5):199–202, 2007. 7

[41] P. Biró, D. F. Manlove, and R. Rizzi. Maximum weight cycle packing in directed graphs, with application to kidney exchange programs. *Discrete Mathematics, Algorithms and Applications*, 1(04):499–517, 2009. 5, 7, 58

[42] F. Bloch and D. Cantala. Markovian assignment rules. *Social Choice and Welfare*, 40(1): 1–25, 2013. 246, 247

[43] A. Blum, A. Gupta, A. D. Procaccia, and A. Sharma. Harnessing the power of two crossmatches. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 123–140, 2013. 5, 19, 73, 194

[44] A. Blum, J. P. Dickerson, N. Haghtalab, A. D. Procaccia, T. Sandholm, and A. Sharma. Ignorance is almost bliss: Near-optimal stochastic matching with few queries. In *Proceedings of the ACM Conference on Economics and Computation (EC)*, pages 325–342, 2015. 5, 10, 72, 73, 104, 105, 110, 194

[45] A. Blum, I. Caragiannis, N. Haghtalab, A. D. Procaccia, E. B. Procaccia, and R. Vaish. Opting into optimal matching, 2016. Working paper. 5, 196

[46] P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, et al. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5(2):186–204, 2008. 64

[47] S. J. Bradtke and A. G. Barto. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22(1-3):33–57, 1996. 113

[48] M. Bray, W. Wang, P. X.-K. Song, A. B. Leichtman, M. A. Rees, V. B. Ashby, R. Eikstadt, A. Goulding, and J. D. Kalbfleisch. Planning for uncertainty and fallbacks can increase the number of transplants in a kidney-paired donation program. *American Journal of Transplantation*, 2015. 71

[49] S. Bronfman, N. Alon, A. Hassidim, and A. Romm. Redesigning the israeli medical internship match. In *Proceedings of the ACM Conference on Economics and Computation (EC)*, pages 753–754. ACM, 2015. 2

[50] R. S. Brown. Live donors in liver transplantation. *Gastroenterology*, 134(6):1802–1813, 2008. 181

[51] E. B. Budish, P. Cramton, and J. J. Shim. The high-frequency trading arms race: Frequent batch auctions as a market design response, 2015. Working paper. 247

[52] R. Burguet and J. Sákovics. Imperfect competition in auction designs. *International Economic Review*, 40(1):231–247, 1999. 247

[53] I. Caragiannis, C. Kaklamanis, P. Kanellopoulos, and M. Kyropoulou. The efficiency of fair division. International Workshop on Internet and Network Economics (WINE), 2009. 11, 96, 133

[54] I. Caragiannis, A. Filos-Ratsikas, and A. D. Procaccia. An improved 2-agent kidney exchange mechanism. *Theoretical Computer Science*, 589:53–60, 2015. 18, 164

[55] M. Carvalho, A. Lodi, J. P. Pedroso, and A. Viana. Nash equilibria in the two-player kidney exchange game. *Mathematical Programming*, pages 1–29, 2016. 5, 196

[56] J. M. Cecka. The UNOS scientific renal transplant registry–ten years of kidney transplants. *Clinical transplants*, pages 1–14, 1996. 3

[57] S. C. Chan, C. M. Lo, B. H. Yong, W. J. Tsui, K. K. Ng, and S. T. Fan. Paired donor interchange to avoid ABO-incompatible living donor liver transplantation. *Liver Transplantation*, 16(4):478–481, 2010. 161

[58] Y. L. Cheah, M. A. Simpson, J. J. Pomposelli, and E. A. Pomfret. Incidence of death and potentially life-threatening near-miss events in living donor hepatic lobectomy: A worldwide survey. *Liver Transplantation*, 19(5):499–506, 2013. 161, 162

[59] P. Cheeseman, B. Kanefsky, and W. Taylor. Where the really hard problems are. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 331–337, Sydney, Australia, 1991. 65

[60] N. Chen, N. Immorlica, A. R. Karlin, M. Mahdian, and A. Rudra. Approximating matches made in heaven. In *Proceedings of the International Conference on Automata, Languages, and Programming (ICALP)*, pages 266–278, 2009. 72

[61] Y. Chen, Y. Li, J. D. Kalbfleisch, Y. Zhou, A. Leichtman, and P. X.-K. Song. Graph-based optimization algorithm and software on kidney exchanges. *IEEE Transactions on Biomedical Engineering*, 59:1985–1991, 2012. 71

[62] M. S. Chung and D. B. West. The p-intersection number of a complete bipartite graph and orthogonal double coverings of a clique. *Combinatorica*, 14(4):453–461, 1994. ISSN 0209-9683. 61

[63] M. Constantino, X. Klimentova, A. Viana, and A. Rais. New insights on integer-programming models for the kidney exchange problem. *European Journal of Operational Research*, 231(1):57–68, 2013. 5, 10, 32, 41, 43, 53, 107, 177, 198

[64] K. P. Costello, P. Tetali, and P. Tripathi. Matching with commitment. In *Proceedings of the International Conference on Automata, Languages, and Programming (ICALP)*, pages 822–833, 2012. 72

[65] S. Das, J. P. Dickerson, Z. Li, and T. Sandholm. Competing dynamic matching markets. In *Proceedings of the Conference on Auctions, Market Mechanisms and Their Applications (AMMA)*, 2015. 112, 245

[66] Datatang Technology Company. URL `http://datatang.com`. [Online; accessed 18-June-2016]. 3

[67] M. De Klerk, K. M. Keizer, F. H. Claas, M. Witvliet, B. J. Haase-Kromwijk, and W. Weimar. The Dutch national living donor kidney exchange program. *American Journal of Transplantation*, 5(9):2302–2305, 2005. 4

[68] M. De Klerk, J. Kal-van Gestel, B. Haase-Kromwijk, F. Claas, and W. Weimar. Eight years of outcomes of the Dutch living donor kidney exchange program. *Clinical Transplants*, pages 287–290, 2010. 194

[69] D. Delen, G. Walker, and A. Kadam. Predicting breast cancer survivability: A comparison of three data mining methods. *Artificial Intelligence in Medicine*, 34(2):113–127, June 2005. 150

[70] J. P. Dickerson and T. Sandholm. FutureMatch: Combining human value judgments and machine learning to match in dynamic environments. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 622–628, 2015. 11, 55, 68, 148, 182, 195, 247

[71] J. P. Dickerson and T. Sandholm. Uncertainty in dynamic matching with application to organ exchange. In *Machine Learning for Healthcare (MLHC) workshop at NIPS-2015*, 2015. 11

[72] J. P. Dickerson and T. Sandholm. Multi-organ exchange: The whole is greater than the sum

of its parts. *Journal of Artificial Intelligence Research*, 2016. To appear. 12, 68, 85, 162, 198, 199

[73] J. P. Dickerson, A. D. Procaccia, and T. Sandholm. Dynamic matching via weighted myopia with application to kidney exchange. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 1340–1346, 2012. 10, 55, 68, 182, 247

[74] J. P. Dickerson, A. D. Procaccia, and T. Sandholm. Optimizing kidney exchange with transplant chains: Theory and reality. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 711–718, 2012. 9, 16, 29, 33, 82, 107, 112, 171

[75] J. P. Dickerson, A. D. Procaccia, and T. Sandholm. Failure-aware kidney exchange. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 323–340, 2013. 5, 6, 7, 10, 15, 19, 33, 41, 72, 112, 113, 142, 171, 177, 180, 194

[76] J. P. Dickerson, A. D. Procaccia, and T. Sandholm. Results about, and algorithms for, robust probabilistic kidney exchange matching. In *American Transplant Congress (ATC)*, 2013. Poster abstract. 72

[77] J. P. Dickerson, A. D. Procaccia, and T. Sandholm. Empirical price of fairness in failure-aware kidney exchange. In *Towards Better and more Affordable Healthcare: Incentives, Game Theory, and Artificial Intelligence (HCAGT) workshop at AAMAS-2014*, 2014. 11, 132, 180

[78] J. P. Dickerson, A. D. Procaccia, and T. Sandholm. Price of fairness in kidney exchange. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 1013–1020, 2014. 10, 11, 18, 132, 171, 177

[79] J. P. Dickerson, A. M. Kazachkov, A. D. Procaccia, and T. Sandholm. Small representations of big kidney exchange graphs. *CoRR*, abs/1605.07728, 2016. 9, 57

[80] J. P. Dickerson, D. Manlove, B. Plaut, T. Sandholm, and J. Trimble. Position-indexed formulations for kidney exchange. In *Proceedings of the ACM Conference on Economics and Computation (EC)*, 2016. 7, 9, 10, 41, 194, 198

[81] J. P. Dickerson, D. Manlove, B. Plaut, T. Sandholm, and J. Trimble. Position-indexed formulations for kidney exchange. *CoRR*, abs/1606.01623, 2016. 41

[82] Y. Ding, D. Ge, S. He, and C. Ryan. A non-asymptotic approach to analyzing kidney exchange graphs. In *Proceedings of the ACM Conference on Economics and Computation (EC)*, pages 257–258, 2015. 5, 19

[83] N. Eaton, R. J. Gould, and V. Rödl. On p-intersection representations. *J. Graph Theory*, 21(4):377–392, 1996. ISSN 0364-9024. doi: 10.1002/(SICI)1097-0118(199612)23:4<377::AID-JGT7>3.0.CO;2-O. URL http://dx.doi.org/10.1002/(SICI)1097-0118(199612)23:4<377::AID-JGT7>3.0.CO;2-O. 61

[84] J. Edmonds. Maximum matching and a polyhedron with 0,1 vertices. *J. Res. Nat. Bur. Standards*, B(69):125–130, 1965. 87

[85] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965. 31

[86] H. Egawa, F. Oike, L. Buhler, A. Shapiro, S. Minamiguchi, H. Haga, K. Uryuhara, T. Kiuchi, S. Kaihara, and K. Tanaka. Impact of recipient age on outcome of ABO-incompatible living-donor liver transplantation. *Transplantation*, 77(3):403, 2004. 176

[87] T. Eisenmann, G. Parker, and M. W. Van Alstyne. Strategies for two-sided markets. *Harvard Business Review*, 84(10):92, 2006. 247

[88] P. Erdős and A. Rényi. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5:17–61, 1960. 77, 248

[89] P. Erdős, A. W. Goodman, and L. Pósa. The representation of a graph by set intersections. *Canad. J. Math.*, 18:106–112, 1966. ISSN 0008-414X. 61

[90] H. Ergin, T. Sönmez, and M. U. Ünver. Lung exchange, 2014. Working paper. 68, 183, 186, 199

[91] H. Ergin, T. Sönmez, and M. U. Ünver. Liver exchange, 2015. Working paper. 68, 199

[92] Factual. URL https://www.factual.com. [Online; accessed 18-June-2016]. 3

[93] W. Fang, P. Tang, and S. Zuo. Digital good exchange. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 1277–1278, 2016. 3, 199

[94] D. Fumo, V. Kapoor, L. Reece, S. Stepkowski, J. Kopke, S. Rees, C. Smith, A. Roth, A. Leichtman, and M. Rees. Historical matching strategies in kidney paired donation: The 7-year evolution of a web-based virtual matching system. *American Journal of Transplantation*, 15(10):2646–2654, 2015. 194

[95] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962. 2

[96] S. E. Gentry and D. L. Segev. The honeymoon phase and studies of nonsimultaneous chains in kidney-paired donation. *American Journal of Transplantation*, 11(12):2778–2779, 2011. 6, 112

[97] S. E. Gentry, R. A. Montgomery, B. J. Swihart, and D. L. Segev. The roles of dominos and nonsimultaneous chains in kidney paired donation. *American Journal of Transplantation*, 9 (6):1330–1336, 2009. 26, 27, 112

[98] K. Glorie, M. Carvalho, M. Constantino, P. Bouman, and A. Viana. Robust models for the kidney exchange problem, 2015. Working paper. 55, 68, 73, 194

[99] K. M. Glorie. Estimating the probability of positive crossmatch after negative virtual crossmatch. Technical report, Erasmus School of Economics, 2012. 95, 109, 186, 194

[100] K. M. Glorie, J. J. van de Klundert, and A. P. M. Wagelmans. Kidney exchange with long chains: An efficient pricing algorithm for clearing barter exchanges with branch-and-price. *Manufacturing & Service Operations Management (MSOM)*, 16(4):498–512, 2014. 5, 10, 33, 34, 35, 36, 41, 48, 49, 51, 54, 55, 82, 107, 207, 208, 209

[101] G. Goel and P. Tripathi. Matching with our eyes closed. In *Symposium on the Foundations of Computer Science (FOCS)*, pages 718–727. IEEE, 2012. 72

[102] A. Goldberg, J. Hartline, and A. Wright. Competitive auctions and digital goods. In *Annual*

ACM-SIAM Symposium on Discrete Algorithms (SODA), Washington, DC, 2001. 199

[103] A. Gupta and V. Nagarajan. A stochastic probing problem with applications. In *Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 205–216, 2013. 73

[104] C. Hajaj, J. P. Dickerson, A. Hassidim, T. Sandholm, and D. Sarne. Strategy-proof and efficient kidney exchange using a credit mechanism. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 921–928, 2015. 5, 196

[105] F. Harary, J. A. Kabell, and F. R. McMorris. Bipartite intersection graphs. *Comment. Math. Univ. Carolin.*, 23(4):739–745, 1982. ISSN 0010-2628. 61

[106] A. Hart, J. M. Smith, M. A. Skeans, S. K. Gustafson, D. E. Stewart, W. S. Cherikh, J. L. Wainright, G. Boyle, J. J. Snyder, B. L. Kasiske, and A. K. Israni. Kidney. *American Journal of Transplantation (Special Issue: OPTN/SRTR Annual Data Report 2014)*, 16, Issue Supplement S2:11–46, 2016. ISSN 1600-6143. 3

[107] HHS/HRSA/HSB/DOT. OPTN/SRTR annual data report, 2011. 131, 182

[108] T. Hogg, B. A. Huberman, and C. P. Williams. Phase transitions and the search problem. *Artificial Intelligence*, 81(1-2):1–15, 1996. 65

[109] J. N. Hooker. Logic, optimization and constraint programming. *INFORMS Journal of Computing*, 14:295–321, 2002. 65

[110] J. N. Hooker and H. P. Williams. Combining equity and utilitarianism in a mathematical programming model. *Management Science*, 58(9):1682–1693, 2012. 96, 145, 197, 198

[111] H. Hosseini, K. Larson, and R. Cohen. Matching with dynamic ordinal preferences. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2015. 247

[112] J. Hsu, Z. Huang, A. Roth, T. Roughgarden, and Z. S. Wu. Private matchings and allocations. In *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, pages 21–30. ACM, 2014. 197

[113] F. Hutter, H. Hoos, K. Leyton-Brown, and T. Stützle. ParamILS: An automatic algorithm configuration framework. *Journal of Artificial Intelligence Research*, 36(1):267–306, 2009. ISSN 1076-9757. 10, 111, 114, 120

[114] F. Hutter, H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Proc. of LION-5*, pages 507–523, 2011. 10, 120, 153

[115] S. Hwang, S.-G. Lee, D.-B. Moon, G.-W. Song, C.-S. Ahn, K.-H. Kim, T.-Y. Ha, D.-H. Jung, K.-W. Kim, N.-K. Choi, G.-C. Park, Y.-D. Yu, Y.-I. Choi, P.-J. Park, and H.-S. Ha. Exchange living donor liver transplantation to overcome ABO incompatibility in adult patients. *Liver Transplantation*, 16(4):482–490, 2010. ISSN 1527-6473. doi: 10.1002/lt.22017. URL http://dx.doi.org/10.1002/lt.22017. 161

[116] A. Hylland and R. Zeckhauser. The efficient allocation of individuals to positions. *The Journal of Political Economy*, pages 293–314, 1979. 3

[117] IBM ILOG Inc. CPLEX 12.2 User's Manual, 2010. 90

[118] IBM ILOG Inc. CPLEX 12.6 User's Manual, 2015. 65

[119] Intervac. URL http://intervac-homeexchange.com. [Online; accessed 18-June-2016]. 3

[120] S. Janson, T. Luczak, and A. Rucinski. *Random Graphs*. Wiley Series in Discrete Mathematics and Optimization. Wiley, 2011. ISBN 9781118030967. 78, 166, 167, 174

[121] W. S. Jevons. *Money and the Mechanism of Exchange*, volume 17. Appleton, London, 1885. 1

[122] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998. 105, 233

[123] S. V. Kadam and M. H. Kotowski. Multi-period matching, 2014. Working paper. 247

[124] A. Kahng. Timing objectives in dynamic kidney exchange. Harvard University Senior Honors Thesis, 2016. Undergraduate thesis. 158

[125] S. Kannan, J. Morgenstern, R. Rogers, and A. Roth. Private pareto optimal exchange. In *Proceedings of the ACM Conference on Economics and Computation (EC)*, pages 261–278. ACM, 2015. 197

[126] S. Kannan, J. Morgenstern, A. Roth, and Z. S. Wu. Approximately stable, school optimal, and student-truthful many-to-one matchings (via differential privacy). In *Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1890–1903. SIAM, 2015. 197

[127] E. H. Kaplan. *Managing the demand for public housing*. PhD thesis, Massachusetts Institute of Technology, 1984. 3, 246, 247

[128] E. H. Kaplan. Tenant assignment models. *Operations Research*, 34(6):832–843, 1986. 3, 246, 247

[129] R. M. Karp, U. V. Vazirani, and V. V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, pages 352–358, 1990. 247

[130] J. Kennes, D. Monte, and N. Tumennasan. The day care assignment: A dynamic matching problem. *American Economic Journal: Microeconomics*, 6(4):362–406, 2014. 247

[131] Kidney Paired Donation Work Group. OPTN KPD pilot program cumulative match report (CMR) for KPD match runs: Oct 27, 2010 – Nov 12, 2012, 2012. 71, 80

[132] Kidney Paired Donation Work Group. OPTN KPD pilot program cumulative match report (CMR) for KPD match runs: Oct 27, 2010 – Apr 15, 2013, 2013. 80, 95, 144, 154

[133] Kidney Transplantation Committee. OPTN concepts for kidney allocation, 2011. 151

[134] X. Klimentova, F. Alvelos, and A. Viana. A new branch-and-price approach for the kidney exchange problem. In *Computational Science and Its Applications (ICCSA-2014)*, pages 237–252. Springer, 2014. 5, 10, 33, 41, 50, 51, 177, 198

[135] X. Klimentova, J. P. Pedroso, and A. Viana. Maximising expectation of the number of transplants in kidney exchange programmes. *Computers & Operations Research*, 73:1–11, 2016. 73

[136] L. Kocsis and C. Szepesvári. Bandit based monte-carlo planning. In *European Conference on Maching Learning (ECML)*, pages 282–293. Springer, 2006. 194

[137] L. T. Kou, L. J. Stockmeyer, and C. K. Wong. Covering edges by cliques with regard to keyword conflicts and intersection graphs. *Comm. ACM*, 21(2):135–139, 1978. ISSN 0001-0782. 62

[138] M. Krivelevich, E. Lubetzky, and B. Sudakov. Longest cycles in sparse random digraphs. *Random Struct. Algorithms*, 43(1):1–15, 2013. 170

[139] M. Kurino. *Essays on dynamic matching markets*. PhD thesis, University of Pittsburgh, 2009. 246, 247

[140] J. Kwak, O. Kwon, K. S. Lee, C. M. Kang, H. Y. Park, and J. Kim. Exchange-donor program in renal transplantation: a single-center experience. In *Transplantation Proceedings*, volume 31, pages 344–345. Elsevier, 1999. 4

[141] M. G. Lagoudakis and R. Parr. Least-squares policy iteration. *Journal of Machine Learning Research*, 4:1107–1149, 2003. 113

[142] S. Lange and M. Riedmiller. Deep auto-encoder neural networks in reinforcement learning. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2010. 128

[143] R. Leishman, R. Formica, K. Andreoni, J. Friedewald, E. Sleeman, C. Monstello, D. Stewart, and T. Sandholm. The Organ Procurement and Transplantation Network (OPTN) Kidney Paired Donation Pilot Program (KPDPP): Review of current results. In *American Transplant Congress (ATC)*, 2013. Talk abstract. 53, 71, 80

[144] R. Leishman, D. Stewart, A. Kucheryavaya, L. Callahan, T. Sandholm, and M. Aeder. Reasons for match offer refusals and efforts to reduce them in the OPTN/UNOS kidney paired donation pilot program (KPDPP). In *American Transplant Congress (ATC)*, 2015. Talk abstract. 194

[145] J. D. Leshno. Dynamic matching in overloaded waiting lists, 2015. Working paper. 246, 247

[146] J. Li, Y. Liu, L. Huang, and P. Tang. Egalitarian pairwise kidney exchange: Fast algorithms via linear programming and parametric flow. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 445–452, 2014. 131, 145, 198

[147] Y. Li, J. Kalbfleisch, P. X. Song, Y. Zhou, A. Leichtman, and M. Rees. Optimization and simulation of an evolving kidney paired donation (KPD) program. Department of biostatistics working paper 90, University of Michigan, May 2011. 6, 10, 71

[148] Y. Liu, P. Tang, and W. Fang. Internally stable matchings and exchanges. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 1433–1439, 2014. 197

[149] S. Luo and P. Tang. Mechanism design and implementation for lung exchange. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 209–215, 2015. 68, 186

[150] S. Luo, P. Tang, C. Wu, and J. Zeng. Approximation of barter exchanges with cycle length

constraints. *CoRR*, abs/1605.08863, 2016. 7

[151] V. Mak-Hau. On the kidney exchange problem: cardinality constrained cycle and chain problems on directed graphs: a survey of integer programming approaches. *Journal of Combinatorial Optimization*, pages 1–25, 2015. ISSN 1382-6905. 31, 32, 198

[152] D. Manlove and G. O'Malley. Paired and altruistic kidney donation in the UK: Algorithms and experimentation. *ACM Journal of Experimental Algorithmics*, 19(1), 2015. 5, 6, 41, 194

[153] D. F. Manlove. *Algorithmics of matching under preferences*, volume 2. World Scientific, 2013. 2

[154] N. Mattei and T. Walsh. Preflib: A library for preferences. In *Algorithmic Decision Theory*, volume 8176 of *Lecture Notes in Computer Science*, pages 259–270. Springer, 2013. 45

[155] M. McDowell, C. Fryar, C. Ogden, and K. Flegal. Anthropometric reference data for children and adults: United States, 2003–2006. *Nutrition*, 10(10):1–45, 2008. 177, 240

[156] T. A. McKee and F. R. McMorris. *Topics in intersection graph theory*. SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1999. ISBN 0-89871-430-3. 61

[157] N. Meadem, N. Verbiest, K. Zolfaghar, J. Agarwal, S.-C. Chin, and S. B. Roy. Exploring preprocessing techniques for prediction of risk of readmission for congestive heart failure patients. In *Data Mining and Healthcare (DMH), at International Conference on Knowledge Discovery and Data Mining (KDD)*, 2013. 150

[158] A. Mehta. Online matching and ad allocation. *Theoretical Computer Science*, 8(4):265–368, 2012. 247

[159] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. Adwords and generalized online matching. *Journal of the ACM*, 54(5), 2007. 247

[160] C. E. Miller, A. W. Tucker, and R. A. Zemlin. Integer programming formulation of traveling salesman problems. *Journal of the ACM*, 7(4):326–329, 1960. 32

[161] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. 128, 194

[162] D. Monderer and M. Tennenholtz. K-price auctions: Revenue inequalities, utility equivalence, and competition in auction design. *Economic Theory*, 24(2):255–270, 2004. 247

[163] R. Montgomery, S. Gentry, W. H. Marks, D. S. Warren, J. Hiller, J. Houp, A. A. Zachary, J. K. Melancon, W. R. Maley, H. Rabb, C. Simpkins, and D. L. Segev. Domino paired kidney donation: a strategy to make best use of live non-directed donation. *The Lancet*, 368 (9533):419–421, 2006. 16

[164] National Odd Shoe Exchange. URL `http://www.oddshoe.org`. [Online; accessed 18-June-2016]. 3

[165] B. L. Neuen, G. E. Taylor, A. R. Demaio, and V. Perkovic. Global kidney disease. *The*

Lancet, 382(9900):1243, 2013. 3

[166] M. Niederle, A. Roth, and T. Sönmez. Matching. In S. Durlauf and L. Blume, editors, *The New Palgrave Dictionary of Economics*. Palgrave Macmillan, 2008. 2

[167] M. Niederle, A. E. Roth, and M. U. Ünver. Unraveling results from comparable demand and supply: An experimental investigation. *Games*, 4(2):243–282, 2013. 197

[168] G. Opelz. Correlation of HLA matching with kidney graft survival in patients with or without cyclosporine treatment: for the collaborative transplant study. *Transplantation*, 40(3): 240–242, 1985. 147, 151, 152

[169] Organize. URL https://organize.org. [Online; accessed 19-June-2016]. 196

[170] J. Orlin. Contentment in graph theory: covering graphs with cliques. *Nederl. Akad. Wetensch. Proc. Ser. A*, 39(5):406–424, 1977. 61, 62

[171] M. Ostrovsky. Stability in supply chain networks. *American Economic Review*, pages 897–923, 2008. 247

[172] J. a. P. Pedroso. Maximizing expectation on vertex-disjoint cycle packing. In *Computational Science and Its Applications (ICCSA-2014)*, pages 32–46. Springer, 2014. 41

[173] J. P. Pedroso and S. Ikeda. Maximum-expectation matching under recourse. *CoRR*, abs/1605.08616, 2016. 73

[174] B. Plaut, J. P. Dickerson, and T. Sandholm. Fast optimal clearing of capped-chain barter exchanges. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 601–607, 2016. xvii, 9, 10, 33, 34, 35, 36, 37, 41, 48, 49, 54, 207, 208

[175] B. Plaut, J. P. Dickerson, and T. Sandholm. Hardness of the pricing problem for chains in barter exchanges. *CoRR*, abs/1606.00117, 2016. 9, 34, 207

[176] F. T. Rapaport. The case for a living emotionally related international kidney donor exchange registry. *Transplantation Proceedings*, 18:5–9, 1986. 4

[177] Read It Swap It. URL http://http://www.readitswapit.co.uk. [Online; accessed 18-June-2016]. 3

[178] M. Rees, J. Kopke, R. Pelletier, D. Segev, M. Rutter, A. Fabrega, J. Rogers, O. Pankewycz, J. Hiller, A. Roth, T. Sandholm, U. Ünver, and R. Montgomery. A nonsimultaneous, extended, altruistic-donor chain. *New England Journal of Medicine*, 360(11):1096–1101, 2009. 4, 6, 16, 241

[179] L. Ross, D. Rubin, M. Siegler, M. Josephson, J. Thistlethwaite, and S. Woodle. Ethics of a paired-kidney-exchange program. *New England Journal of Medicine*, 336(24):1752–5, 1997. 4

[180] A. Roth. The evolution of the labor market for medical interns and residents: a case study in game theory. *The Journal of Political Economy*, pages 991–1016, 1984. 2

[181] A. Roth. Repugnance as a constraint on markets. *Journal of Economic Perspectives*, 21(3): 37–58, 2007. 1

[182] A. Roth. *Who Gets What and Why: The New Economics of Matchmaking and Market*

Design. Eamon Dolan/Houghton Mifflin Harcourt, 2015. 2

[183] A. Roth and M. Sotomayor. Two-sided matching. *Handbook of game theory with economic applications*, 1:485–541, 1992. 2

[184] A. Roth, T. Sönmez, and U. Ünver. Kidney exchange. *Quarterly Journal of Economics*, 119 (2):457–488, 2004. 4, 5, 24, 58, 247

[185] A. Roth, T. Sönmez, and U. Ünver. A kidney exchange clearinghouse in New England. *American Economic Review*, 95(2):376–380, 2005. 4, 58

[186] A. Roth, T. Sönmez, and U. Ünver. Pairwise kidney exchange. *Journal of Economic Theory*, 125(2):151–188, 2005. 4, 5, 31, 58, 96, 131, 145

[187] A. Roth, T. Sönmez, U. Ünver, F. Delmonico, and S. L. Saidman. Utilizing list exchange and nondirected donation through 'chain' paired kidney donations. *American Journal of Transplantation*, 6:2694–2705, 2006. 4, 16

[188] A. Roth, T. Sönmez, and U. Ünver. Efficient kidney exchange: Coincidence of wants in a market with compatibility-based preferences. *American Economic Review*, 97:828–851, 2007. 5, 7, 31, 32, 41, 45, 260

[189] A. E. Roth and X. Xing. Jumping the gun: Imperfections and institutions related to the timing of market transactions. *American Economic Review*, pages 992–1044, 1994. 197

[190] I. O. Ryzhov and W. B. Powell. Information collection for linear programs with uncertain objective coefficients. *SIAM Journal on Optimization*, 22(4):1344–1368, 2012. 194

[191] S. L. Saidman, A. Roth, T. Sönmez, U. Ünver, and F. Delmonico. Increasing the opportunity of live kidney donation by matching for two and three way exchanges. *Transplantation*, 81 (5):773–782, 2006. xiii, 8, 15, 18, 23, 27, 29, 45, 71, 90, 94, 99, 105, 106, 116, 121, 143, 176, 177, 178, 237, 240, 256, 259

[192] D. Segev, S. Gentry, D. S. Warren, B. Reeb, and R. A. Montgomery. Kidney paired donation and optimizing the use of live donor organs. *Journal of the American Medical Association*, 293(15):1883–1890, 2005. 31

[193] D. L. Segev and R. A. Montgomery. The application of paired donation to live donor liver transplantation. *Liver Transplantation*, 16(4):423–425, 2010. 161

[194] M. Sen, S. Das, A. B. Roy, and D. B. West. Interval digraphs: an analogue of interval graphs. *J. Graph Theory*, 13(2):189–202, 1989. ISSN 0364-9024. 61, 62

[195] L. Shapley and H. Scarf. On cores and indivisibility. *Journal of Mathematical Economics*, 1(1):23–37, 1974. 3

[196] Y. Shoham and K. Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008. 247

[197] T. Sönmez and M. U. Ünver. Matching, allocation, and exchange of discrete resources. *Handbook of Social Economics*, 1:781–852, 2011. 2

[198] T. Sönmez and M. U. Ünver. Market design for kidney exchange. *The Handbook of Market Design*, pages 93–137, 2013. 196

[199] T. Sönmez and U. Ünver. Altruistically unbalanced kidney exchange. *Journal of Economic Theory*, 152(1):105–129, 2014. 5

[200] D. Stewart, R. Leishman, E. Sleeman, C. Monstello, G. Lunsford, J. Maghirang, T. Sandholm, S. Gentry, R. Formica, J. Friedewald, and K. Andreoni. Tuning the OPTN's KPD optimization algorithm to incentivize centers to enter their easy-to-match pairs. In *American Transplant Congress (ATC)*, 2013. Talk abstract. 196

[201] A. Stoler, J. B. Kessler, T. Ashkenazi, A. E. Roth, and J. Lavee. Incentivizing organ donor registrations with organ allocation priority. *Health Economics*, 2016. 197

[202] X. Su and S. A. Zenios. Patient choice in kidney allocation: A sequential stochastic assignment model. *Operations Research*, 53:443–455, 2005. 3, 73, 113, 246, 247

[203] Swap.com. URL https://www.swap.com. [Online; accessed 18-June-2016]. 3

[204] K. Takahashi. Recent findings in ABO-incompatible kidney transplantation: classification and therapeutic strategy for acute antibody-mediated rejection due to ABO-blood-group-related antigens during the critical period preceding the establishment of accommodation. *Clinical and Experimental Nephrology*, 11(2):128–141, 2007. 238

[205] N. Thakral. Matching with stochastic arrival. In *Proceedings of the ACM Conference on Economics and Computation (EC)*, pages 343–343. ACM, 2015. 3

[206] The University of Maryland Room Exchange. URL http://reslife.umd.edu/housing/reassignments/roomexchange/. [Online; accessed 18-June-2016]. 3

[207] G. Thiel, P. Vogelbach, L. Gürke, T. Gasser, K. Lehmann, T. Voegele, A. Kiss, and G. Kirste. Crossover renal transplantation: hurdles to be cleared! In *Transplantation Proceedings*, volume 33, pages 811–816. Elsevier, 2001. 4

[208] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 847–855. ACM, 2013. 128

[209] P. Toulis and D. C. Parkes. Design and analysis of multi-hospital kidney exchange mechanisms using random graphs. *Games and Economic Behavior*, 91(0):360–382, 2015. 5, 18, 19, 74, 164, 196

[210] TradeAway. URL http://www.tradeaway.com. [Online; accessed 18-June-2016]. 3

[211] Uber. URL https://uber.com. [Online; accessed 19-June-2016]. 2

[212] UNOS. United Network for Organ Sharing (UNOS). http://www.unos.org/. 131, 154

[213] U. Ünver. Dynamic kidney exchange. *Review of Economic Studies*, 77(1):372–414, 2010. 5, 18, 55, 67, 73, 74, 109, 111, 112, 182, 247

[214] Upwork. URL https://upwork.com. [Online; accessed 19-June-2016]. 2

[215] USRDS. United States Renal Data System (USRDS), 2007. http://www.usrds.org/. 26, 121

[216] S. Vajda. *Mathematical Programming*. Addison-Wesley, 1961. 43

[217] H. Van Hasselt, A. Guez, and D. Silver. Deep reinforcement learning with double Q-learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2015. 128, 194

[218] T. Walsh. Where are the hard manipulation problems? *Journal of Artificial Intelligence Research*, 42:1–29, 2011. 65

[219] S. Woodle, J. Daller, M. Aeder, R. Shapiro, T. Sandholm, V. Casingal, D. Goldfarb, R. Lewis, J. Goebel, and M. Siegler. Ethical considerations for participation of nondirected living donors in kidney exchange programs. *American Journal of Transplantation*, 10:1460–1467, 2010. 25, 197

[220] L. Xu, F. Hutter, H. Hoos, and K. Leyton-Brown. Satzilla: portfolio-based algorithm selection for SAT. *Journal of Artificial Intelligence Research*, 32(1):565–606, 2008. 198

[221] Ö. Yılmaz. Kidney exchange: An egalitarian mechanism. *Journal of Economic Theory*, 146 (2):592–618, 2011. 131

[222] ZenDesk ShiftPlanning. URL https://shiftplanning.zendesk.com. [Online; accessed 18-June-2016]. 3

[223] S. A. Zenios. Optimal control of a paired-kidney exchange program. *Management Science*, 48(3):328–342, 2002. 3, 73, 111, 246, 247