Making Peer Review Robust to Undesirable Behavior

Steven Jecmen

CMU-CS-24-100

February 2024

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Nihar B. Shah, co-chair
Fei Fang, co-chair
Christos Faloutsos
Yiling Chen (Harvard University)
Ashish Goel (Stanford University)

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

This research was sponsored by the Pennsylvania State University under award number 4938CMUARMY0045, the Office of Naval Research under award number N000142212181, the and the National Science Foundation under award numbers 1763734, 1850477, 1942124, 2046640, and 2200410.

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: peer evaluation, peer review, strategic behavior, paper assignment, matching, randomization, strategyproofness

To my parents.

Abstract

Scientific peer review is a critical part of the academic publication process, used across disciplines and venues in various forms. Peer review generally relies on the good-faith participation of many reviewers and authors. However, the peer review process must also deal with different kinds of undesirable behavior from participants, including both malicious attempts to cheat the system and non-malicious cases of unreliability. In this thesis, I describe several practical methods that we have proposed for handling different forms of undesirable behavior in peer review.

First, we consider the problem of reviewer-author collusion, in which malicious reviewers manipulate the paper assignment in order to get assigned to each others' papers so that they can give them positive reviews. We provide efficient algorithms for finding high-quality randomized assignments that limit the probability that a colluding reviewer-author pair succeeds at manipulating the paper assignment. These randomized assignments also mitigate attempts by malicious reviewers to "torpedo" a disliked paper and attempts by malicious authors to de-anonymize their reviewers.

Second, we provide an in-depth analysis of the cost of deploying a randomized assignment in terms of the resulting review quality. We propose methods that leverage the randomness introduced by these randomized assignments in order to evaluate alternative paper assignment policies, and apply these methods to estimate the quality of various potential changes to the assignment policy.

Third, we address the issue of unresponsive reviewers. We provide a simple procedure for finding high-quality paper assignments in a two-phase review process, which allows replacement reviewers to be assigned for any missing or low-effort reviews in the first-phase.

Fourth, we tackle the problem of strategic reviewing, in which reviewers give low scores to their assigned papers in the hopes of increasing their own paper's chances of acceptance. We provide algorithms for finding high-quality assignments that are strategyproof to this form of strategic reviewing.

Finally, we analyze other approaches to addressing the manipulation of paper assignments, which we categorize into mitigation-based and detection-based approaches. We compare the tradeoffs between various proposed approaches to mitigating the impact of manipulated bidding. We also empirically analyze the problem of explicitly detecting reviewer-author collusion rings from the manipulated paper bidding, and furthermore release a dataset on this kind of bidding.

Acknowledgments

I would first like to sincerely thank my advisors, Fei Fang and Nihar B. Shah, for all of the advice and support that they have provided throughout this journey. The work in this thesis would never have been possible without their help, both regarding research and more broadly with other aspects of academic or professional life. I would not have made it here without them.

I would also like to thank the other members of my thesis committee for taking their time to review and provide feedback on this thesis: Christos Faloutsos, Yiling Chen, and Ashish Goel.

In addition, I would like to thank all of the collaborators that I've worked with in my research career so far: Leman Akoglu, Erik Brinkman, Vincent Conitzer, Komal Dhull, Pravesh Kothari, Zun Li, Ryan Liu, Martin Saveski, Arunesh Sinha, Zimeng Song, Long Tran-Thanh, Johan Ugander, Yixuan Even Xu, Minji Yoon, and Hanrui Zhang. Each of these collaborations has taught me something, and I consider myself fortunate to have had the opportunity to work with all of you. Special thanks go to Michael Wellman for introducing me to research and advising me during my undergraduate years.

Finally, I want to thank my friends and family for their invaluable support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"Peer review" is the evaluation of an academic work by others with similar expertise as the authors. It is a ubiquitous part of the academic publication process, used to determine each paper's significance, novelty, and correctness before making the decision of which papers to publish [111, 114, 155]. In turn, the decision of whether to publish a paper can have major impacts on the careers of that paper's authors, as publication records are used to determine who is selected for faculty positions or similar opportunities. Ensuring a high-quality peer review process is thus critically important for the entire scientific community.

Peer review can take a variety of different forms depending on the needs and desires of the venue. Across disciplines, forms of peer review are used by academic journals to evaluate submissions and by funding agencies to evaluate grant proposals. In computer science, publishing is primarily focused around academic conferences. These conferences, led by one or more "program chairs," accept paper submissions from authors during a specified time period and invite researchers (including the authors themselves) to serve as reviewers. Reviewers are then assigned to submissions and given a fixed period of time to complete their reviews. After reviews are completed (and perhaps after the authors are offered a chance to respond), the results are used to determine which papers are accepted to the conference and which are rejected. Within the field of artificial intelligence, there has been an explosion in the number of submissions to major annual conferences, such as the Conference on Neural Information Processing Systems (NeurIPS) and the AAAI Conference on Artificial Intelligence (AAAI), in recent years [134]. Handling such a large number of submissions requires inviting large numbers of reviewers, so that modern peer review in this environment involves the contribution of a huge number of agents.

In recent years, an increasing body of work in computer science has proposed improvements to the peer review process from various directions. The survey [134] provides a comprehensive overview of this literature. In this dissertation, we focus specifically on the paper assignment phase of a conference's peer review process. We propose improvements to and provide analysis of paper assignment policies, with the goal of providing robustness to various forms of undesirable behavior. While conference peer review is the primary setting we consider throughout this thesis, the work in individual chapters is often applicable to other scientific peer review settings or to other forms of peer evaluation.

1.1 Paper Assignment in Conference Peer Review

A crucial part of the conference peer review process is the paper assignment. Before this stage, the conference has recruited a set of reviewers. Upon receiving the set of submitted papers, each paper needs to simultaneously be assigned for review to some number of the impaneled reviewers (often three or four). The assigned reviewers for each paper need to have appropriate expertise in order to provide a high-quality review. Ideally, the assigned reviewers are also interested in reviewing the paper, so that they put sufficient effort into the review. Given the scale of these conferences and the large number of reviewers needed for peer review, reviewer assignment is usually done in an automated fashion. The standard framework for reviewer assignment consists of the following two steps.

Similarity computation. In the first step, the conference computes a "similarity" score (usually in $[0, 1]$) for each reviewer-paper pair, representing the expected quality of review that could be provided by that reviewer for that paper [29, 52, 60, 102, 144, 145]. These similarities are usually computed from three factors:

- text-similarity between the paper and the reviewer's past work, computed using one of a variety of methods [29, 100, 110, 113, 127, 149];
- subject areas selected by each reviewer and each paper's authors; and
- "bids" from each reviewer indicating their level of interest in each paper.

There is no standard method for computing similarities from these components. In practice, different conferences have used different hand-crafted formulas [96, 136]. Notably, reviewer bids are often given a high weight in this computation so that reviewer preferences have a significant influence on the final assignment; however, this can also allow reviewers to manipulate their assigned papers by strategically bidding (i.e., providing bids that do not correspond to their true preferences).

Optimization. In the second step, the paper assignment is chosen to maximize some function of the similarities for assigned reviewer-paper pairs. This maximization is done subject to constraints that each paper is assigned the correct number of reviewers, each reviewer is not assigned too many papers, and no reviewer is assigned to a paper that they have a conflict of interest with. The most common objective is to simply maximize the total similarity of assigned pairs. With this objective, the problem is a standard maximum-weight matching problem, and so can be efficiently solved as a linear program or as a min-cost flow problem. We provide concrete formulations of this optimization problem in later chapters. Other objectives for this problem, such as the max-min fairness [54, 85, 138], are also possible.

1.2 Undesirable Behavior in Conference Peer Review

Peer review is dependent on the good-faith participation of both reviewers and authors. However, there are many ways that participants can behave undesirably and undermine the effectiveness of peer review. Some of this behavior comes in the form of malicious attempts to cheat the peer review process, motivated by the high-stakes nature of publication in a highly competitive academic environment. Other forms of undesirable behavior are instead caused by a lack of effort from participants. Recent conferences have implemented a variety of methods to detect and

mitigate some forms of misbehavior, but there exists a critical need for principled and effective methods that can also be applied in practice. We describe here five forms of undesirable behavior that this dissertation aims to address.

Reviewer-author collusion. Malicious reviewers and authors may form "collusion rings," in which the colluding participants unethically work together to improve each others' chances of paper acceptance. The reviewers attempt to manipulate the paper assignment with the aim of being assigned to review the papers authored by other members of the ring. This manipulation can be most easily done via strategic bidding; however, reviewers and authors can also manipulate their text-similarity scores (e.g., via attacks such as the ones examined in [43]) or their selected subject areas. Once assigned, these reviewers can provide positive reviews and push for the papers to be accepted. Instances of such attacks have been reported on in both an ACM architecture conference [154] and a machine learning conference [99]. Colluding reviewers may go to significant lengths to secure acceptances of their papers [99]: "*The colluders write very positive reviews of these papers, perhaps even lobbying area chairs through back channels outside the view of the other reviewers. Colluders occasionally send threatening email messages to non-colluding reviewers if the colluders discover their names and believe the non-colluding reviewers can be influenced.*"

Torpedo reviewing. Malicious reviewers may target assignment to papers that they dislike with the aim of unfairly rejecting them [16, 92, 117]. This may occur if the targeted paper is on a similar topic to that of the malicious reviewer's own work or if the targeted paper is in a research direction with which the reviewer disagrees. As in the case of reviewer-author collusion, the malicious reviewer manipulates their bids or other aspects of their similarity computation in order to get assigned to the target paper. Upon being assigned, they give the paper an unfair negative review. Over the long term, this can significantly affect the research landscape [92]: "*If a research direction is controversial in the sense that just 2-or-3 out of hundreds of reviewers object to it, those 2 or 3 people can bid for the paper, give it terrible reviews, and prevent publication. Repeated indefinitely, this gives the power to kill off new lines of research to the 2 or 3 most close-minded members of a community, potentially substantially retarding progress for the community as a whole.*"

Strategic reviewing. Since a conference will only accept a limited number of papers, any reviewer who submits a paper to a conference is in competition with the papers that they are assigned to review. Thus, reviewers may maliciously attempt to improve their own paper's chances of acceptance by falsely giving negative reviews to any paper they review, regardless of the content. Unlike in the case of torpedo reviewing, reviewers do not need to manipulate the paper assignment, since the malicious behavior occurs after the paper assignment phase is complete. A controlled experiment found that "strategic reviewing" of this kind did occur in a competitive peer assessment setting (outside of academic peer review) [14].

Reviewer de-anonymization. Reviews in conferences are anonymous, meaning that authors do not know the identity of the reviewers assigned to their paper. This is important since otherwise, malicious authors may be able to pressure reviewers to provide positive reviews. Even without explicit pressure from the authors, a junior reviewer may implicitly feel pressure to not reject the paper of a more senior author, on whom the junior reviewer's career may later depend.

However, if the details of a conference review process are public, including the reviewer-paper similarities, authors could re-run the assignment algorithm and determine their assigned reviewers. Even if the reviewer identities are removed before the similarities are released, this does not provide any guarantee of anonymity and reviewer de-anonymization may still be possible through careful analysis. As a result, conferences must keep the details of their assignment algorithm secret, hindering the transparency of the review process and making future research on peer review more challenging.

Unreliable reviewers. Beyond malicious behavior, reviewers often provide low-effort reviews or fail to submit their reviews at all. These issues are especially of concern as the scale of peer review grows and handling them manually becomes difficult. One possible contributing factor to reviewer unreliability is a mismatch between reviewer expertise and similarity scores, as reviewers assigned to papers in which they are not experts may be less likely to complete high-effort reviews. Thus, improving the quality of the paper assignment may help to prevent this issue from occurring. However, evaluating the accuracy of the computed similarities is itself a challenging problem.

Once a reviewer has gone missing, it is likely necessary to assign additional "emergency" reviewers to fill in the missing reviews. In the tight timeline of a conference review process, this is a major logistical challenge. Many conferences have recently implemented two-phase review processes, in which a subset of the papers are assigned additional reviewers after the initial reviews are completed. One benefit of a two-phase review process is that missing or low-quality reviews in the first phase can be addressed by assigning additional reviewers in the second phase. However, this introduces a new challenge: finding a high-quality paper assignment across the two phases.

1.3 Contributions and Organization

The contributions of this thesis consist of various algorithms, methods, and analyses that altogether make peer review more robust to the five forms of undesirable behavior identified above.

In the first part of this dissertation (Chapter 2), we propose a method for randomizing reviewer assignments in order to provide robustness to various forms of malicious behavior: reviewer-author collusion, torpedo reviewing, and reviewer de-anonymization. Our algorithms find high-quality randomized paper assignments, subject to a constraint that each reviewer-paper pair has only a limited probability of being assigned. By setting this probability limit appropriately, conference program chairs can limit the influence that malicious actors can have on the paper assignment. This chapter is based on our NeurIPS 2020 paper [72] with collaborators Hanrui Zhang, Ryan Liu, Nihar B. Shah, Vincent Conitzer, and Fei Fang. Our randomized assignment algorithms have been **deployed in several venues**, including the AAAI 2022 and 2023 conferences (with **over 17000 submissions in total**) and the 2023 ACM Conference on Knowledge Discovery and Data Mining (KDD). Our assignment algorithms are also **implemented at** `OpenReview.net`, a popular conference management site.

In the second part of this dissertation (Chapter 3), we propose a method for evaluating the quality of counterfactual paper assignment policies; that is, paper assignment policies different from the one actually deployed. This method crucially leverages the randomization introduced by our randomized assignment algorithm in order to evaluate the "cost of randomness" in terms

of reviewer self-reported expertise. As our methodology is very general, it can also be used to evaluate a very broad range of potential paper assignment policies, helping to reduce instances mismatched reviewer expertise and potentially alleviating the issue of unreliable reviewers. This chapter is based on our NeurIPS 2023 paper [132] with collaborators Martin Saveski, Nihar B. Shah, and Johan Ugander.

In the third part of this dissertation (Chapter 4), we consider the problem of finding paper assignments in a two-phase review process. The inclusion of a second phase can provide robustness to a large variety of issues that arise during the first phase, including assigning reviewers to fill in for missing reviews (due to unreliable reviewers) as well as assigning additional reviewers to papers with suspicious reviews (e.g., reviews that appear to be torpedo reviewing). We show that a simple randomized approach suffices to find high-quality paper assignments across both phases. This chapter is based on our paper [74] published at the 2022 AAAI Conference on Human Computation and Crowdsourcing (HCOMP) with collaborators Hanrui Zhang, Ryan Liu, Fei Fang, Vincent Conitzer, and Nihar B. Shah, which received a **"Best Paper" Honorable Mention**.

In the fourth part of this dissertation (Chapter 5), we propose algorithms for finding high-quality paper assignments with guarantees that reviewers cannot improve the outcome of their own submission by rating other submissions lower (i.e., strategic reviewing). This chapter is based on contributions made to our HCOMP 2022 paper [36] with collaborators Komal Dhull, Pravesh Kothari, and Nihar B. Shah.

In the final part of this dissertation (Chapters 6-8), we analyze alternative approaches to addressing the manipulation of paper assignments via bidding, with a focus on the problem of reviewer-author collusion. In Chapter 6, we compare a variety of different approaches to mitigating bid-based manipulation of paper assignments. This chapter is based on our work [73] with collaborators Nihar B. Shah, Fei Fang, and Vincent Conitzer, which received an **"Outstanding Paper" Award** at the Machine Learning Evaluation Standards Workshop at ICLR 2022. In Chapter 7, we consider the problem of detecting reviewer-author collusion from a conference's bidding data. This chapter is based on work currently under review with collaborators Fei Fang, Nihar B. Shah, and Leman Akoglu. In Chapter 8, we collect and release a dataset on the malicious bidding of reviewer-author collusion rings in a mock conference setting. This chapter is based on our paper [75] published at The Web Conference 2023 (WWW) with collaborators Minji Yoon, Vincent Conitzer, Nihar B. Shah, and Fei Fang.

Chapter 2

Mitigating Manipulation via Randomized Paper Assignments

In this chapter, we introduce efficient and practical algorithms for finding randomized paper assignments. Chapter 1 described several forms of undesirable behavior that exploit the predictable nature of the standard, deterministic paper assignment algorithms: reviewer-author collusion, torpedo reviewing, and reviewer de-anonymization. In the former two cases, malicious reviewers aim to manipulate the assignment in order to get assigned to target papers; in the latter, malicious authors aim to discover the identity of their assigned reviewers by re-running the paper assignment algorithm. We propose to address these issues by setting a limit (chosen by conference program chairs) on the probability that each reviewer-paper pair is assigned. The randomized assignment algorithms presented in this chapter have been deployed in several venues for the purpose of preventing reviewer-author collusion and are available for use on the popular conference management site `OpenReview.net`.

The issue of reviewer-author collusion in particular has received serious attention from conferences, and various attempted solutions have been proposed and deployed. For example, the 2021 AAAI Conference on Artificial Intelligence (AAAI) implemented several different techniques to combat this problem [96], including preventing cycles in the reviewer assignment and constraining the geographic regions of the reviewers assigned to the same paper. In Chapter 6, we introduce a variety of these alternative approaches to the problem of collusion and examine the tradeoffs between them. The solutions proposed and analyzed in this chapter are distinct from these other approaches, with their own strengths and weaknesses.

Randomized assignments have previously been used to address the problem of fair division of indivisible goods such as jobs or courses [20, 69], as well as in the context of Stackelberg security games [87]. In the peer review setting, the paper [108] considers fractional paper assignments in order to reason about reviewer incentives when bidding in peer review, where the fractional assignment can be interpreted as reviewers' perceived assignment probabilities under an abstracted assignment algorithm. However, they do not tackle the problem of malicious reviewers nor actually randomize the assignment. The use of randomized reviewer-paper assignments to address the issues of malicious reviewers or reviewer de-anonymization in peer review has not been studied previously.

Our contributions in this chapter are as follows:

- **Conceptual:** We formulate problems concerning the three aforementioned issues in peer review, and propose a framework to address them through the use of randomized paper assignments (Section 2.1).
- **Theoretical:** We design computationally efficient, randomized assignment algorithms that optimally assign reviewers to papers subject to given restrictions on the probability of assigning any particular reviewer-paper pair (Section 2.2). We further consider the more complex case of preventing suspicious *pairs* of reviewers from being assigned to the same paper (Section 2.3). We show that finding the optimal assignment subject to arbitrary constraints on the probabilities of reviewer-reviewer-paper assignments is NP-hard. In the practical special case where the program chairs want to prevent pairs of reviewers within the same subset of some partition of the reviewer set (for example, reviewers at the same academic institution or with the same geographical area of residence) from being assigned to the same paper, we present an algorithm that finds the optimal randomized assignment with this guarantee.
- **Empirical:** We test our algorithms on datasets from past conferences and show their practical effectiveness (Section 2.4). As a representative example, on data reconstructed from the 2018 International Conference on Learning Representations (ICLR), our algorithms can limit the chance of any reviewer-paper assignment to 50% while achieving 90.8% of the optimal total similarity. Our algorithms can continue to achieve this similarity while also preventing reviewers with close associations from being assigned to the same paper. We further demonstrate, using the ICLR 2018 dataset, that our algorithm successfully prevents manipulation of the assignment by a simulated malicious reviewer.

All of the code for our algorithms and our empirical results is freely available online at `https://github.com/theryanl/mitigating_manipulation_via_randomized_reviewer_assignment/`.

2.1 Background and Problem Statements

We first define the standard paper assignment problem, followed by our problem setting. In the standard paper assignment setting, we are given a set $\mathcal{R}$ of m reviewers and a set $\mathcal{P}$ of n papers, along with desired reviewer load ℓ_r (that is, the maximum number of papers any reviewer should be assigned) and desired paper load ℓ_p (that is, the exact number of reviewers any paper should be assigned to).[1] An assignment of papers to reviewers is a bipartite matching between the sets that obeys the load constraints on all reviewers and papers. In addition, we are given a similarity matrix $S \in \mathbb{R}^{m \times n}$ where $S_{r,p}$ denotes how good of a match reviewer r is for paper p. These similarities can be derived from the reviewers' bids on papers, prior publications, conflicts of interest, etc.

The standard problem of finding a maximum sum-similarity assignment [29, 30, 52, 60, 85] is then written as an integer linear program. The decision variables $Z \in \{0,1\}^{m \times n}$ specify the assignment, where $Z_{r,p} = 1$ if and only if reviewer r is assigned to paper p. The objective is

[1]For ease of exposition, we assume that all reviewer and paper loads are equal. In practice, program chairs may want to set different loads for different reviewers or papers; all of our algorithms and guarantees still hold for this case (as does our code).

to maximize $\sum_{r\in\mathcal{R}}\sum_{p\in\mathcal{P}} S_{r,p}Z_{r,p}$ subject to the load constraints $\sum_{p\in\mathcal{P}} Z_{r,p} \le \ell_r, \forall r \in \mathcal{R}$ and $\sum_{r\in\mathcal{R}} Z_{r,p} = \ell_p, \forall p \in \mathcal{P}$. Since the constraint matrix of the linear program (LP) relaxation of this problem is totally unimodular, the solution to the LP relaxation will be integral and so this problem can be solved as an LP. This method of assigning papers has been used by numerous conferences [29, 52, 134], as well as by popular conference management systems EasyChair (`easychair.org`), HotCRP (`hotcrp.com`), and `OpenReview.net`.

Now, suppose there exists a reviewer who wishes to get assigned to a specific paper for some malicious reason and manipulates their similarities in order to do so (as in the cases of reviewer-author collusion and torpedo reviewing). When the assignment algorithm is deterministic, as in previous work [29, 30, 52, 60, 85, 144], a malicious reviewer who knows the algorithm may be able to effectively manipulate it in order to get assigned to the desired paper. To address this issue, we aim to provide a guarantee that regardless of the reviewer bids and similarities, this reviewer-paper pair has only a limited probability of being assigned.

Consider now the challenge of preserving anonymity in releasing conference data. If a conference releases its similarity matrix and its deterministic assignment algorithm, then anyone could reconstruct the full paper assignment. Interestingly, this problem can be solved in the same way as the malicious reviewer problems described above. If the assignment algorithm provides a guarantee that each reviewer-paper pair has only a limited probability of being assigned, then no reviewer's identity can be discovered with certainty.

With this motivation, we now consider Z as stochastic and aim to find a *randomized assignment*, a probability distribution over deterministic assignments. This naturally leads to the following problem formulation.

Definition 2.1 (Pairwise-Constrained Problem)**.** *The input to the problem is a similarity matrix S and a matrix $Q \in [0,1]^{m\times n}$. The goal is to find a randomized assignment of papers to reviewers (i.e., a distribution of Z) that maximizes $\mathbb{E}\left[\sum_{r\in\mathcal{R}}\sum_{p\in\mathcal{P}} S_{r,p}Z_{r,p}\right]$ subject to the constraints $\mathbb{P}[Z_{r,p}=1] \le Q_{r,p}, \forall r \in \mathcal{R}, p \in \mathcal{P}$.*

Since a randomized assignment is a distribution over deterministic assignments, all assignments Z in the support of the randomized assignment must still obey the load constraints $\sum_{p\in\mathcal{P}} Z_{r,p} \le \ell_r, \forall r \in \mathcal{R}$ and $\sum_{r\in\mathcal{R}} Z_{r,p} = \ell_p, \forall p \in \mathcal{P}$. The optimization objective is the expected sum-similarity across all paper-reviewer pairs, the natural analogue of the deterministic sum-similarity objective. The matrix Q is provided by the program chairs of the conference. In practice, all entries are usually set to a constant value, assuming that the chairs have no special prior information about any particular reviewer-paper pair.

To prevent dishonest reviews of papers, program chairs may want to do more than just control the probability of individual paper-reviewer pairs. For example, suppose that we have three reviewers assigned per paper (a very common arrangement in computer science conferences). We might not be particularly concerned about preventing any single reviewer from being assigned to some paper, since even if that reviewer dishonestly reviews the paper, there are likely two other honest reviewers who can overrule the dishonest one. However, it would be much worse if we have two reviewers dishonestly reviewing the same paper, since they could likely overrule the sole honest reviewer.

A second issue is that there may be dependencies within certain pairs of reviewers that cannot be accurately represented by constraints on individual reviewer-paper pairs. For example, we

may have two reviewers a and b who are close collaborators, each of which we are not individually very concerned about assigning to paper p. However, we may believe that in the case where reviewer a has entered into a quid-pro-quo deal to dishonestly review paper p, reviewer b is likely to also be involved in the same deal. Therefore, one may want to strictly limit the probability that **both** reviewers a and b are assigned to paper p, regardless of the limits on the probability that either reviewer individually is assigned to paper p.

With this motivation, we define the following generalization of the Pairwise-Constrained Problem.

Definition 2.2 (Triplet-Constrained Problem)**.** *The input to the problem is a similarity matrix S, a matrix $Q \in [0,1]^{m \times n}$, and a 3-dimensional tensor $T \in [0,1]^{m \times m \times n}$. The goal is to find a randomized assignment of papers to reviewers that maximizes $\mathbb{E}\left[\sum_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}} S_{r,p} Z_{r,p}\right]$ subject to the constraints $\mathbb{P}[Z_{r,p} = 1] \leq Q_{r,p}, \forall r \in \mathcal{R}, p \in \mathcal{P}$ and $\mathbb{P}[Z_{a,p} = 1 \wedge Z_{b,p} = 1] \leq T_{a,b,p}, \forall a, b \in \mathcal{R}$ s.t. $a \neq b, p \in \mathcal{P}$.*

The randomized assignments that solve these problems can be used to address all three challenges we identified earlier:

- **Reviewer-author collusion:** By guaranteeing a limit on the probability that any malicious reviewer or any malicious pairs of reviewers can be assigned to the paper they want, we mitigate the effectiveness of any unethical deals between reviewers and authors by capping the probability that such a deal can be upheld. This guarantee holds regardless of how extreme a reviewers' manipulation of the assignment is and without any assumptions on reviewers' exact incentives. The entries of Q can be set by the program chairs based on their assessment of the risk of allowing the corresponding reviewer-paper pair; for example, an entry can be set low if the reviewer and author have collaborated in the past. The entries of T can be set similarly based on known associations between reviewers.
- **Torpedo reviewing:** By limiting the probability that any reviewer or pair of reviewers can be assigned to a paper they wish to torpedo, we make it much more difficult for a small group of reviewers to shut down a new research direction or to take out competing papers.
- **Reviewer de-anonymization:** To allow for the release of similarities and the assignment algorithm after a conference, all of the entries in Q can simply be set to some reasonable constant value. Even if reviewer and paper names are fully identified through analysis of the similarities, only the distribution over assignments can be recovered and not the specific assignment that was actually used. This guarantees that for each paper, no reviewer's identity can be identified with high confidence, since every reviewer has only a limited chance to be assigned to that paper.

In Sections 2.2 and 2.3, we consider the Pairwise-Constrained Problem and Triplet-Constrained Problem respectively. We also consider several extensions of these problems in Section 2.5.

- We extend our results to an objective based on *fairness*, which we call the stochastic fairness objective, in Section 2.5.1. Following the max-min fairness concept, we aim to maximize the minimum expected similarity assigned to any paper under the randomized assignment: $\min_{p \in \mathcal{P}} \mathbb{E}\left[\sum_{r \in \mathcal{R}} S_{r,p} Z_{r,p}\right]$. We present a version of the Pairwise-Constrained Problem using this objective and an algorithm to solve it, as well as experimental results.
- We address an alternate version of the Pairwise-Constrained Problem in Section 2.5.2

which takes as input the probabilities with which any reviewer may intend to untruthfully review any paper.

2.2 Randomized Assignment with Reviewer-Paper Constraints

In this section we present our main algorithm to solve the Pairwise-Constrained Problem (Definition 2.1), thereby addressing the challenges identified earlier. Before delving into the details of the algorithm, the following theorem states our main result.

Theorem 2.1. *There exists an algorithm which returns an optimal solution to the Pairwise-Constrained Problem in* $poly(m, n)$ *time.*

We describe the algorithm, thereby proving this result, in the next two subsections. Our algorithm that realizes this result has two parts. In the first part, we find an optimal "fractional assignment matrix," which gives the marginal probabilities of individual reviewer-paper assignments. The second part of the algorithm then samples an assignment, respecting the marginal probabilities specified by this fractional assignment.

2.2.1 Finding the Fractional Assignment

Define a *fractional assignment matrix* as a matrix $F \in [0, 1]^{m \times n}$ that obeys the load constraints $\sum_{p \in \mathcal{P}} F_{r,p} \leq \ell_r$ for all reviewers $r \in \mathcal{R}$ and $\sum_{r \in \mathcal{R}} F_{r,p} = \ell_p$ for all papers $p \in \mathcal{P}$. Note that any deterministic assignment can be represented by a fractional assignment matrix with all entries in $\{0, 1\}$. Any randomized assignment is associated with a fractional assignment matrix where $F_{r,p}$ is the marginal probability that reviewer r is assigned to paper p. Furthermore, randomized assignments associated with the same fractional assignment matrix have the same expected sum-similarity. The paper [25] proves an extension of the Birkhoff-von Neumann theorem which shows that all fractional assignment matrices are implementable, i.e., they are associated with at least one randomized assignment. On the other hand, any probability matrix not obeying the load constraints cannot be implemented by a lottery over deterministic assignments, since all deterministic assignments do obey the constraints. Therefore, finding the optimal randomized assignment is equivalent to solving the following LP, which we call $\mathcal{LP}1$:

$$\begin{aligned}
\underset{F \in \mathbb{R}^{m \times n}}{\arg\max} \quad & \sum_{p \in \mathcal{P}} \sum_{r \in \mathcal{R}} S_{r,p} F_{r,p} && && (2.1)\\
\text{subject to} \quad & 0 \leq F_{r,p} \leq 1 && \forall r \in \mathcal{R}, \forall p \in \mathcal{P} && (2.2)\\
& \sum_{p \in \mathcal{P}} F_{r,p} \leq \ell_r && \forall r \in \mathcal{R} && (2.3)\\
& \sum_{r \in \mathcal{R}} F_{r,p} = \ell_p && \forall p \in \mathcal{P} && (2.4)\\
& F_{r,p} \leq Q_{r,p} && \forall r \in \mathcal{R}, \forall p \in \mathcal{P}. && (2.5)
\end{aligned}$$

$\mathcal{LP}1$ has $O(nm)$ bounded variables and $O(n + m)$ other constraints. Using techniques from [153], $\mathcal{LP}1$ can be solved with high probability in $\tilde{O}(nm(n + m) + (n + m)^{2.5})$ time.

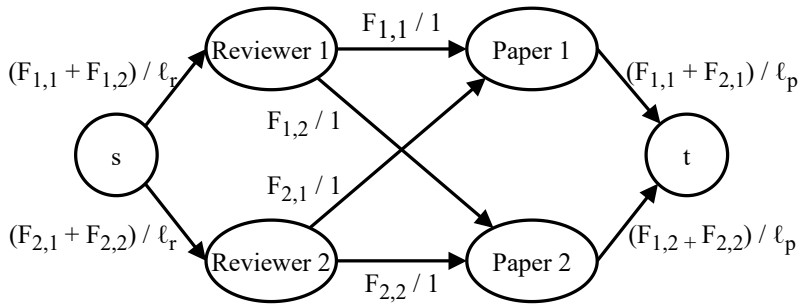


Figure 2.1: Example graph used in the sampling algorithm. Each edge is labeled with its initial flow (left) and its capacity (right).

2.2.2 Implementing the Probabilities

$\mathcal{LP}1$ only finds the optimal marginal assignment probabilities F (where F now refers to a solution to $\mathcal{LP}1$). It remains to show whether and how these marginal probabilities can be implemented as a randomization over deterministic paper assignments. The paper [25] provides a method for sampling a deterministic assignment from a fractional assignment matrix, which completes our algorithm once applied to the optimal solution of $\mathcal{LP}1$. Here we propose a simpler version of the sampling algorithm. Pseudocode for the algorithm is presented as Algorithm 2.1; we describe the algorithm in detail below. In Section 2.5.3, we present a supplementary algorithm to compute the full distribution over deterministic assignments, which [25] does not. Knowing the full distribution may be useful in order to compute other properties of the randomized assignment not calculable from F directly.

We begin by constructing a directed graph $G = (V, E)$ for our problem, along with a capacity function $h : E \to \mathbb{Z}$ (Lines 1-3). First, construct one vertex for each reviewer, one vertex for each paper, and source and destination vertices s, t. Add an edge from the source vertex to each reviewer's vertex with capacity ℓ_r. Add an edge from each paper's vertex to the destination vertex with capacity ℓ_p. Finally, add an edge from each reviewer to each paper with capacity 1. We also construct a flow function $f : E \to \mathbb{R}$, which obeys the flow conservation constraints $\sum_{e \in E \cap (V \times \{v\})} f(e) = \sum_{e \in E \cap (\{v\} \times V)} f(e), \forall v \in V \setminus \{s, t\}$ and the capacity constraints $f(e) \leq h(e), \forall e \in E$ (Line 4). A (possibly fractional) assignment F can be represented as a flow on this graph, where the flow from reviewer i to paper j corresponds to the probability reviewer i is assigned to paper j and the other flows are set uniquely by flow conservation. Due to the load constraints on assignments, the flows on the edges from the papers to the destination must be equal to those edges' capacities and the flows on the edges from the source to the reviewers must be less than or equal to the capacities. An example of this graph with two reviewers and two papers is shown in Figure 2.1.

The algorithm then proceeds in an iterative manner, modifying the flow function f on each iteration. On each iteration, we first check if there exists a "fractional edge," an edge with non-integral flow. If no such edge exists, our current assignment is integral and so we can stop iterating. If there does exist a fractional edge, we then find an arbitrary cycle of fractional edges, ignoring direction (Line 6); this can be done by starting at any fractional edge and walking along fractional edges until a previously-visited vertex is returned to. On finding a cycle, we randomly

Algorithm 2.1 Sampling algorithm for the Pairwise-Constrained Problem.

Input: Fractional assignment matrix F, reviewer set $\mathcal{R}$, paper set $\mathcal{P}$

Ouput: Deterministic assignment matrix Z

Algorithm:

Construct vertex set $V \leftarrow \mathcal{R} \cup \mathcal{P} \cup \{s\} \cup \{t\}$

Construct directed edge set $E \leftarrow \{(r,p)|\forall r \in \mathcal{R}, p \in \mathcal{P}\} \cup \{(s,r)|\forall r \in \mathcal{R}\} \cup \{(p,t)|\forall p \in \mathcal{P}\}$

Construct capacity function $h : E \to \mathbb{Z}$ as $h(e) \leftarrow \begin{cases} 1 & \text{if } e \in \mathcal{R} \times \mathcal{P} \\ \ell_r & \text{if } e \in \{s\} \times \mathcal{R} \\ \ell_p & \text{if } e \in \mathcal{P} \times \{t\} \end{cases}$

Construct initial flow function $f : E \to \mathbb{R}$ as $f(e) \leftarrow \begin{cases} F_{r,p} & \text{if } e = (r,p) \in \mathcal{R} \times \mathcal{P} \\ \sum_{p \in \mathcal{P}} F_{r,p} & \text{if } e = (s,r) \in \{s\} \times \mathcal{R} \\ \sum_{r \in \mathcal{R}} F_{r,p} & \text{if } e = (p,t) \in \mathcal{P} \times \{t\} \end{cases}$

while $\exists e \in E$ such that $f(e) \notin \mathbb{Z}$ **do**

  Find a cycle of edges (ignoring direction) $C = \{e_1, \ldots, e_k\}$ such that $f(e_i) \notin \mathbb{Z}, \forall i \in [k]$

  $A \leftarrow \{e \in C |\ e$ is directed in the same direction as e_1 along the cycle$\}$

  $B \leftarrow C \setminus A$

  $\alpha \leftarrow \min\left(\min_{e \in A} f(e), \min_{e \in B} h(e) - f(e)\right)$

  for $e \in A$ **do**

    $f_1(e) \leftarrow f(e) - \alpha$

  end for

  for $e \in B$ **do**

    $f_1(e) \leftarrow f(e) + \alpha$

  end for

  $\beta \leftarrow \min\left(\min_{e \in A} h(e) - f(e), \min_{e \in B} f(e)\right)$

  for $e \in A$ **do**

    $f_2(e) \leftarrow f(e) + \beta$

  end for

  for $e \in B$ **do**

    $f_2(e) \leftarrow f(e) - \beta$

  end for

  $\gamma \leftarrow \frac{\beta}{\alpha + \beta}$

  With probability γ, $f \leftarrow f_1$; else $f \leftarrow f_2$

end while

$Z_{r,p} = f((r,p)), \forall (r,p) \in \mathcal{R} \times \mathcal{P}$

modify the flow on each of the edges in the cycle in order to guarantee that at least one of the flows becomes integral. In what follows, we first prove that such a cycle of fractional edges can always be found. We then show how to modify the flows in order to guarantee the implementation of the marginal assignment probabilities.

We now show that a directionless cycle of fractional edges must exist whenever one fractional edge exists. Initially, by the properties of F, the total flow on each edge going into vertex t is integral; further, the algorithm only ever changes the flow on edges with non-integral flow. Therefore, the total flow going into t is always integral. By flow conservation, the total flow leaving s is also always integral. So, if there is a fractional edge adjacent to s, there must also be another fractional edge adjacent to s. As already stated, there are no fractional edges adjacent to t. Finally, for each vertex $v \in V \setminus \{s, t\}$, by flow conservation, there can never be only one fractional edge adjacent to v. Therefore, every vertex that is adjacent to a fractional edge must also be adjacent to another fractional edge. This proves that a directionless cycle of fractional edges must exist if one fractional edge exists.

We now show how to modify the flow on the edges in this cycle. We can keep pushing flow in some direction on this cycle (pushing negative flow if the edge is directed backwards) until some edge is at capacity or has 0 flow. Call this amount of additional flow α, and the resulting flow f_1. We can do the same thing in the other direction on the cycle, calling the additional flow β and the resulting flow f_2. Both f_1 and f_2 must have at least one more integral edge than f, since some edge is at capacity. Further, both f_1 and f_2 obey the flow conservation and capacity constraints. Defining $\gamma \leftarrow \frac{\beta}{\alpha+\beta}$, we set $f \leftarrow f_1$ with probability γ and $f \leftarrow f_2$ with probability $1 - \gamma$ (Lines 23-24).

Once all edges are integral (after the final iteration), we construct the sampled deterministic assignment Z from the flow on the reviewer-paper edges (Line 26). Since f obeys the capacity constraints on all edges, Z obeys the load constraints and so is in fact an assignment. Since on each iteration the initial flow f satisfies $f(e) = \gamma f_1(e) + (1 - \gamma) f_2(e), \forall e \in E$, the expected final flow on each edge is always equal to the current flow on that edge. Since the expectation of a Bernoulli random variable is exactly the probability it equals one, each final reviewer-paper assignment $M_{r,p}$ has been chosen with the desired marginal probabilities $F_{r,p}$.

Each iteration of this algorithm takes $O(n+m)$ time to find a cycle in the $O(n+m)$ vertices (if a list of fractional edges adjacent to each vertex is maintained), and it can take $O(nm)$ iterations to terminate since one edge becomes integral every iteration. Therefore, the sampling algorithm is overall $O(nm(n + m))$.

The time complexity of our full algorithm, including both $\mathcal{LP}1$ and the sampling algorithm, is dominated by the complexity of solving the LP. Since standard paper assignment algorithms such as TPMS can be implemented by solving an LP of the same size, our algorithm is comparable in complexity. If a conference currently does solve an LP to find their assignment, whatever LP solver a conference currently uses for their paper assignment algorithm could be used in our algorithm as well.

2.3 Randomized Assignment with Constraints on Pairs of Reviewers

We now turn to the problem of controlling the probabilities that certain pairs of reviewers are assigned to the same paper, defined in Section 2.1 as the Triplet-Constrained Problem (Definition 2.2). In the following subsections, we first show that the problem of finding an optimal randomized assignment given arbitrary constraints on the maximum probabilities of each reviewer-reviewer-paper grouping is NP-hard. We then show that, for the practical special case

of restrictions on reviewers from the same subset of a partition of $\mathcal{R}$ (such as the same primary academic institution or geographical area of residence), an optimal randomized assignment can be found efficiently.

2.3.1 NP-Hardness of Arbitrary Constraints

As described in Section 2.1, solving the Triplet-Constrained Problem would allow the program chairs of a conference maximum flexibility in how they control the probabilities of the assignments of pairs of reviewers. Unfortunately, as the following theorem shows, this problem cannot be efficiently solved.

Theorem 2.2. *The Triplet-Constrained Problem is NP-hard, by reduction from 3-Dimensional Matching.*

3-Dimensional Matching is an NP-complete decision problem that takes as input three sets A, B, C of size s as well as a collection of tuples in $A \times B \times C$; the goal is to find a choice of s tuples out of the collection such that no elements of any set are repeated [79]. Our reduction maps sets $A \cup B$ to $\mathcal{R}$ and C to $\mathcal{P}$, and constructs $T \in \{0,1\}^{m \times m \times n}$ to allow only the assignments where the corresponding tuples are allowable in the 3-Dimensional Matching instance. The full proof is stated in Section 2.6.

Theorem 2.2 implies a more fundamental result about the feasible region of implementable reviewer-reviewer-paper probability tensors, that is, the tensors $G \in [0,1]^{m \times m \times n}$ where entry $G_{i,j,p}$ represents the marginal probability that both reviewers i and j are assigned to paper p under some randomized assignment. We can represent any deterministic assignment by a 3-dimensional tensor $M \in \{0,1\}^{m \times m \times n}$ where $M_{i,j,p} = 1$ if and only if both reviewers i and j are assigned to paper p. Just as in the earlier case of fractional assignment matrices, the set of implementable probability tensors is a polytope with deterministic assignment tensors at the vertices (since any implementable probability tensor is a convex combination of deterministic assignment tensors). For fractional reviewer-paper assignment matrices, this polytope was defined by a small number ($O(nm)$) of linear inequalities, despite the fact that it has a large number of vertices (factorial in n and m). However, this is no longer the case for reviewer-reviewer-paper probabilities.

Corollary 2.1. *The polytope of implementable reviewer-reviewer-paper probabilities is not expressible in a polynomial (in m and n) number of linear inequality constraints (assuming $P \neq NP$).*

The proof of this result is also stated in Section 2.6.

2.3.2 Constraints on Disjoint Reviewer Sets

Since the most general problem of arbitrary constraints on reviewer-reviewer-paper triples is NP-hard, we must restrict ourselves to tractable special cases of interest. One such special case arises when the program chairs of a conference can partition the reviewers in such a way that they wish to prevent any two reviewers within the same subset from being assigned to the same paper. For example, reviewers can be partitioned by their primary academic institution. Since reviewers at the same institution are likely closely associated, program chairs may believe that placing them together as co-reviewers is more risky than would be implied by our concern about either reviewer individually. In this case, there may not even be any concern about the reviewers'

motivations; the concern may simply be that the reviewers' opinions would not be sufficiently independent. Other partitions of interest could be the reviewer's geographical area of residence or research sub-field, as each of these defines a "community" of reviewers that may be more closely associated. This special case corresponds to instances of the Triplet-Constrained Problem where $T_{a,b,p} = 0$ if reviewers a and b are in the same subset, and $T_{a,b,p} = 1$ otherwise. In fact, both the "Geographic Diversity" and "Seniority" constraints defined in [96], which were used by the AAAI 2021 conference (among other venues), can be represented in this form.

We formally define this problem as follows:

Definition 2.3 (Partition-Constrained Problem)**.** *The input to the problem is a similarity matrix S, a matrix $Q \in [0,1]^{m \times n}$, and a partition of the reviewer set into subsets $I_1, \ldots, I_d \subseteq \mathcal{R}$. The goal is to find a randomized assignment of papers to reviewers that maximizes $\mathbb{E}\left[\sum_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}} S_{r,p} Z_{r,p}\right]$ subject to the constraints that $\mathbb{P}[Z_{r,p} = 1] \leq Q_{r,p}, \forall r \in \mathcal{R}, p \in \mathcal{P}$, and $\mathbb{P}[Z_{a,p} = 1 \wedge Z_{b,p} = 1] = 0, \forall a, b \in I_i, \forall i \in [d]$.*

For this special case of the Triplet-Constrained Problem, we show that the problem is efficiently solvable, as stated in the following theorem.

Theorem 2.3. *There exists an algorithm which returns an optimal solution to the Partition-Constrained Problem in poly(m, n) time.*

We present the algorithm that realizes this result in the following subsections, thus proving the theorem. The algorithm has two parts: it first finds a fractional assignment matrix F meeting certain requirements, and then samples an assignment while respecting the marginal assignment probabilities given by F and additionally never assigning two reviewers from the same subset to the same paper. For ease of exposition, we first present the sampling algorithm, and then present an LP which finds the optimal fractional assignment matrix meeting the necessary requirements.

Partition-Constrained Sampling Algorithm

The sampling algorithm we present in this section takes as input a fractional assignment matrix F and samples an assignment while respecting the marginal assignment probabilities given by F. The sampling algorithm is based on the following lemma:

Lemma 2.1. *Consider any fractional assignment matrix F and any partition of $\mathcal{R}$ into subsets $I_1, \ldots, I_d$.*

- *(i) There exists a sampling algorithm that implements the marginal assignment probabilities given by F and runs in $O(nm(n+m))$ time such that, for all papers $p \in \mathcal{P}$ and subsets $I \in \{I_1, \ldots, I_d\}$ where $\sum_{r \in I} F_{r,p} \leq 1$, the algorithm never samples an assignment assigning two reviewers from subset I to paper p.*
- *(ii) For any sampling algorithm that implements the marginal assignment probabilities given by F, for all papers $p \in \mathcal{P}$ and subsets $I \in \{I_1, \ldots, I_d\}$ where $\sum_{r \in I} F_{r,p} > 1$, the expected number of pairs of reviewers from subset I assigned to paper p is strictly positive.*

The sampling algorithm which realizes Lemma 2.1 has an additional helpful property, which holds *simultaneously* for all papers and subsets. We state the property in the following corollary and make use of it later:

Corollary 2.2. *For any fractional assignment matrix F, the sampling algorithm that realizes Lemma 2.1 minimizes the expected number of pairs of reviewers from subset I assigned to paper p simultaneously for all papers $p \in \mathcal{P}$ and subsets $I \in \{I_1, \ldots, I_d\}$ among all sampling algorithms*

implementing the marginal assignment probabilities given by F.

We present the sampling algorithm that realizes these results here, and prove the guarantees stated in Lemma 2.1 and Corollary 2.2 in Section 2.6. This algorithm is a modification of the sampling algorithm from Theorem 2.1 presented earlier as Algorithm 2.1.

We first provide some high-level intuition about the modifications to Algorithm 2.1. For any fractional assignment matrix F, for any subset I and paper p, the expected number of reviewers from subset I assigned to paper p is $\sum_{r \in I} F_{r,p}$. This is equal to the initial load from subset I on paper p in Algorithm 2.1 (that is, the sum of the flow on all edges from reviewers in subset I to paper p). Note that at Algorithm 2.1's conclusion, when all edges are integral, the load from subset I on paper p is equal to the number of reviewers from subset I assigned to paper p. Therefore, if the fractional assignment F is such that the initial expected number of reviewers from subset I assigned to paper p is no greater than 1 (as stated in part (i) of Lemma 2.1), we want to keep the load from subset I on paper p close to its initial value so that the final number of reviewers from subset I assigned to paper p is also no greater than 1. With this reasoning, we modify Algorithm 2.1 so that in each iteration, it ensures that the total load on each paper from each subset is unchanged if originally integral and is never moved past the closest integer in either direction if originally fractional.

The algorithm realizing Lemma 2.1 and Corollary 2.2 is obtained by changing three lines in Algorithm 2.1, as follows:

- Line 6 is replaced with the subroutine in Algorithm 2.2.
- Line 9 is changed to:

$$\alpha \leftarrow \min\left(\min_{e \in A} f(e), \min_{e \in B} h(e) - f(e), \min_{t \in D_1} t - \lfloor t \rfloor, \min_{t \in D_2} \lceil t \rceil - t\right).$$

- Line 16 is changed to:

$$\beta \leftarrow \min\left(\min_{e \in A} h(e) - f(e), \min_{e \in B} f(e), \min_{t \in D_1} \lceil t \rceil - t, \min_{t \in D_2} t - \lfloor t \rfloor\right).$$

The primary modification we make to Algorithm 2.1 is replacing Line 6 with the subroutine in Algorithm 2.2. In each iteration, when we look for an undirected cycle of fractional edges in the graph, we now choose the cycle carefully rather than arbitrarily. We find a cycle by starting from an arbitrary fractional edge in the graph and walk along adjacent fractional edges (ignoring direction) until we repeat a previously-visited vertex. As we do this, whenever we take a fractional edge from a reviewer in subset I into paper p, there are two cases.

- Case 1: If there exists a different fractional edge from paper p to subset I (Line 8 in Algorithm 2.2), we take this edge next. Note that if the total load from subset I on paper p is integral, such an edge must exist.
- Case 2: Otherwise (Line 12 in Algorithm 2.2), we must take a fractional edge from paper p to some other subset J. In this case, the total load from subset I on paper p must not be integral. We choose the subset J so that the total load from subset J on paper p is also not integral. Such a subset must exist since the total load on paper p is always integral. We keep track of both the total load from I and from J on p, for every occurrence of this case along the cycle (Lines 14 and 15 in Algorithm 2.2).

Algorithm 2.2 Loop-finding subroutine (replacing Line 6 in Algorithm 2.1).

Construct the set of undirected edges $E_U \leftarrow E \cup \{(v,u) \mid (u,v) \in E\}$
Construct the undirected flow function $f_U : E_U \to \mathbb{R}$ as $f_U((u,v)) \leftarrow \begin{cases} f((u,v)) & \text{if } (u,v) \in E \\ f((v,u)) & \text{otherwise} \end{cases}$
Find arbitrary edge $(u,v) \in E$ such that $f((u,v)) \notin \mathbb{Z}$
$C \leftarrow \{(u,v)\}$
$D_1 \leftarrow \{\}$, $D_2 \leftarrow \{\}$
while v has not previously been visited **do**
    Visit v
    if $u \in \mathcal{R}$ and $v \in \mathcal{P}$ **then**
        Set $I \in \{I_1, \ldots, I_d\}$ such that $u \in I$
        if $\exists w \in I \setminus \{u\}$ such that $(v,w) \in E_U$ and $f_U((v,w)) \notin \mathbb{Z}$ **then**
            Find such a w
        else
            For some $J \in \{I_1, \ldots, I_d\} \setminus \{I\}$ such that $\sum_{r \in J} f((r,v)) \notin \mathbb{Z}$, find $w \in J$ such that $(v,w) \in E_U$ and $f_U((v,w)) \notin \mathbb{Z}$
            $D_1 \leftarrow D_1 \cup \{\sum_{r \in I} f((r,v))\}$ (corresponding to (u,v))
            $D_2 \leftarrow D_2 \cup \{\sum_{r \in J} f((r,v))\}$ (corresponding to (v,w))
        end if
    else
        Find $w \in V \setminus \{u\}$ such that $(v,w) \in E_U$ and $f_U((v,w)) \notin \mathbb{Z}$
    end if
    $C \leftarrow C \cup \{(v,w)\}$
    $u \leftarrow v$
    $v \leftarrow w$
end while
Set e_1 as the first edge in C leaving v
Set e_{-1} as the last edge in C (entering v)
Remove edges preceding e_1 from C, and remove the corresponding elements from D_1 and D_2
if $v \in \mathcal{P}$ and $\exists I \in \{I_1, \ldots, I_d\}$ such that $e_1 \in \{v\} \times I$ and $e_{-1} \in I \times \{v\}$ **then**
    Remove the elements corresponding to e_1 and e_{-1} from D_1 and D_2
end if
if $e_1 \notin E$ **then**
    Swap D_1 and D_2
end if
Replace each edge in C from E_U with the corresponding edge from E

In Case 1, no matter how much flow is pushed on the cycle, the total load from subset I on paper p will be preserved exactly. However, due to Case 2, we must modify the choice of how much flow to push on the cycle to ensure that the loads are preserved as desired. Specifically, we only

push flow in a given direction on the cycle until the total load for either subset I or J on paper p is integral, for any I, J, p found in Case 2. The total loads from each subset on each paper found in Case 2 are saved in either set D_1 or set D_2 depending on the direction of the corresponding edges in the cycle, and each subset-paper pair with an edge corresponding to an element of D_1 or D_2 has only that one edge in the cycle. If the total (fractional) load from subset I on paper p is t, then only $\lceil t \rceil - t$ additional flow can be added to any edge from subset I to paper p before the load becomes integral; similarly, only $t - \lfloor t \rfloor$ flow can be removed from any edge before the load becomes integral. This leads to the stated changes to Lines 9 and 16 in Algorithm 2.1.

Therefore, on each iteration, we push flow until either the flow on some edge is integral (as in the original algorithm), or until the total load on some paper from some subset is integral. This implies that the algorithm still terminates in a finite number of iterations. In addition, by the end of the algorithm, the total load on each paper from each subset is preserved exactly if originally integral and rounded in either direction if originally fractional, as desired.

The time complexity of this modified algorithm is identical to that of the original algorithm from Theorem 2.1, since finding a cycle takes the same amount of time (if a fractional adjacency list for each subset is used) and only a maximum of $O(m)$ extra iterations are performed (if an subset's total load becomes integral rather than an edge's flow). Therefore, the algorithm is overall $O(nm(n+m))$.

Finding the Optimal Partition-Constrained Fractional Assignment

Lemma 2.1 provides necessary and sufficient conditions for the fractional assignment matrices for which it is possible to prevent all pairs of same-subset reviewers from being assigned to the same paper. Therefore, to find an optimal fractional assignment with this property, we just need to add dn constraints to $\mathcal{LP}1$. We call this new LP $\mathcal{LP}2$:

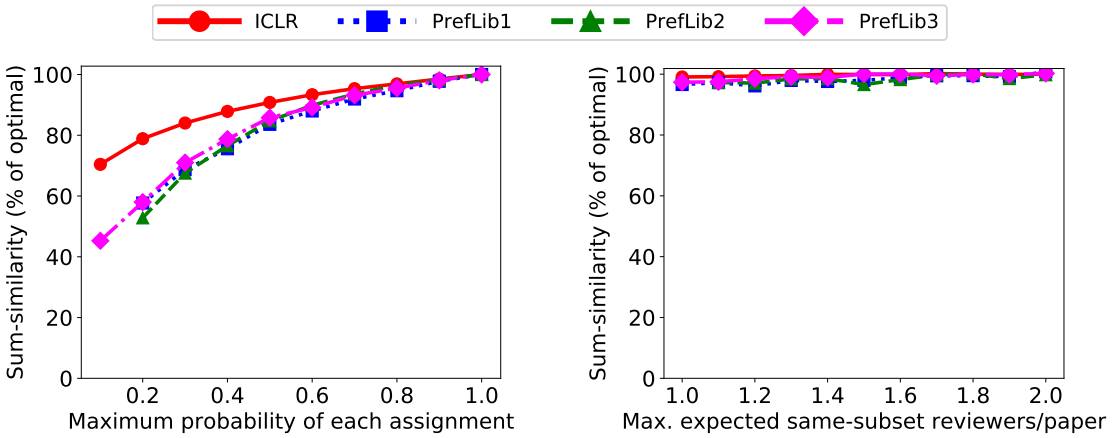
$$\underset{F \in \mathbb{R}^{m \times n}}{\arg\max} \quad \sum_{p \in \mathcal{P}} \sum_{r \in \mathcal{R}} S_{r,p} F_{r,p} \tag{2.6}$$

subject to Constraints (2.2–2.5) from $\mathcal{LP}1$ and

$$\sum_{r \in I} F_{r,p} \leq 1 \qquad \forall I \in \{I_1, \ldots, I_d\}, \forall p \in \mathcal{P}. \tag{2.7}$$

The solution to $\mathcal{LP}2$ when paired with the sampling algorithm from Section 2.3.2 never assigns two reviewers from the same subset to the same paper. Furthermore, since any fractional assignment F not obeying Constraint (2.7) will have a strictly positive probability of assigning two reviewers from the same subset to the same paper, $\mathcal{LP}2$ finds the optimal fractional assignment with this guarantee. This completes the algorithm for the Partition-Constrained Problem.

Additionally, Corollary 2.2 shows that the sampling algorithm from Section 2.3.2 is optimal in the expected number of same-subset reviewer pairs, for any fractional assignment. If the guarantee of entirely preventing same-subset reviewer pairs is not strictly required, Constraint (2.7) in $\mathcal{LP}2$ can be loosened (constraining the subset loads to a higher value) without removing it entirely. For the resulting fractional assignment F, the sampling algorithm from Section 2.3.2 still minimizes the expected number of pairs of reviewers from any subset on any paper, as compared to any other sampling algorithm implementing F. Since the subset loads are still constrained, the expected number of same-subset reviewer pairs will be lower than in the solution to the



(a) Pairwise-Constrained Problem

(b) Partition-Constrained Problem with three random subsets

Figure 2.2: Experimental results on four conference datasets.

Pairwise-Constrained Problem (at the cost of some expected sum-similarity). We examine this tradeoff experimentally in Section 2.4.

2.4 Experiments

We test our algorithms on several real-world datasets. The first real-world dataset is a similarity matrix recreated from ICLR 2018 data in [160]; this dataset has $m = 2435$ reviewers and $n = 911$ papers. We also run experiments on similarity matrices created from reviewer bid data for three AI conferences from PrefLib dataset MD-00002 [106], with sizes $(m = 31, n = 54)$, $(m = 24, n = 52)$, and $(m = 146, n = 176)$ respectively. For all three PrefLib datasets, we transformed "yes," "maybe," and "no response" bids into similarities of 4, 2, and 1 respectively, as is often done in practice [136]. As done in [160], we set loads $\ell_r = 6$ and $\ell_p = 3$ for all datasets since these are common loads for computer science conferences (except on the PrefLib2 dataset, for which we set $\ell_r = 7$ for feasibility).

We run all experiments on a computer with 8 cores and 16 GB of RAM, running Ubuntu 18.04 and using Gurobi 9.0.2 [101] to solve the LPs. Our algorithm for the Pairwise-Constrained Problem takes an average of 41 seconds to complete on ICLR; our algorithm for the Partition-Constrained Problem takes an average of 45 seconds. As expected, the running time is dominated by the time taken to solve the LP.

2.4.1 Quality of Resulting Assignments

We first study our algorithm for the Pairwise-Constrained Problem, as described in Section 2.2. In this setting, program chairs must make a tradeoff between the quality of the output assignments and guarding against malicious reviewers or reviewer de-anonymization by setting the values of the maximum-probability matrix Q. We investigate this tradeoff on real datasets. All results in this section are averaged over 10 trials with error bars plotted representing the standard error of the mean, although they are sometimes not visible since the variance is very low.

In Figure 2.2a, we set all entries of the maximum-probability-matrix Q equal to the same constant value q (varied on the x-axis), and observe how the sum-similarity value of the assignment computed via our algorithm from Section 2.2 changes as q increases from 0.1 to 1 with an interval of 0.1. We report the sum-similarity as a percentage of the unconstrained optimal solution's objective. This unconstrained optimal solution maximizes sum-similarity through a deterministic assignment as is popularly done today [30, 52, 60, 85, 98, 102, 144], and does not address the aforementioned challenges. We see that our algorithm trades off the maximum probability of an assignment gracefully against the sum-similarity on all datasets. For instance, with $q = 0.5$, our algorithm achieves 90.8% of the optimal objective value on the ICLR dataset. In practice, this would allow the program chairs of a conference to limit the chance that any malicious reviewer is assigned to their desired paper to 50% without suffering a significant loss of assignment quality. When q is too small, a feasible assignment may not exist in some datasets (e.g., $q = 0.1$ for PrefLib2).

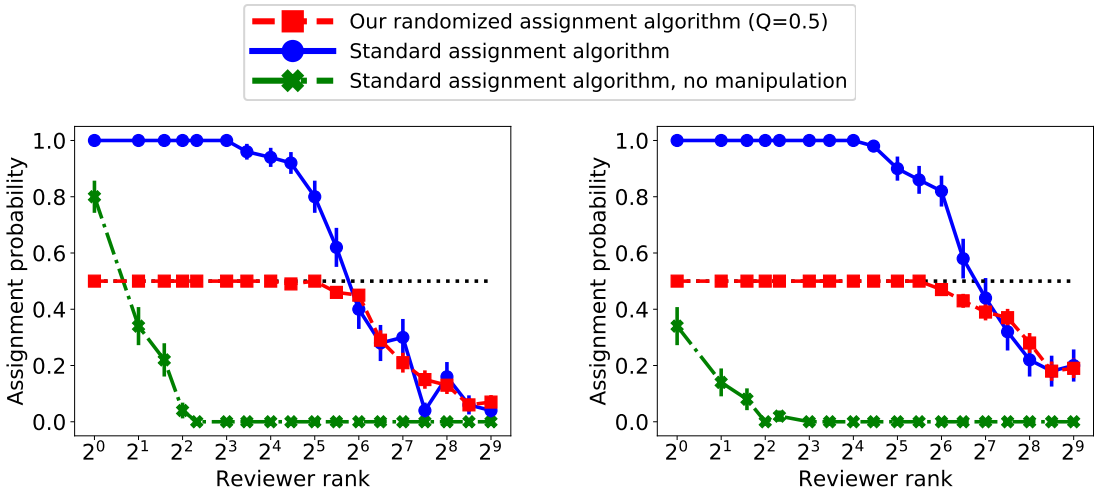
We next test our algorithm for the Partition-Constrained Problem discussed in Section 2.3.2. In this algorithm, program chairs can navigate an additional tradeoff between the number of same-subset reviewers assigned to the same paper and the assignment quality; we investigate this tradeoff here. On ICLR, we fix $q = 0.5$ and randomly assign reviewers to subsets of size 15, using this as our partition of $\mathcal{R}$ (since the dataset does not include any reviewer information). Each subset represents a group of reviewers with close associations, such as reviewers from the same institution. Our algorithm is able to achieve 100% of the optimal objective for the Pairwise-Constrained Problem with $q = 0.5$ while preventing any pairs of reviewers from the same subset from being assigned to the same paper.

Since our algorithm achieves the full possible objective in this setting, we now run experiments with a considerably more restrictive partition constraint. In Figure 2.2b, we show an extreme case where we randomly assign reviewers to 3 subsets of equal size (sizes 811, 11, 8 and 48 on ICLR and the PrefLib datasets, respectively, with the remainder assigned to a dummy fourth subset), again fixing $q = 0.5$. We then gradually loosen the constraints on the expected number of same-subset reviewers assigned to the same paper by increasing the constant in Constraint (2.7) from 1 to 2 in increments of 0.1, shown on the x-axis. We plot the sum-similarity objective of the resulting assignment, expressed as a percentage of the optimal non-partition-constrained solution's objective (i.e., the solution to the Pairwise-Constrained Problem with $q = 0.5$). Even in this extremely constrained case with only a few subsets, we still achieve 99.1% of the non-partition-constrained objective while entirely preventing same-subset reviewer pairs on ICLR.

In Section 2.5.4, we present results for additional experiments on synthetic similarities, where we find results qualitatively similar to those presented here. We also run experiments for a fairness objective, which we present in Section 2.5.1.

2.4.2 Effectiveness at Preventing Manipulation

We now describe experiments evaluating the effectiveness of our algorithm at preventing manipulation on the ICLR dataset against a simulated reviewer bidding model. We assume that there is one malicious reviewer who is attempting to maximize their chances of being assigned to a target paper solely through bidding (and not through other means). Since the ICLR similarities are reconstructed purely from the text similarity with each reviewers' past work and do not contain any bidding, we supplement them with synthetic bids. Specifically, each reviewer r chooses a



(a) Bidding scale $\gamma = 2$

(b) Bidding scale $\gamma = 4$

Figure 2.3: Effectiveness of bidding manipulation on the ICLR dataset. One malicious reviewer manipulates their bids to get assigned to a target paper. The probability of the malicious reviewer-target paper assignment varies on the y-axis as the rank of the malicious reviewer's pre-bid similarity with the target paper changes on the x-axis.

bid $b_{r,p} \in \{-1, 0, 1\}$ for each paper p, indicating "not interested," "neutral," or "interested" respectively. Based on the similarity function used in the 2016 Conference on Neural Information Processing Systems (NeurIPS) [136], we compute the final similarity between reviewer r and paper p as $S'_{r,p} = \gamma^{b_{r,p}} S_{r,p}$, where $S_{r,p}$ is the text similarity from the ICLR dataset and γ is a fixed scale parameter.

In our experiment, the malicious reviewer bids 1 on their target paper and -1 on all other papers. The other (honest) reviewers bid according to a simple randomized model constructed to match characteristics of the bidding observed in NeurIPS 2016 [136]. We divide the reviewers uniformly at random into three groups. The first group contains 20% of the reviewers, who all bid 0 on all papers. The second group contains 50% of the reviewers, who bid non-zero on a low number of papers. These reviewers consider each paper within the 10% of papers that have highest text similarity with them, and independently choose to bid non-zero on each one with probability 0.016. If a paper is selected to bid non-zero, the bid is chosen from $\{-1, 1\}$ with uniform probability. The third group contains 30% of the reviewers, who bid non-zero on a high number of papers. They follow the same bidding procedure as the second group, but bid with probability 0.24.

The results of this experiment are shown in Figure 2.3. We choose a target paper uniformly at random, and choose the malicious reviewer to be the reviewer with the x^{th} highest text similarity with that paper (varying x on the x-axis). Note that the x-axis is on a log-scale. We then have all reviewers bid in the manner described above, and compute the assignment with either the standard deterministic assignment algorithm described in Section 2.1 or our randomized assignment algorithm for the Pairwise-Constrained Problem, setting all entries of Q to 0.5. We then observe the probability that the malicious reviewer is assigned to the target paper (that is, the probability

with which the manipulation is successful), which is in $\{0, 1\}$ for the deterministic algorithm but can be non-integral for our algorithm. For each point on the x-axis, we average results over 50 choices of target paper, giving an overall success rate for the manipulation under a uniform choice of papers (reported on the y-axis, with error bars plotted representing the standard error of the mean). For comparison, we also plot the case where only the honest reviewers bid and the malicious reviewer does not bid.

There are three key takeaways from this experiment. First, we see that when a reviewer does not bid, their assignment probability is low for any reviewer not ranked in the top 4 for that paper in terms of the text similarity. Second, when the malicious reviewer does bid, the manipulation has a high success rate under the standard assignment algorithm. For example, the 8^{th} ranked reviewer for any paper is never assigned if they do not bid, but with bids they can manipulate in order to guarantee their assignment. Moreover, even the 100^{th} ranked reviewer has a decent probability (above 0.25) of getting assigned the target paper if the reviewer bids maliciously. This indicates that manipulation from reviewers is quite powerful in standard assignment algorithms, potentially compromising the integrity of the assignment. Third, our algorithm always limits the probability of successful manipulation to the desired level of 0.5, reflecting the theoretical guarantees presented earlier in the paper. For malicious reviewers who have low text similarity with the target paper (e.g., reviewers from a different subject area), our algorithm occasionally gives the manipulation a marginally higher probability to succeed as compared to the standard assignment algorithm since the set of possible reviewers for each paper is larger. However, manipulation from these low-similarity reviewers is unlikely to succeed in the first place (with probability below the desired limit of 0.5), and it is envisaged to be easier for program chairs to manually spot unusual bids from reviewers outside of a paper's subject area.

2.5 Supplemental Material

In this section, we present various supplemental results, including problem variants and additional experiments.

2.5.1 Stochastic Fairness Objective

An alternate objective to the sum-similarity objective has been studied in past work [54, 138], aiming to improve the fairness of the assignment with respect to the papers. Rather than maximizing the sum-similarity across all papers, this objective maximizes the minimum total similarity assigned to any paper:

$$\begin{aligned}
\underset{Z \in \mathbb{R}^{m \times n}}{\arg\max} \quad & \min_{p \in \mathcal{P}} \sum_{r \in \mathcal{R}} S_{r,p} Z_{r,p} \\
\text{subject to} \quad & Z_{r,p} \in \{0, 1\} && \forall r \in \mathcal{R}, \forall p \in \mathcal{P} \\
& \sum_{p \in \mathcal{P}} Z_{r,p} \leq \ell_r && \forall r \in \mathcal{R} \\
& \sum_{r \in \mathcal{R}} Z_{r,p} = \ell_p && \forall p \in \mathcal{P}.
\end{aligned}$$

Due to the minimum in the objective, this problem is NP-hard [54]; the paper [138] presents an algorithm to find an approximate solution.

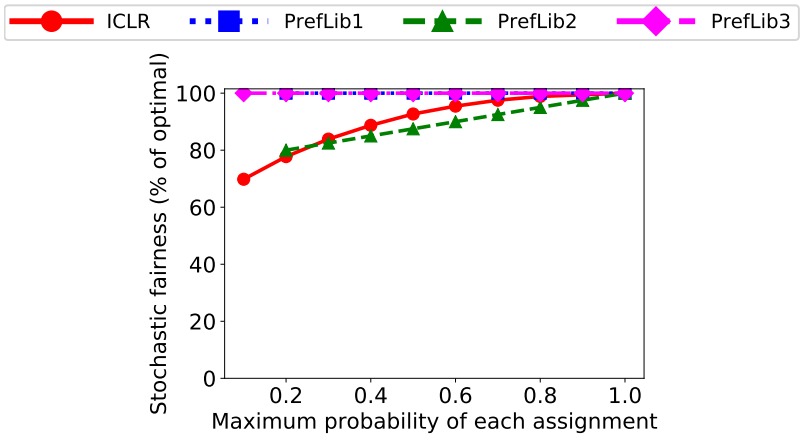


Figure 2.4: Experimental results for the Fair Pairwise-Constrained Problem.

In our setting of randomized assignments, we consider an analogous fairness objective, which we call the stochastic fairness objective: $\min_{p \in \mathcal{P}} \mathbb{E}\left[\sum_{r \in \mathcal{R}} S_{r,p} Z_{r,p}\right]$. The problem involving this objective is defined as follows.

Definition 2.4 (Fair Pairwise-Constrained Problem)**.** *The input to the problem is a similarity matrix S and a matrix $Q \in [0,1]^{m \times n}$. The goal is to find a randomized assignment of papers to reviewers that maximizes $\min_{p \in \mathcal{P}} \mathbb{E}\left[\sum_{r \in \mathcal{R}} S_{r,p} Z_{r,p}\right]$ subject to the constraints that $\mathbb{P}[Z_{r,p} = 1] \leq Q_{r,p}, \forall r \in \mathcal{R}, p \in \mathcal{P}$.*

This problem definition is identical to that of the Pairwise-Constrained Problem (Definition 2.1), with the exception that the objective to maximize is now the stochastic fairness objective rather than the sum-similarity. Note that this objective is not equal to the "expected fairness" (i.e., $\mathbb{E}\left[\min_{p \in \mathcal{P}} \sum_{r \in \mathcal{R}} S_{r,p} Z_{r,p}\right]$), but by Jensen's inequality it is an upper bound on the expected fairness.

Fortunately, this problem is solvable efficiently, as the following theorem states.

Theorem 2.4. *There exists an algorithm which returns an optimal solution to the Fair Pairwise-Constrained Problem in poly(m, n) time.*

We now present our algorithm for solving the Fair Pairwise-Constrained Problem, thereby proving the theorem. It proceeds in a similar manner as the algorithm for the Pairwise-Constrained Problem presented in Section 2.2.

The algorithm first finds an optimal fractional assignment matrix, since the stochastic fairness objective depends only on the marginal probabilities in the fractional assignment matrix. The

optimal fractional assignment is found by the following LP, which we call $\mathcal{LP}3$:

$$\begin{aligned}
\underset{F\in\mathbb{R}^{m\times n},x\in\mathbb{R}}{\arg\max} \quad & x && && (2.8)\\
\text{subject to} \quad & 0 \le F_{r,p} \le 1 && \forall r \in \mathcal{R}, \forall p \in \mathcal{P} && (2.9)\\
& \sum_{p\in\mathcal{P}} F_{r,p} \le \ell_r && \forall r \in \mathcal{R} && (2.10)\\
& \sum_{r\in\mathcal{R}} F_{r,p} = \ell_p && \forall p \in \mathcal{P} && (2.11)\\
& F_{r,p} \le Q_{r,p} && \forall r \in \mathcal{R}, \forall p \in \mathcal{P} && (2.12)\\
& x \le \sum_{r\in\mathcal{R}} S_{r,p} F_{r,p} && \forall p \in \mathcal{P}. && (2.13)
\end{aligned}$$

For any F, the optimal value of x is always $\min_{p\in\mathcal{P}} \sum_{r\in\mathcal{R}} S_{r,p}F_{r,p}$, the stochastic fairness of F. For a fixed x, the feasible region of F in $\mathcal{LP}3$ is exactly the space of fractional assignment matrices with stochastic fairness no less than x. Therefore, $\mathcal{LP}3$ will find an optimal fractional assignment matrix for the stochastic fairness objective.

Once an optimal fractional assignment matrix has been found, it only remains to sample a deterministic assignment from it. This is done with the sampling algorithm described in Section 2.2.2, just as in the Pairwise-Constrained Problem.

We now present some empirical results for this algorithm on the four conference datasets described in Section 2.4. We set all entries of Q equal to the same constant value q (varied on the x-axis), and observe how the stochastic fairness objective of the assignment changes as q increases from 0.1 to 1 with an interval of 0.1. Since the expectation is inside a minimum in the objective, the objective cannot be estimated without bias by averaging together the stochastic fairness of sampled deterministic assignments. Due to this difficulty, we plot the exact objective of our randomized assignment (i.e., the optimal objective value of $\mathcal{LP}3$) rather than averaging over multiple samples, and report the objective as a percentage of the unconstrained optimal solution's objective (that is, the algorithm's solution when $q = 1$). As Figure 2.4 shows, our algorithm finds a randomized assignment achieving 92.7% of the optimal fairness objective on the ICLR dataset when $q = 0.5$.

2.5.2 Bad-Assignment Probability Problem Variants

An input to both the Pairwise-Constrained Problem (Definition 2.1) and the Partition-Constrained Problem (Definition 2.3) is the matrix Q, where $Q_{r,p}$ denotes the maximum probability with which reviewer r should be assigned to paper p. In practice, program chairs can set the values in this matrix based on their own beliefs about each reviewer-paper pair. However, it may be difficult for program chairs to translate their beliefs about the risk of assigning any reviewer-paper pair into appropriate values for Q. In this section, we define alternate versions of these problems that allow the program chairs to codify their beliefs in a different way.

Define the assignment of reviewer r to paper p as "bad" if reviewer r intends to untruthfully review paper p (either because they intend to give a dishonest favorable review or because they intend to torpedo-review). Further define a matrix $W \in [0,1]^{m\times n}$ of bad-assignment probabilities, where $W_{r,p}$ represents the probability that the assignment of reviewer r to paper p would be

a bad assignment; we assume that the events of each reviewer-paper assignment being bad are all independent of each other. The "true value" of W may not be known, but it can be set based on the program chairs' beliefs about the reviewers and authors or potentially estimated based on some data from prior conferences. The problem variants we present in the following subsections make use of these bad-assignment probabilities.

We first consider the problem of limiting the probabilities of bad reviewer-paper assignments. We then consider the problem of limiting the probabilities that bad pairs of reviewers are assigned to the same paper.

Handling Bad Reviewer-Paper Assignments. We define an alternate version of the Pairwise-Constrained Problem using the bad-assignment probabilities:

Definition 2.5 (Bad-Assignment Probability Pairwise-Constrained Problem)**.** *The input to the problem is a similarity matrix S, a matrix $W \in [0,1]^{m \times n}$ of bad-assignment probabilities, and a value $\lambda \in [0,1]$. The goal is to find a randomized assignment of papers to reviewers that maximizes $\mathbb{E}\left[\sum_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}} S_{r,p} Z_{r,p}\right]$ subject to the constraints that $W_{r,p}\mathbb{P}[Z_{r,p} = 1] \leq \lambda, \forall r \in \mathcal{R}, p \in \mathcal{P}$.*

$W_{r,p}\mathbb{P}[Z_{r,p} = 1]$ is exactly the probability that both (i) reviewer r is assigned to paper p and (ii) this assignment is bad, so the constraints in the problem limit this at λ for all $r \in \mathcal{R}$ and $p \in \mathcal{P}$. This version of the Pairwise-Constrained Problem may be useful in practice if program chairs find it easier to set the values of W than they would for Q.

We now show how to solve the Bad-Assignment Probability Pairwise-Constrained Problem, by translating it to the original Pairwise-Constrained Problem. Suppose that we have access to the matrix F of marginal assignment probabilities that occur under some randomized assignment. The randomized assignment obeys our constraints if and only if $F_{r,p}W_{r,p} \leq \lambda, \forall r \in \mathcal{R}, p \in \mathcal{P}$. This observation leads to the following method of solving the Bad-Assignment Probability Pairwise-Constrained Problem:

- Transform the given instance of the Bad-Assignment Probability Pairwise-Constrained Problem into an instance of the Pairwise-Constrained Problem by constructing a matrix of maximum probabilities Q where

$$Q_{r,p} = \min\{\lambda / W_{r,p}, 1\} \qquad \forall r \in \mathcal{R}, p \in \mathcal{P}.$$

- Solve the Pairwise-Constrained Problem using the algorithm from Theorem 2.1, described in Section 2.2.

Handling Bad Pairs of Reviewers. Here, we first present an alternative version of the Partition-Constrained Problem and show how to solve it. We then present a different approach to handling the issue of bad reviewer pairs.

Constraints on Disjoint Reviewer Sets. In the same way as done above for the Pairwise-Constrained Problem, we define an alternate version of the Partition-Constrained Problem:

Definition 2.6 (Bad-Assignment Probability Partition-Constrained Problem)**.** *The input to the problem is a similarity matrix S, a matrix $W \in [0,1]^{m \times n}$ of bad-assignment probabilities, a value $\lambda \in [0,1]$, and a partition of the reviewer set into subsets $I_1, \ldots, I_d \subseteq \mathcal{R}$. The goal is to find a randomized assignment of papers to reviewers that maximizes $\mathbb{E}\left[\sum_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}} S_{r,p} Z_{r,p}\right]$*

subject to the constraints that $W_{r,p}\mathbb{P}[Z_{r,p} = 1] \leq \lambda, \forall r \in \mathcal{R}, p \in \mathcal{P}$ *and* $\mathbb{P}[Z_{a,p} = 1 \wedge Z_{b,p} = 1] = 0, \forall a, b \in I_i, \forall i \in [d]$.

Just as for the Bad-Assignment Probability Pairwise-Constrained Problem, we solve this problem by first transforming an instance of this problem into an equivalent instance of the Partition-Constrained Problem, done by constructing a matrix of maximum probabilities Q where $Q_{r,p} = \min(\lambda/W_{r,p}, 1), \forall r \in \mathcal{R}, p \in \mathcal{P}$. We then solve this instance using the algorithm in Section 2.3.2.

Constraints on the Expected Number of Bad Reviewers. The Bad-Assignment Probability Partition-Constrained Problem requires a partition of the reviewer set and prevents pairs of reviewers from being assigned to the same paper if they are in the same subset of this partition. Alternatively, one may want to prevent pairs of reviewers from being assigned to the same paper based on whether W indicates that they are both likely to be bad assignments on this paper, rather than based on some partition of the reviewer set. In this way, we now present an alternative approach to handling the issue of bad reviewer pairs, which does not require a partition of the reviewer set. Rather than explicitly constraining the probabilities of certain same-subset reviewer-reviewer-paper triples as in the Bad-Assignment Partition-Constrained Problem, we limit the *expected* number of bad reviewers on each paper.

The following problem states this goal:

Definition 2.7 (Bad-Assignment Probability Expectation-Constrained Problem)**.** *The input to the problem is a similarity matrix* S*, a matrix* $W \in [0,1]^{m \times n}$ *of bad-assignment probabilities, a value* $\lambda \in [0,1]$*, and a value* $\mu \in \mathbb{R}$*. The goal is to find a randomized assignment of papers to reviewers that maximizes* $\mathbb{E}\left[\sum_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}} S_{r,p} Z_{r,p}\right]$ *subject to the constraints that* $W_{r,p}\mathbb{P}[Z_{r,p} = 1] \leq \lambda, \forall r \in \mathcal{R}, p \in \mathcal{P}$ *and* $\sum_{r \in \mathcal{R}} W_{r,p}\mathbb{E}[M_{r,p}] \leq \mu, \forall p \in \mathcal{P}$.

We now present the algorithm that optimally solves this problem. The following LP, $\mathcal{LP}4$, finds a fractional assignment with expected number of bad reviewers on each paper no greater than μ:

$$\begin{aligned}
\underset{F \in \mathbb{R}^{m \times n}}{\arg\max} \quad & \sum_{p \in \mathcal{P}} \sum_{r \in \mathcal{R}} S_{r,p} F_{r,p} && && (2.14)\\
\text{subject to} \quad & 0 \leq F_{r,p} \leq 1 && \forall r \in \mathcal{R}, \forall p \in \mathcal{P} && (2.15)\\
& \sum_{p \in \mathcal{P}} F_{r,p} \leq \ell_r && \forall r \in \mathcal{R} && (2.16)\\
& \sum_{r \in \mathcal{R}} F_{r,p} = \ell_p && \forall p \in \mathcal{P} && (2.17)\\
& F_{r,p} W_{r,p} \leq \lambda && \forall r \in \mathcal{R}, \forall p \in \mathcal{P} && (2.18)\\
& \sum_{r \in \mathcal{R}} F_{r,p} W_{r,p} \leq \mu && \forall p \in \mathcal{P}. && (2.19)
\end{aligned}$$

Constraints (2.15-2.17) define the space of fractional assignment matrices, Constraint (2.18) ensures that the probability of each bad assignment occurring is limited at λ, and Constraint (2.19) ensures that the expected number of bad reviewer-paper assignments for each paper is at most μ. Therefore, $\mathcal{LP}4$ finds the optimal fractional assignment for the Bad-Assignment Probability

Expectation-Constrained Problem. This fractional assignment can then be sampled from using the sampling algorithm in Section 2.2.2.

The above approach to controlling bad reviewer pairs is not directly comparable to the approach taken earlier when solving the Bad-Assignment Probability Partition-Constrained Problem. The Bad-Assignment Probability Expectation-Constrained Problem indirectly restricts pairs of reviewers from being assigned to the same paper based on whether W indicates that they are both likely to be bad assignments on that paper, instead of based on a partition of the reviewer set. This could be advantageous if the sets of likely-bad reviewers for each paper (as given by the probabilities in W) are not expressed well by any partition of the reviewer set. However, handling suspicious reviewer pairs through constraining the expected number of bad reviewers per paper is weaker than directly constraining the probabilities of certain reviewer-reviewer-paper triples (as in the Bad-Assignment Probability Partition-Constrained Problem). First, it provides a guarantee only in expectation, and does not guarantee anything about the probabilities of the events we wish to avoid (that is, bad reviewer pairs being assigned to a paper). In addition, we here are assuming that the event of paper p and reviewer r being a bad assignment is independent of this event for all other reviewer-paper pairs; so, this method cannot address the issue of associations between reviewers, such as their presence at the same academic institution.

2.5.3 Decomposition Algorithm for the Pairwise-Constrained Problem

In Section 2.2, we provided the sampling algorithm that realizes Theorem 2.1, thus solving the Pairwise-Constrained Problem (Definition 2.1). We here provide a decomposition algorithm to compute a full distribution over deterministic assignments for a given fractional assignment matrix (which the prior work [25] does not). For simplicity, we assume here that all reviewer loads are met with equality (that is, $\sum_{p \in \mathcal{P}} F_{r,p} = \ell_r$ for all $r \in \mathcal{R}$); the extension to the case when reviewer loads are met with inequality is simple.

We first define certain concepts necessary for the algorithm. We then present a subroutine of the algorithm and prove its correctness. We then present the overall algorithm and prove its correctness. Finally, we analyze the time complexity of the algorithm.

Preliminaries. We define here three concepts used in the algorithm and its proof.

- A capacitated matching instance consists of a set of papers $\mathcal{P}$, a set of reviewers $\mathcal{R}$, and a capacity function $h : \mathcal{P} \cup \mathcal{R} \to \mathbb{Z}$. A solution to $(\mathcal{P}, \mathcal{R}, h)$ is a matrix $F \in [0,1]^{m \times n}$, where for any $p \in \mathcal{P}$,
$$\sum_{r \in \mathcal{R}} F_{r,p} = h(p),$$
and for any $r \in \mathcal{R}$,
$$\sum_{p \in \mathcal{P}} F_{r,p} = h(r).$$
The solution F is integral if $F_{r,p} \in \{0,1\}$ for all $p \in \mathcal{P}$ and $r \in \mathcal{R}$.
- For any $\mathcal{R}$ and $\mathcal{P}$, a maximum matching on a set $S \subseteq \mathcal{R} \times \mathcal{P}$ subject to capacities h is a set $M \subseteq S$ such that $\sum_{r \in \mathcal{R}} \mathbb{I}[(r,p) \in M] \leq h(p), \forall p \in \mathcal{P}$ and $\sum_{p \in \mathcal{P}} \mathbb{I}[(r,p) \in M] \leq h(r), \forall r \in \mathcal{R}$, and $|M|$ is maximized.

- For any $\mathcal{R}$ and $\mathcal{P}$, a perfect matching on a set $S \subseteq \mathcal{R} \times \mathcal{P}$ subject to capacities h is a maximum matching on S subject to h that additionally satisfies $\sum_{r \in \mathcal{R}} \mathbb{I}[(r,p) \in M] = h(p), \forall p \in \mathcal{P}$ and $\sum_{p \in \mathcal{P}} \mathbb{I}[(r,p) \in M] = h(r), \forall r \in \mathcal{R}$.

Decomposition subroutine. The following procedure, a subroutine of the overall algorithm, takes an instance $(\mathcal{P}, \mathcal{R}, h)$ and a solution to that instance F as input, and outputs an integral solution F_0 to $(\mathcal{P}, \mathcal{R}, h)$ with weight α_0 and a fractional solution F' to $(\mathcal{P}, \mathcal{R}, h)$ with strictly fewer fractional entries than F. Moreover, F, F_0, α_0, and F' satisfy $F = \alpha_0 F_0 + (1 - \alpha_0)F'$.

1. Let $E \subseteq \mathcal{R} \times \mathcal{P}$ be $E = \{(r,p) \mid F_{r,p} \in (0,1)\}$, and let $M_0 \subseteq \mathcal{R} \times \mathcal{P}$ be $M_0 = \{(r,p) \mid F_{r,p} = 1\}$. With this, define capacity function h' as, for any $p \in \mathcal{P}$,
$$h'(p) = h(p) - |\{(r,p) \mid r \in \mathcal{R}\} \cap M_0|$$
and for any $r \in \mathcal{R}$,
$$h'(r) = h(r) - |\{(r,p) \mid p \in \mathcal{P}\} \cap M_0|.$$
2. Find a maximum matching $M \subseteq E$ on E subject to capacity constraints h'.
3. Set F_0 as
$$(F_0)_{r,p} = \mathbb{I}\left[(r,p) \in M \cup M_0\right], \forall r \in \mathcal{R}, p \in \mathcal{P}.$$
Set F' as
$$F'_{r,p} = \frac{1}{(1-\alpha_0)}(F_{r,p} - \alpha_0 (F_0)_{r,p}), \forall r \in \mathcal{R}, p \in \mathcal{P}.$$
Set $\alpha_0 = \min(\{F_{r,p} \mid (r,p) \in M\} \cup \{1 - F_{r,p} \mid (r,p) \in E \setminus (M \cup M_0)\})$.

We prove the correctness of this subroutine in Lemma 2.3. Before we do, we restate a result from prior work [25] that we use in the proof, using our own notation.

Lemma 2.2 ([25, Thm. 1])**.** *For any $(\mathcal{P}, \mathcal{R}, h)$ and any solution F to $(\mathcal{P}, \mathcal{R}, h)$, there exists some $z \in \mathbb{Z}$, integral solutions $\{F_1, \ldots, F_z\}$ to $(\mathcal{P}, \mathcal{R}, h)$, and α lying on the z-dimensional simplex, such that $F = \sum_{i=1}^{z} \alpha_i F_i$.*

Now, the following lemma proves the correctness of the subroutine.

Lemma 2.3. *The decomposition subroutine finds F_0, α_0, and F', such that (i) F_0 is an integral solution to $(\mathcal{P}, \mathcal{R}, h)$, (ii) F' is a fractional solution to $(\mathcal{P}, \mathcal{R}, h)$, (iii) F' has strictly fewer fractional entries than F, and (iv) $F = \alpha_0 F_0 + (1 - \alpha_0)F'$.*

Proof. We first consider (i). The key step is to show that the maximum matching M found in step 2 is a perfect matching with respect to h', or equivalently, to show there is a perfect matching on E with respect to h'. Consider the capacitated matching instance $(\mathcal{P}, \mathcal{R}, h')$, and the solution F'' where
$$F''_{r,p} = \begin{cases} F_{r,p} & \text{if } F_{r,p} < 1 \\ 0 & \text{otherwise.} \end{cases}$$
F'' is a solution to $(\mathcal{P}, \mathcal{R}, h')$ by the construction of h'. By Lemma 2.2, F'' is a convex combination of integral solutions to $(\mathcal{P}, \mathcal{R}, h')$. For some z, let $\{F_1, \ldots, F_z\}$ and α be such a decomposition of F'', where each F_i is an integral solution to $(\mathcal{R}, \mathcal{P}, h')$ and α_i is its associated weight. For each $i \in [z]$, let $M_i \subseteq \mathcal{R} \times \mathcal{P}$ be the set of (r,p) pairs where $(F_i)_{r,p} = 1$. Since F_i is a solution to $(\mathcal{R}, \mathcal{P}, h')$, M_i is a perfect matching with respect to h'. By the definition of F'', $(r,p) \in E$ if

and only if $F''_{r,p} > 0$. Now since $F'' = \sum_{i=1}^{z} \alpha_i F_i$, $E = \bigcup_{i=1}^{z} M_i$. Since each M_i is a perfect matching with respect to h', E contains a perfect matching with respect to h' and so the maximum matching M found is in fact a perfect matching with respect to h'. Therefore, $M \cup M_0$ is a perfect matching with respect to h by the definition of h'. Therefore, F_0 is an integral solution to $(\mathcal{P}, \mathcal{R}, h)$.

For (ii), by the construction of F', all capacity constraints hold with equality. We only need to show that $F'_{r,p} \in [0, 1]$ for any (r, p). Consider any (r, p). There are 3 cases. If $(r, p) \in M_0$, then $F'_{r,p} = 1$. If $(r, p) \notin M \cup M_0$, then the choice of α_0 ensures that

$$F'_{r,p} = \frac{1}{(1-\alpha_0)} F_{r,p} \leq \frac{1}{(1-(1-F_{r,p}))} F_{r,p} = 1$$

and

$$F'_{r,p} = \frac{1}{(1-\alpha_0)} F_{r,p} \geq F_{r,p} \geq 0.$$

If $(r, p) \in M$, the choice of α_0 ensures that

$$F'_{r,p} = \frac{1}{(1-\alpha_0)} (F_{r,p} - \alpha_0) \geq \frac{1}{(1-\alpha_0)} (F_{r,p} - F_{r,p}) = 0$$

and

$$F'_{r,p} = \frac{1}{(1-\alpha_0)} (F_{r,p} - \alpha_0) \leq F_{r,p} \leq 1.$$

As a result, F' is a solution to $(\mathcal{P}, \mathcal{R}, h)$.

For (iii), the choice of α_0 ensures that at least one of the inequalities above achieves equality. That is, there exists (r, p) where $F_{r,p} \in (0, 1)$ such that $F'_{r,p} \in \{0, 1\}$.

Finally, (iv) holds by the construction of F_0 and F'. □

Overall algorithm. Using the above subroutine, the overall algorithm proceeds in the following recursive way. It takes as input a capacitated matching instance $(\mathcal{P}, \mathcal{R}, h)$ and a solution to that instance F. It outputs integral solutions $\{F_1, \ldots, F_z\}$ to $(\mathcal{P}, \mathcal{R}, h)$ and α lying on the z-dimensional simplex, such that $F = \sum_{i=1}^{z} \alpha_i F_i$.

1. If F is integral, return solution $\{F\}$ and weight 1.
2. Otherwise, decompose F into F_0 (with weight α_0) and F' using the above subroutine.
3. Recursively call this algorithm with $(\mathcal{P}, \mathcal{R}, h)$ and F' as input, decomposing F' into solutions $\{F_1, \ldots, F_z\}$ with weights α.
4. Define $\beta = (1-\alpha_0)\alpha$. Return the solutions $\{F_0, F_1, \ldots, F_z\}$ with weights $(\alpha_0, \beta_1, \ldots, \beta_z)$.

We now prove the correctness of this algorithm.

Theorem 2.5. *The decomposition algorithm correctly outputs integral solutions $\{F_1, \ldots, F_z\}$ to $(\mathcal{P}, \mathcal{R}, h)$ and α lying on the z-dimensional simplex, such that $F = \sum_{i=1}^{z} \alpha_i F_i$.*

Proof. We prove this statement by induction. If the algorithm returns in step 1, the theorem's statement holds. Now, assume that the theorem's statement holds for the decomposition returned by the recursive call to the algorithm in step 3, so that the following all hold: $\{F_1, \ldots, F_z\}$ are integral solutions to $(\mathcal{P}, \mathcal{R}, h)$, α lies on the z-dimensional simplex, and $F' = \sum_{i=1}^{z} \alpha_i F_i$. By

Lemma 2.3, F_0 is an integral solution to $(\mathcal{P}, \mathcal{R}, h)$, so the $z+1$ solutions returned in step 4 are integral solutions to $(\mathcal{P}, \mathcal{R}, h)$. Since $\alpha_0 \in [0,1]$, $\beta \in [0,1]^z$, and $\alpha_0 + \sum_{i=1}^{z} \beta_z = 1$, the weights returned in step 4 lie on the $z+1$ dimensional simplex. Finally, by Lemma 2.3,

$$\begin{aligned} F &= \alpha_0 F_0 + (1-\alpha_0) F' \\ &= \alpha_0 F_0 + (1-\alpha_0) \sum_{i=1}^{z} \alpha_i F_i \\ &= \alpha_0 F_0 + \sum_{i=1}^{z} \beta_i F_i. \end{aligned}$$

Therefore, the theorem's statement holds for the output of the algorithm in step 4. By induction, this proves the desired statement. □

This decomposition algorithm can be used as part of the algorithm that solves the Pairwise-Constrained Problem, substituting for the sampling algorithm described in Section 2.2.2. It finds the full decomposition of the fractional assignment matrix into deterministic assignments rather than sampling a deterministic assignment. The capacity function h used as the original input to the algorithm is defined as $h(r) = \ell_r, \forall r \in \mathcal{R}$ and $h(p) = \ell_p, \forall p \in \mathcal{P}$, and the input solution F is exactly the fractional assignment matrix found as the solution to $\mathcal{LP}1$. The output integral solutions represent deterministic assignments, and the corresponding weights represent the probability with which each assignment should be chosen.

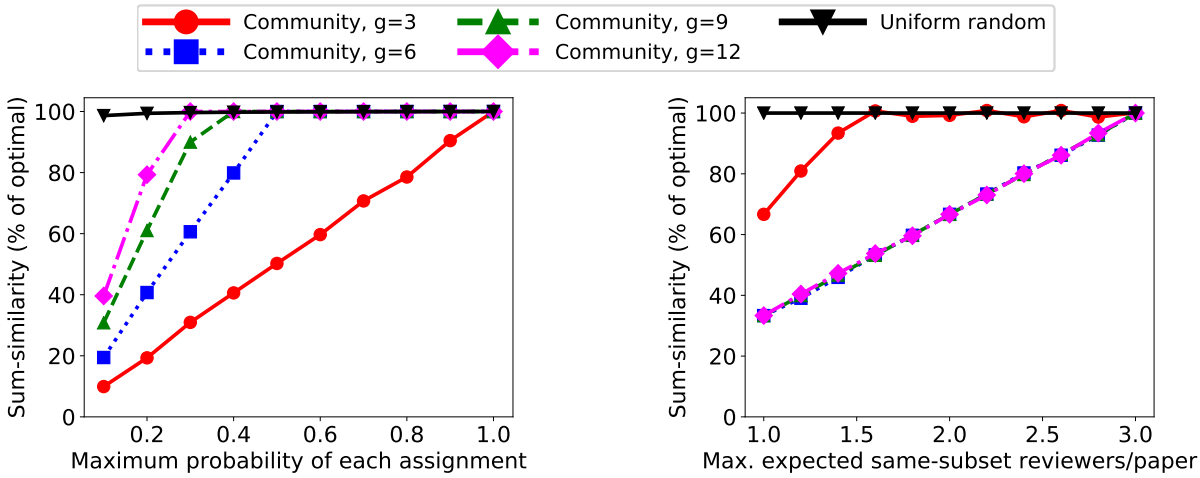
Time complexity. Since F' has at least one fewer fractional entry than F, the recursive procedure has depth $O(nm)$ and therefore makes $O(nm)$ calls to the decomposition subroutine. In each call, the bottleneck is finding a maximum matching on E subject to capacities h. This can be solved as a max-flow problem on a graph with $O(n+m)$ vertices and $O(nm)$ edges [32]. Using Dinic's algorithm [40], the computation of each matching takes $O(nm(n+m)^2)$ time, giving an overall time complexity of $O(n^2m^2(n+m)^2)$.

2.5.4 Synthetic Simulations

We now present experimental results on synthetic simulations. All results are averaged over 10 trials with error bars plotted representing the standard error of the mean, although error bars are sometimes not visible since the variance is low. All experiments were run on a computer with 8 cores and 16 GB of RAM, running Ubuntu 18.04 and solving the LPs with Gurobi 9.0.2 [101].

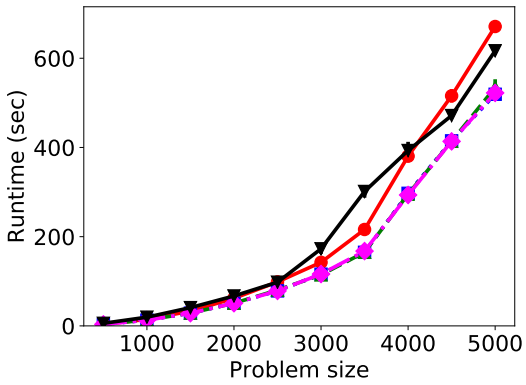
We consider two different simulations. First, we consider a simulated "community model" as used in past work [49]. In this model, $m = n = 360$ and $\ell_r = \ell_p = 3$; it is further parameterized by a group size g. For all $i \in \{0, g, 2g, \ldots, m\}$, reviewers i through $i+g-1$ have similarity 1 with papers i through $i+g-1$ and similarity 0 with all other papers. We consider four different group sizes g: 3, 6, 9, 12. We also consider a uniform random simulation, where each entry of the similarity matrix is independently and uniformly drawn from $[0,1)$, fixing $m = n = 1000$ and $\ell_r = \ell_p = 3$.

In Figure 2.5a, we examine the performance of our algorithm for the Pairwise-Constrained Problem. For each simulation, we set all entries of Q to a constant q and observe the sum-similarity as we vary q (on the x-axis). The objective value is reported here as a percentage of



(a) Pairwise-Constrained Problem

(b) Partition-Constrained Problem



(c) Runtime on Pairwise-Constrained Problem

Figure 2.5: Experimental results on synthetic simulations.

the optimal unconstrained solution's objective, as was done in Section 2.4. For the community models, the group size makes a large difference as to what an acceptable value of q is. For example, with group size 6 and $q = 0.5$, our algorithm will always assign all good reviewers to all papers; however, for any lower value of q it can no longer do this and so the objective deteriorates rapidly. Note that since our algorithm is optimal, this deterioration is due to the problem being overconstrained for low values of q and not due to an issue with the algorithm. For the uniform random simulation, our algorithm performs very well, since there are likely many reviewers with high similarity for each paper.

We also examine the performance of our algorithm for the Partition-Constrained Problem in Figure 2.5b. For each simulation, we fix $q = 0.5$ and gradually loosen Constraint (2.7) in $\mathcal{LP}2$ by increasing the constant from 1 to 3 in increments of 0.2, shown on the x-axis. We plot the sum-similarity objective of the resulting assignment, reported as a percentage of the optimal non-partition-constrained solution's objective (that is, the solution to the Pairwise-Constrained Problem with $q = 0.5$). For the community model simulations, we assign all reviewers in each

group to the same subset of the partition. Since all of the reviewers who can review each paper well are in the same subset, this presents a highly constrained problem (which our algorithm is solving optimally). As expected, our algorithm trades off the number of same-subset reviewer pairs assigned to the same paper and the sum-similarity objective rather poorly (as would any other algorithm). Since $q = 0.5$, there is no difference between the cases with group size 6 or greater. For the uniform random simulation, we assign random subsets of size 100. Since there are likely many reviewers with high similarity for each paper in different subsets, our algorithm again performs very well.

In Figure 2.5c, we show the runtime of our algorithm for the Pairwise-Constrained Problem on the various simulations, fixing $q = 0.5$ and varying $m = n$ on the x-axis. The runtime of our algorithm is similar across the different simulations. Our algorithm solves the uniform random simulation case with $m = n = 5000$ in just over 10 minutes.

2.6 Omitted Proofs

In this section, we present the proofs omitted from the previous sections.

Proofs of Theorem 2.2 and Corollary 2.1

Proof of Theorem 2.2. We first define a decision variant of the Triplet-Constrained Problem, called "Arbitrary-Constraint Feasibility." An instance of this problem is defined by the paper and reviewer loads ℓ_p and ℓ_r, and a 3-dimensional tensor $T \in [0,1]^{m \times m \times n}$. For all $i, j \in \mathcal{R}, i \neq j$ and for all $p \in \mathcal{P}$, $T_{i,j,p}$ denotes the maximum probability that both reviewers i and j are assigned to paper p. The question is: does there exist a randomized assignment that obeys the constraints given by T? We next show that Arbitrary-Constraint Feasibility is NP-hard by a reduction from 3-Dimensional Matching.

An instance of 3-Dimensional Matching consists of three sets A, B, C of size s, and a collection of tuples in $A \times B \times C$. It asks whether there exists a selection of s tuples that includes each element of A, B, and C exactly once. This problem is known to be NP-complete [79].

Given such an instance of 3-Dimensional Matching, we construct an instance of Arbitrary-Constraint Feasibility. Set loads of $\ell_p = 2$ reviewers per paper and $\ell_r = 1$ paper per reviewer. Consider $|A| + |B|$ reviewers (one for each element of $A \cup B$) and $|C|$ papers (one for each element of C). Define the tensor T to have $T_{i,j,p}$ equal to 1 if (i, j, p) is one of the tuples, and 0 otherwise.

We now show that a 3-Dimensional Matching instance is a yes instance (that is, the answer to it is "yes") if and only if the corresponding Arbitrary-Constraint Feasibility instance is a yes instance, thus proving that solving Arbitrary-Constraint Feasibility in polynomial time would allow us to solve 3-Dimensional Matching in polynomial time. If there exists a feasible reviewer-paper assignment in the corresponding Arbitrary-Constraint Feasibility instance, then we would answer yes for the original 3-Dimensional Matching instance; otherwise, if there does not exist a feasible reviewer-paper assignment, then we would answer no for the original 3-Dimensional Matching instance.

If the 3-Dimensional Matching instance is a yes (that is, there exists a valid selection of s tuples), then consider the paper assignment that assigns the corresponding reviewers and paper within each triple in the matching. Each paper has exactly 2 reviewers and each reviewer has

exactly 1 paper, so this is a deterministic assignment. Since it includes only the triples in the matching instance, it obeys the probability constraints of T, so the Arbitrary-Constraint Feasibility instance is a yes.

If the 3-Dimensional Matching instance is a no, then all choices of s tuples include some element of A, B, or C twice. If some element of C is chosen twice, then there must exist another element of C that is not included in any tuple. Therefore, any assignment of reviewer pairs to papers must either (a) include some reviewer-pair-to-paper assignment disallowed by T (i.e., an assignment not in the collection of tuples), (b) make less than s assignments of pairs to papers (and thus not assign to some paper), or (c) assign a reviewer twice or not assign some paper. So, no deterministic reviewer-paper assignment can meet the constraints of T. Now consider any randomized assignment, and select an arbitrary deterministic assignment in support of the randomized assignment. This deterministic assignment does not meet the constraints of T, so it must assign some reviewer r to some paper p that T requires to have probability 0. Therefore, since this deterministic assignment is in support, the randomized assignment assigns reviewer r to paper p with non-zero probability, thereby violating the constraints of T. Therefore, no randomized assignment can meet the constraints of T. Therefore, the Arbitrary-Constraint Feasibility instance is a no. This proves that Arbitrary-Constraint Feasibility is NP-hard.

Since even telling if the feasible region of randomized assignments is non-empty is NP-hard, optimizing any objective over this region is also NP-hard. Therefore, the Triplet-Constrained Problem is NP-hard.

Proof of Corollary 2.1. Suppose that the polytope of implementable reviewer-reviewer-paper probabilities could be expressed in a polynomial number of linear inequality constraints (with the reviewer-reviewer-paper probabilities as variables). An LP could then be constructed with these inequalities as well as the inequalities given by a tensor T of maximum reviewer-reviewer-paper probabilities. Solving this LP with any linear objective would then find a feasible point, solving Arbitrary-Constraint Feasibility. Since LPs can be solved in time polynomial in the number of variables and constraints, this is a contradiction unless $P \neq NP$.

Proof of Lemma 2.1 and Corollary 2.2

In Section 2.3.2, we described the sampling algorithm that realizes Lemma 2.1 and Corollary 2.2. Here, we present proofs of these results.

Proof of Lemma 2.1. We first prove part (i) of the lemma. Consider any subset I and any paper p, and recall that in Section 2.3.2 we showed that the algorithm presented there has the property that the total load on each paper from each subset is preserved exactly if originally integral and rounded in either direction if originally fractional. If the total load from subset I on paper p is less than or equal to 1 originally (i.e., $\sum_{r \in I} F_{r,p} \leq 1$), then this algorithm will only ever sample assignments with either 0 or 1 reviewers, so it never samples a integral assignment that assigns two reviewers from subset I to paper p.

We now prove part (ii) of the lemma. Suppose that the total load from subset I on paper p is originally strictly greater than 1 (i.e., $\sum_{r \in I} F_{r,p} > 1$). Let X denote a random variable that represents the number of reviewers from subset I on paper p, that is, $X = \sum_{r \in I} Z_{r,p}$. Hence, we have $\mathbb{E}[X] = \sum_{r \in I} F_{r,p} > 1$. Suppose that we implement the marginal probabilities F as a distribution over deterministic assignments that places zero mass on any deterministic

assignment where $X \geq 2$. Since X is integral in any deterministic assignment, all of the mass must be placed on deterministic assignments where $X \leq 1$. Since $\mathbb{E}[X] > 1$, this is impossible. Therefore, F cannot be implemented without having some probability of placing two reviewers from subset I on paper p, so the expected number of pairs of reviewers from subset I assigned to paper p must be non-zero for any sampling algorithm.

Proof of Corollary 2.2. We now show that the distribution sampled from by the algorithm realizing Lemma 2.1 minimizes the expected number of pairs of reviewers from each subset assigned to each paper. Consider any subset I and paper p, and again let X denote a random variable that represents the number of reviewers from subset I on paper p. The expected number of pairs of reviewers from subset I assigned to paper p is $\mathbb{E}\left[\binom{X}{2}\right] = \frac{1}{2}\mathbb{E}[X^2] - \frac{1}{2}\mathbb{E}[X]$. Since $\mathbb{E}[X]$ is fixed for a given F, we must only show that our chosen decomposition minimizes $\mathbb{E}[X^2]$.

Let f be the probability mass function of X under the distribution of X produced by our sampling algorithm, so that $f(i) = P[X = i]$ for $i \in \{0, \ldots, |I|\}$. Let f' be the probability mass function of X under any different distribution produced by some sampling algorithm, so that $\exists i \in \{0, \ldots, |I|\}$ such that $f'(i) \neq f(i)$. Since both f and f' are produced by sampling algorithms, they must respect the marginal assignment probabilities given by F.

First, assume that $\mathbb{E}[X] = \mu$ is integral. $\mathbb{E}[X] = \sum_{r \in I} F_{r,p}$, so μ is equal to the total load from subset I on paper p. From Section 2.3.2, our sampling algorithm preserves exactly the loads from any subset on any paper that are originally integral, meaning that it will always assign exactly μ reviewers from subset I to paper p. In other words, our sampling algorithm always gives the distribution of X where $f(\mu) = 1$ and $f(i) = 0$ for $i \neq \mu$. Since all distributions of X have the same expectation, $\sum_{i=0}^{|I|} f'(i)i = \mu$; we also know that $f'(i) > 0$ for some $i \neq \mu$. For this distribution, we have that

$$\mathbb{E}_{f'}[X^2] = \sum_{i=0}^{|I|} f'(i)i^2 = \sum_{\Delta=-\mu}^{|I|-\mu} f'(\mu+\Delta)(\mu+\Delta)^2 = \mu^2 + \sum_{\Delta=-\mu}^{|I|-\mu} f'(\mu+\Delta)\Delta^2 > \mu^2 = \mathbb{E}_f[X^2].$$

Now, suppose that $\mathbb{E}[X] = \mu$ is not integral. From Section 2.3.2, our sampling algorithm rounds to a neighboring integer the loads from any subset on any paper that are originally not integral, meaning that it will always assign exactly $\lceil \mu \rceil$ or $\lfloor \mu \rfloor$ reviewers from subset I to paper p. In other words, our sampling algorithm only places probability mass on outcomes $X = \lceil \mu \rceil$ or $X = \lfloor \mu \rfloor$, so $f(i) = 0$ for $i \notin \{\lceil \mu \rceil, \lfloor \mu \rfloor\}$. There is only one way to do this so that $\mathbb{E}[X] = \mu$; exactly $f(\lceil \mu \rceil) = \mu - \lfloor \mu \rfloor$ and $f(\lfloor \mu \rfloor) = \lceil \mu \rceil - \mu$. Then under this distribution, via some algebraic simplifications,

$$\begin{aligned} \mathbb{E}_f[X^2] &= f(\lceil \mu \rceil)\lceil \mu \rceil^2 + f(\lfloor \mu \rfloor)\lfloor \mu \rfloor^2 \\ &= -\lceil \mu \rceil^2 + \lceil \mu \rceil - \mu + 2\lceil \mu \rceil \mu. \end{aligned} \tag{2.20}$$

Under any other distribution of X giving the probability mass function f',

$$\mathbb{E}_{f'}[X^2] = \sum_{i=0}^{|I|} f'(i)i^2$$

$$= \sum_{\Delta=-\lceil\mu\rceil}^{|I|-\lceil\mu\rceil} \left(f'(\lceil\mu\rceil+\Delta)(\lceil\mu\rceil+\Delta)^2\right) + 2\lceil\mu\rceil \sum_{\Delta=-\lceil\mu\rceil}^{|I|-\lceil\mu\rceil} (f'(\lceil\mu\rceil+\Delta)\Delta) + \sum_{\Delta=-\lceil\mu\rceil}^{|I|-\lceil\mu\rceil} \left(f'(\lceil\mu\rceil+\Delta)\Delta^2\right)$$

$$= \lceil\mu\rceil^2 + 2\lceil\mu\rceil(\mu - \lceil\mu\rceil) + \sum_{\Delta=-\lceil\mu\rceil}^{|I|-\lceil\mu\rceil} \left(f'(\lceil\mu\rceil+\Delta)\Delta^2\right). \tag{2.21}$$

We want to show that $\mathbb{E}_{f'}[X^2] > \mathbb{E}_f[X^2]$. From (2.20) and (2.21), it remains to show that

$$\sum_{i=0}^{|I|} f'(i)(i - \lceil\mu\rceil)^2 > \lceil\mu\rceil - \mu.$$

Note that because $f'(i) \neq f(i)$ for some i, there exists some $j \notin \{\lceil\mu\rceil, \lfloor\mu\rfloor\}$ such that $f'(j) > 0$. Further, $(i - \lceil\mu\rceil)^2 \geq (\lceil\mu\rceil - i)$ for all integers i and $(i - \lceil\mu\rceil)^2 > (\lceil\mu\rceil - i)$ for all integers $i \notin \{\lceil\mu\rceil, \lfloor\mu\rfloor\}$. Therefore,

$$\sum_{i=0}^{|I|} f'(i)(i - \lceil\mu\rceil)^2 > \sum_{i=0}^{|I|} f'(i)(\lceil\mu\rceil - i) = \lceil\mu\rceil - \sum_{i=0}^{|I|} f'(i)i = \lceil\mu\rceil - \mu.$$

Therefore, $\mathbb{E}_{f'}[X^2] > \mathbb{E}_f[X^2]$ as desired, so f is the probability mass function corresponding to the distribution of X which minimizes $\mathbb{E}[X^2]$ (uniquely, since the inequality is strict). This concludes the proof that our algorithm minimizes $\mathbb{E}[X^2]$ and therefore minimizes the expected number of pairs from the same subset assigned to the same paper.

2.7 Discussion

We have presented here a framework and a set of algorithms for addressing three challenges of practical importance to the peer review process: reviewer-author collusion, torpedo reviewing, and reviewer de-anonymization on the release of assignment data. By design, our algorithms are quite flexible to the needs of the program chairs, depending on which challenges they are most concerned with addressing. Our empirical evaluations demonstrate some of the tradeoffs that can be made between total similarity and maximum probability of each paper-reviewer pair or between total similarity and number of reviewers from the same subset on the same paper. The exact parameters of the algorithm can be set based on how the program chairs weigh the relative importance of each of these factors. Note that an empirical evaluation of exactly how much our algorithm reduces manipulation in a real conference is not possible, since the ground truth of which reviewers were manipulating their assignments is not known.

This work leads to a number of open problems of interest. First, since the general Triplet-Constrained Problem is NP-hard, we considered one special structure—the Partition-Constrained Problem—of practical relevance. A direction for future research is to find additional special

cases under which optimizing over constraints on the probabilities of reviewer-pair-to-paper assignments is feasible. For example, there may be a known network of reviewers where program chairs wish to prevent connected reviewers from being assigned to the same paper. A second problem of interest is to develop methods to detect potentially malicious reviewer-paper pairs before papers are assigned (e.g., based on the bids), which we begin to examine in Chapter 7. Third, this work does not address the problem of reviewers colluding with each other to give dishonest favorable reviews after being assigned to each others' papers. We discuss this problem further in Chapter 9, but ultimately leave this issue for future work.

The randomized assignment algorithms in this chapter focus primarily on maximizing an expected total similarity objective (although we consider a variant with a fairness objective in Section 2.5.1). However, additional objectives beyond just the total similarity are often of interest to conference organizers in practice. One such objective might be the total similarity of papers within each of a conference's various subject areas. For example, one may be concerned that choosing a single value for Q based on the total similarity objective could significantly harm the quality of paper assignments in smaller subject areas. In this case, program chairs may want to partition the papers by subject area and choose a different value of Q for each subject area. Developing a principled and efficient method for choosing such values is an important practical question. Another objective of interest may be a notion of "topic coverage": that each paper's assigned reviewers collectively have expertise in as many of the paper's subject areas as possible. If the total similarity objective that we consider is modified to incorporate topic coverage in some way, it may be possible to develop a variant of our randomized assignment algorithm for this modified objective using tools from submodular optimization.

Following the publication of the work in this chapter, our work [161] presented an alternative approach to randomizing paper assignments that builds upon our randomized assignment algorithm. Rather than maximizing expected similarity subject to constraints on the assignment probabilities, this work perturbs the objective function such that it is concave in each assignment probability. In this way, the proposed assignment algorithm can distinguish between multiple similar-quality assignments with the same maximum probability, adding additional randomness into the assignment at low cost. From another perspective, this algorithm can be viewed as implicitly adapting the probability limits for different reviewer-paper pairs based on the cost in similarity. In this way, this algorithm helps to address the concern about disproportionate subject-area impact described in the previous paragraph.

Impact. Our proposed randomized assignments have been deployed by real computer-science conferences for the purpose of improving robustness against reviewer-author collusion. The basic randomized assignment algorithm defined in Section 2.2 was used by the AAAI 2022 and 2023 conferences, the 2023 ACM Conference on Knowledge Discovery and Data Mining (KDD), as well as by other smaller venues (e.g., the 2021 Workshop on Theory and Practice of Differential Privacy). For context, these AAAI conferences received 9020 and 8536 submissions respectively, making them some of the largest conferences in the field of artificial intelligence. The randomized assignment algorithm in Section 2.2 has also been implemented for use by any future conference at the popular conference-management site `OpenReview.net`.

Chapter 3

Leveraging Randomization to Evaluate Counterfactual Paper Assignment Policies

In Chapter 2, we presented an algorithm to compute randomized paper assignments, thus providing robustness to reviewer-author collusion, torpedo reviewing, and reviewer de-anonymization. To combat these forms of undesirable behavior, conference organizers have adopted and deployed these randomized assignments in real conferences. In this chapter, we propose a technique to "harvest" [94] the randomness introduced in these paper assignments to perform off-policy evaluation, enabling the comparison of many alternative paper assignment policies in terms of a measure of review quality. As one example, we can compare how review quality varies with the level of randomization in the policy, allowing us to evaluate the cost of deploying our randomized assignment algorithm.

These techniques address a persistent challenge in the design of effective peer-review systems: evaluating how changes to peer-review assignment algorithms affect review quality. An implicit assumption underlying the standard paper assignment approach introduced in Chapter 1 is that review quality is an increasing function of bid enthusiasm, text similarity, and subject area match, but how to combine these signals into a score is approached via heuristics. Researchers typically observe only the reviews actually assigned by the algorithm and have no way of measuring the quality of reviews under an assignment generated by an alternative algorithm.

One approach to comparing different paper assignment policies is running randomized control trials or A/B tests. Several conferences (NeurIPS'14 [93, 124], WSDM'17 [148], ICML'20 [142], and NeurIPS'21 [17]) have run A/B tests to evaluate various aspects of their review process, such as differences between single- vs. double-blind review. However, such experiments are extremely costly in the peer review context, with the NeurIPS experiments requiring a significant number of additional reviews, overloading already strained peer review systems. Moreover, A/B tests typically compare only a handful of design decisions, while assignment algorithms typically require making many such decisions (see Section 3.1).

The key insight of this methodology is that under a randomized assignment policy, a range of reviewer-paper pairs other than the exactly optimal assignment become probable to observe. We can then adapt the tools of off-policy evaluation and importance sampling to evaluate the quality of many alternative policies. A major challenge, however, is that off-policy evaluation assumes overlap between the on-policy and the off-policy, i.e., that each reviewer-paper assignment that

has a positive probability under the off-policy also had a positive probability under the on-policy. In practice, positivity violations are inevitable even when the maximum probability of assigning any reviewer-paper pair is low enough to induce significant randomization, especially as we are interested in evaluating a wide range of design choices of the assignment policy. To address this challenge, we build on existing literature for partial identification and propose methods that bound the off-policy estimates while making weak assumptions on how positivity violations arise.

More specifically, we propose two approaches for analysis that rely on different assumptions on the mapping between the covariates (e.g., bid, text similarity, subject area match) and the outcome (e.g., review quality) of the reviewer-paper pairs. First, we assume *monotonicity* in the covariates-outcome mapping. Understood intuitively, this assumption states that if reviewer-paper pair i has higher or equal bid, text similarity, and subject area match than a reviewer-paper pair j, then we assume that the quality of the review for pair i is higher or equal to the review for pair j. Alternatively, we assume *Lipschitz smoothness* in the covariate-outcome mapping. Intuitively, this assumption captures the idea that two reviewer-paper pairs that have similar bids, text similarity, and subject area match, should result in a similar review quality. We find that this Lipschitz assumption naturally generalizes so-called *Manski bounds* [104], the partial identification strategy that assumes only bounded outcomes.

We apply our methods to data collected by two computer science venues that used randomized assignment policies: the 2021 Workshop on Theory and Practice of Differential Privacy (*TPDP*) with 95 papers and 35 reviewers, and the 2022 AAAI Conference on Advancement in Artificial Intelligence (*AAAI*) with 8450 papers and 3145 reviewers. TPDP is an annual workshop co-located with the machine learning conference ICML, and AAAI is one of the largest annual artificial intelligence conferences. We evaluate two design choices: (*i*) how varying the weights of the bids vs. text similarity vs. subject area match (latter available only in AAAI) affects the overall quality of the reviews, and (*ii*) the "cost of randomization", i.e., how much the review quality decreased as a result of introducing randomness in the assignment. As our measure of assignment quality, we consider the expertise and confidence reported by the reviewers for their assigned papers. We find that our proposed methods for partial identification assuming monotonicity and Lipschitz smoothness significantly reduce the bounds of the estimated review quality off-policy, leading to more informative results. Substantively, we find that placing a larger weight on text similarity results in higher review quality, and that introducing randomization in the assignment leads to a very small reduction in review quality.

The greater understanding of the relationship between the paper assignment policy and the resulting review quality enabled by these techniques helps to further address issues of undesirable behavior in two ways. First, analysis of the "cost of randomization" using our methodology allows conference program chairs to more accurately understand the tradeoffs involved in deploying randomization and helps them to choose an appropriate level of randomization for future iterations of their conferences. A better understanding of these tradeoffs should give program chairs more confidence in deploying randomized assignments to guard against malicious behavior. Beyond just randomized assignments, the cost of deploying any other proposed defense against malicious behavior can be evaluated via our methodology. Second, our methods allow program chairs to improve the review quality of their conferences year over year (e.g., by evaluating alternative functions to compute similarities). By reducing the number of cases

where reviewers are assigned to review papers for which they have low expertise (according to the reviewer's own evaluation), we may be able to reduce the number of missing and low-effort reviews.

Beyond our contributions to the design and study of peer review systems, the methods proposed in this chapter should also apply to other matching systems such as recommendation systems [56, 130, 133], advertising [21], and ride-sharing assignment systems [158]. Further, our contributions to off-policy evaluation under partial identification should be of independent interest.

In Section 3.1, we introduce the notation and concepts referenced in this chapter. In Section 3.2, we discuss the fundamentals of off-policy evaluation. In Section 3.3, we describe the methods and assumptions we propose to address issues with applying standard techniques for off-policy evaluation. In Section 3.4, we introduce our experimental setup. In Section 3.5, we show the results of our analysis. This chapter is based on work done jointly with my collaborators [132]. Our code is available at `https://github.com/msaveski/counterfactual-peer-review`.

3.1 Preliminaries

We start by reviewing the fundamentals of peer-review assignment algorithms.

Reviewer-Paper Similarity. Consider a peer review scenario with a set of reviewers $\mathcal{R}$ and a set of papers $\mathcal{P}$. As introduced in Chapter 1, standard assignment algorithms for large-scale peer review rely on "similarity scores" for every reviewer-paper pair $i = (r, p) \in \mathcal{R} \times \mathcal{P}$, representing the assumed quality of review by that reviewer for that paper. These scores S_i, typically non-negative real values, are commonly computed from a combination of up to three sources of information:

- T_i: text-similarity between each paper and reviewer's past work, using various techniques [29, 100, 110, 113, 127, 149];
- K_i: overlap between the subject areas selected by each reviewer and each paper's authors; and
- B_i: reviewer-provided "bids" on each paper.

Without any principled methodology for evaluating the choice of similarity score, conference organizers manually select a parametric functional form and choose parameter values by spot-checking a few reviewer-paper assignments. For example, a simple similarity function is a convex combination of the component scores: $S_i = w_{\text{text}} T_i + (1 - w_{\text{text}}) B_i$. Conferences have also used more complex non-linear functions: NeurIPS'16 [136] used the functional form $S_i = (0.5T_i + 0.5K_i)2^{B_i}$, while AAAI'21 [96] used $S_i = (0.5T_i + 0.5K_i)^{1/B_i}$. Beyond the choice of how to combine the component scores, numerous other aspects of the similarity computation also imply choices: the language-processing techniques used to compute text-similarity scores, the input given to them, the range and interpretation of bid options shown to reviewers, etc. The range of possible functional forms results in a wide design space, which we explore in this work.

Deterministic Assignment. Let $Z \in \{0, 1\}^{m \times n}$ be an assignment matrix where Z_i denotes whether the reviewer-paper pair i was assigned or not. Given a matrix of reviewer-paper similarity scores $S \in \mathbb{R}_{\geq 0}^{m \times n}$, a standard objective is to find an assignment of reviewers to papers that

maximizes the sum of similarities of the assigned pairs, subject to constraints that each paper is assigned to an appropriate number of reviewers, each reviewer is assigned no more than a maximum number of papers, and conflicts of interest are respected [29, 30, 52, 59, 102, 144, 145]. This optimization problem can be formulated as a linear program. We provide a detailed formulation in Section 3.6.1. While other objective functions have been proposed [36, 85, 138], here we focus on the sum-of-similarities objective.

Randomized Assignment. In Chapter 2, we introduce the idea of using randomization to prevent colluding reviewers and authors from being able to guarantee their assignments. Specifically, the program chairs first choose a parameter $q \in [0, 1]$. Then, the algorithm computes a randomized paper assignment, where the marginal probability $\mathbb{P}[Z_i = 1]$ of assigning any reviewer-paper pair i is at most q. These marginal probabilities are determined by a linear program, which maximizes the expected similarity of the assignment subject to the probability limit q (detailed formulation in Section 3.6.1). A reviewer-paper assignment is then sampled using a randomized procedure that iteratively redistributes the probability mass placed on each reviewer-paper pair until all probabilities are either zero or one.

Review Quality. The above assignments are chosen based on maximizing the (expected) similarities of assigned reviewer-paper pairs, but those similarities may not be accurate proxies for the quality of review that the reviewer can provide for that paper. In practice, automated similarity-based assignments result in numerous complaints of low-expertise paper assignments from both authors and reviewers [71], and recent work [143] finds that current text-similarity algorithms make significant errors in predicting reviewer expertise. Meanwhile, self-reported assessments of reviewer-paper assignment quality can be collected from the reviewers themselves after the review. Conferences often ask reviewers to score their *expertise* in the paper's topic and/or *confidence* in their review [96, 136, 141]. Other indicators of review quality can also be considered; e.g., some conferences ask "meta-reviewers" or other reviewers to evaluate the quality of written reviews directly [9, 141]. Indicators of review quality could also potentially be constructed from features of the review that may be correlated with quality, such as the length of the review. In this work, we consider self-reported expertise and confidence as our measures of review quality.

3.2 Off-Policy Evaluation

One attractive property of the randomized assignment described above is that while only one reviewer-paper assignment is sampled and deployed, many other assignments could have been sampled, and those assignments could equally well have been generated by some alternative assignment policy. The positive probability of other assignments allows us to investigate whether alternative assignment policies might have resulted in higher-quality reviews.

Let A be a randomized assignment policy with a probability density $\mathbb{P}_A$, where $\sum_{Z \in \{0,1\}^{m \times n}} \mathbb{P}_A(Z) = 1$; $\mathbb{P}_A(Z) \geq 0, \forall Z$; and $\mathbb{P}_A(Z) > 0$ only for feasible assignments Z. Let B be another policy with density $\mathbb{P}_B$, defined similarly. We denote by $\mathbb{P}_A(Z_i)$ and $\mathbb{P}_B(Z_i)$ the marginal probabilities of assigning reviewer-paper pair i under A and B respectively. Finally, let $Y_i \in \mathbb{R}$, where $i = (r, p) \in \mathcal{R} \times \mathcal{P}$, be the measure of the quality of reviewer r's review of paper p, e.g., reviewer self-reported expertise or confidence as introduced in Section 3.1.

We follow the potential outcomes framework for causal inference [129]. Throughout this

work, we will let A be the on-policy or the logging policy, i.e., the policy that the review data was collected under, while B will denote one of several alternative policies of interest. In Section 3.5, we will describe the specific alternative policies we consider in this work. Define $N = \sum_{i \in \mathcal{R} \times \mathcal{P}} Z_i$ as the total number of reviews, fixed across policies and set ahead of time. We are interested in the following estimands:

$$\mu_A = \mathbb{E}_{Z \sim \mathbb{P}_A}\left[\frac{1}{N}\sum_{i \in \mathcal{R}\times\mathcal{P}} Y_i Z_i\right], \quad \mu_B = \mathbb{E}_{Z \sim \mathbb{P}_B}\left[\frac{1}{N}\sum_{i \in \mathcal{R}\times\mathcal{P}} Y_i Z_i\right],$$

where μ_A and μ_B are the expected review quality under policy A and B, respectively.

In practice, we do not have access to all Y_i, but only those that were assigned. Let $Z^A \in \{0,1\}^{m \times n}$ be the assignment sampled under the on-policy A, drawn from $\mathbb{P}_A$. We define the following Horvitz-Thompson estimators of the means:

$$\begin{aligned}
\widehat{\mu}_A &= \frac{1}{N}\sum_{i \in \mathcal{R}\times\mathcal{P}} Y_i Z_i^A, \\
\widehat{\mu}_B &= \frac{1}{N}\sum_{i \in \mathcal{R}\times\mathcal{P}} Y_i Z_i^A W_i, \quad \text{where } W_i = \frac{\mathbb{P}_B(Z_i)}{\mathbb{P}_A(Z_i)} \; \forall i \in \mathcal{R}\times\mathcal{P}.
\end{aligned} \tag{3.1}$$

For now, suppose that B has positive probability only where A is positive (also known as satisfying "positivity"): $\mathbb{P}_A(Z_i) > 0$ for all $i \in \mathcal{R} \times \mathcal{P}$ where $\mathbb{P}_B(Z_i) > 0$. Then, all weights W_i where $\mathbb{P}_B(Z_i) > 0$ are bounded. As we will see, many policies of interest B go beyond the support of A.

Under the positivity assumption, $\widehat{\mu}_A$ and $\widehat{\mu}_B$ are unbiased estimators of μ_A and μ_B respectively [68]. Moreover, the Horvitz-Thompson estimator is admissible in the class of all unbiased estimators [57]. Note that $\widehat{\mu}_A$ is simply the empirical mean of the observed assignment sampled on-policy, and $\widehat{\mu}_B$ is a weighted mean of the observed assignment based on inverse probability weighting: placing weights greater than one on reviewer-paper pairs that are more likely off- than on-policy and less than or equal to one otherwise. These estimators also rely on a standard causal inference assumption of "no interference"; i.e., that the outcomes Y_i do not depend on the assignments Z_j^A for any other reviewer-paper pair $j \neq i$. In Section 3.6.2, we discuss the implications of this assumption in the peer review context.

Challenges. In off-policy evaluation, we are interested in evaluating a policy B based on data collected under policy A. However, our ability to do so is typically limited to policies where the assignments that would be made under B are possible under A. In practice, many interesting policies step outside of the support of A. Outcomes for reviewer-paper pairs outside the support of A but with positive probability under B ("positivity violations") cannot be estimated and must either be imputed by some model or have their contribution to the average outcome (μ_B) bounded.

In addition to positivity violations, we identify three other mechanisms through which missing data with potential confounding may arise in the peer review context: absent reviewers, selective attrition, and manual reassignments. For absent reviewers, i.e., reviewers who have not submitted *any* reviews, we do not have a reason to believe that the reviews are missing due to

the quality of the reviewer-paper assignment. Hence, we assume that their reviews are missing at random, and impute them with the weighted mean outcome of the observed reviews. For selective attrition, i.e., when some but not all reviews are completed, we instead employ conservative bounding techniques as for policy-based positivity violations. Finally, reviews might be missing due to manual reassignments by the program chairs, after the assignment has been sampled. As a result, the originally assigned reviews will be missing and new reviews will be added. In such cases, we treat removed assignments as attrition (i.e., bounding their contribution) and ignore the newly introduced assignments as they did not arise from any determinable process.

Concretely, we partition the reviewer-paper pairs into the following (mutually exclusive and exhaustive) sets:

- $\mathcal{I}^{-}$: positivity violations, $\{i = (r, p) \in \mathcal{R} \times \mathcal{P} : \mathbb{P}_A(Z_i) = 0 \wedge \mathbb{P}_B(Z_i) > 0\}$,
- $\mathcal{I}^{Abs}$: missing reviews where the reviewer was absent (submitted no reviews),
- $\mathcal{I}^{Att}$: remaining missing reviews, and
- $\mathcal{I}^{+}$: remaining pairs without positivity violations or missing reviews, $(\mathcal{R} \times \mathcal{P}) \setminus (\mathcal{I}^{Att} \cup \mathcal{I}^{Abs} \cup \mathcal{I}^{-})$.

In the next section, we present methods for imputing or bounding the contribution of $\mathcal{I}^{-}$ to the estimate of $\widehat{\mu}_B$, and $\mathcal{I}^{Abs}$ and $\mathcal{I}^{Att}$ to the estimates of $\widehat{\mu}_A$ and $\widehat{\mu}_B$.

3.3 Imputation and Partial Identification

In the previous section, we defined three sets of reviewer-paper pairs i for which outcomes Y_i must be imputed rather than estimated: $\mathcal{I}^{-}$, $\mathcal{I}^{Abs}$, $\mathcal{I}^{Att}$. In this section, we describe varied methods for imputing these outcomes that rely on different strengths of assumptions, including methods that output point estimates (Sections 3.3.1 and 3.3.2) and methods that output lower and upper bounds of $\widehat{\mu}_B$ (Sections 3.3.3 and 3.3.4). In Section 3.5, we apply these methods to peer-review data from two computer science venues.

For missing reviews where the reviewer is absent ($\mathcal{I}^{Abs}$), we assume that the reviewer did not participate in the review process for reasons unrelated to the assignment quality (e.g., too busy). Specifically, we assume that the reviewers are missing at random and thus impute the mean outcome among $\mathcal{I}^{+}$, the pairs with no positivity violations or missing reviews:

$$\overline{Y} = \frac{\sum_{i \in \mathcal{I}^{+}} Y_i Z_i^A W_i}{\sum_{i \in \mathcal{I}^{+}} Z_i^A W_i}.$$

Correspondingly, we set $Y_i = \overline{Y}$ for all $i \in \mathcal{I}^{Abs}$ in estimator (3.1).

In contrast, for positivity violations ($\mathcal{I}^{-}$) and the remaining missing reviews ($\mathcal{I}^{Att}$), we allow for the possibility that these reviewer-paper pairs being unobserved is correlated with their unobserved outcome. Thus, we consider imputing arbitrary values for i in these subsets, which we denote by Y_i^{Impute} and place into a matrix $Y^{Impute} \in \mathbb{R}^{m \times n}$, leaving entries for $i \notin \mathcal{I}^{-} \cup \mathcal{I}^{Att}$ undefined. This strategy corresponds to setting $Y_i = Y_i^{Impute}$ for $i \in \mathcal{I}^{-} \cup \mathcal{I}^{Att}$ in estimator (3.1). To obtain bounds, we impute both the assumed minimal and maximal values of Y_i^{Impute}.

These modifications result in a Horvitz-Thompson off-policy estimator with imputation. To denote this, we redefine $\widehat{\mu}_B$ to be a function $\widehat{\mu}_B : \mathbb{R}^{m \times n} \to \mathbb{R}$, where $\widehat{\mu}_B(Y^{Impute})$ denotes the

estimator resulting from imputing entries from a particular choice of Y^{Impute}:

$$\widehat{\mu}_B(Y^{Impute}) = \frac{1}{N}\left(\sum_{i\in\mathcal{I}^+} Y_i Z_i^A W_i + \sum_{i\in\mathcal{I}^{Att}} Y_i^{Impute} Z_i^A W_i + \sum_{i\in\mathcal{I}^{Abs}} \overline{Y} Z_i^A W_i + \sum_{i\in\mathcal{I}^-} Y_i^{Impute}\mathbb{P}_B(Z_i)\right).$$

The estimator computes the weighted mean of the observed (Y_i) and imputed outcomes (Y_i^{Impute} and $\overline{Y}$). We impute Y_i^{Impute} for the attrition ($\mathcal{I}^{Att}$) and positivity violation ($\mathcal{I}^-$) pairs, and $\overline{Y}$ for the absent reviewers ($\mathcal{I}^{Abs}$). Note that we weight the imputed positivity violations ($\mathcal{I}^-$) by $\mathbb{P}_B(Z_i)$ rather than $Z_i W_i$, since the latter is undefined. Under the assumption that the imputed outcomes are accurate, $\widehat{\mu}_B(Y^{Impute})$ is an unbiased estimator of μ_B.

To construct confidence intervals, we estimate the variance of $\widehat{\mu}_B(Y^{Impute})$ as follows:

$$\widehat{\text{Var}}[\widehat{\mu}_B(Y^{Impute})] = \frac{1}{N^2}\sum_{(i,j)\in(\mathcal{R}\times\mathcal{P})^2} \text{Cov}[Z_i, Z_j] Z_i^A Z_j^A W_i W_j Y_i' Y_j',$$

$$\text{where } Y_i' = \begin{cases} Y_i & \text{if } i \in \mathcal{I}^+ \\ Y_i^{Impute} & \text{if } i \in \mathcal{I}^{Att} \cup \mathcal{I}^- \\ \overline{Y} & \text{if } i \in \mathcal{I}^{Abs}. \end{cases}$$

The covariance terms (taken over $Z \sim \mathbb{P}_A$) are not known exactly, owing to the fact that the randomized assignment algorithm only constrains the marginal probabilities of individual reviewer-paper pairs, but pairs of pairs can be non-trivially correlated. In the absence of a closed-form expression, we use Monte Carlo methods to tightly estimate these covariances (further details provided in Section 3.6.3).

In the following subsections, we detail several methods by which we choose Y^{Impute}. These methods rely on various assumptions of different strength about the unobserved outcomes.

3.3.1 Mean Imputation

As a first approach, we assume that the mean outcome within $\mathcal{I}^+$ is representative of the mean outcome among the other pairs. This is a strong assumption, since the presence of a pair in $\mathcal{I}^-$ or $\mathcal{I}^{Att}$ may not be independent of their outcome. For example, if reviewers choose not to submit reviews when the assignment quality is poor, $\overline{Y}$ is not representative of the outcomes in $\mathcal{I}^{Att}$. Nonetheless, under this strong assumption, we can simply impute the mean outcome $\overline{Y}$ for all pairs necessitating imputation. Setting $Y_i^{Impute} = \overline{Y}$ for all $i \in \mathcal{I}^- \cup \mathcal{I}^{Att}$, we consider the following point estimate of μ_B: $\widehat{\mu}_B(\overline{Y})$. While following from an overly strong assumption, we find it useful to compare our findings under this assumption to findings under subsequent weaker assumptions.

3.3.2 Model Imputation

Instead of simply imputing the mean outcome, we can assume that the unobserved outcomes Y_i are some simple function of known covariates $X_i \in \mathbb{R}^c$ (where c is the number of covariates) for each reviewer-paper pair i. If so, we can directly estimate this function using a variety of statistical models, resulting in a point estimate of μ_B. In doing so, we implicitly take on the assumptions made by each model, which determine how to generalize the covariate-outcome

mapping from the observed pairs to the unobserved pairs. These assumptions are typically quite strong, since this mapping may be very different between the observed pairs (typically good matches) and unobserved pairs (typically less good matches).

Specifically, given the set of all observed reviewer-paper pairs $\mathcal{O} = \{i \in \mathcal{I}^+ : Z_i^A = 1\}$, we train a model m using the observed data $\{(X_i, Y_i) : i \in \mathcal{O}\}$. Let $\widehat{Y}^{(m)} \in \mathbb{R}^{m \times n}$ denote the outcomes predicted by that model for each pair. We then consider $\widehat{\mu}_B(\widehat{Y}^{(m)})$ as a point estimate of μ_B. In our experiments, we employ standard methods for classification, ordinal regression, and collaborative filtering:

- Logistic regression (*clf-logistic*);
- Ridge classification (*clf-ridge*);
- Ordered logit (*ord-logit*);
- Ordered probit (*ord-probit*);
- SVD++, collaborative filtering (*cf-svd++*);
- K-nearest-neighbors, collaborative filtering (*cf-knn*).

Note that, unlike the other methods, the methods based on collaborative filtering model the missing data by using only the observed reviewer-paper outcomes (Y). We discuss our choice of methods, hyperparameters, and implementation details in Section 3.6.5.

3.3.3 Manski Bounds

As a more conservative approach, we can exploit the fact that the outcomes Y_i are bounded, letting us bound the mean of the counterfactual policy without making any assumptions on how the positivity violations arise. Such bounds are often called *Manski bounds* [104] in the econometrics literature on partial identification. To employ Manski bounds, we assume that all outcomes Y can take only values between $y_{\min}$ and $y_{\max}$, e.g., self-reported expertise and confidence scores are limited to a pre-specified range on the review questionnaire. Then, setting $Y_i^{\textit{Impute}} = y_{\min}$ or $Y_i^{\textit{Impute}} = y_{\max}$ for all $i \in \mathcal{I}^- \cup \mathcal{I}^{\textit{Att}}$, we can estimate the upper and lower bound of μ_B as $\widehat{\mu}_B(y_{\min})$ and $\widehat{\mu}_B(y_{\max})$.

We adopt an inference procedure for constructing 95% confidence intervals that asymptotically contain the true value of μ_B with probability at least 95%. Following Imbens and Manski [70], we construct the interval:

$$\widehat{\mu}_B^{CI} \in \left[\widehat{\mu}_B(y_{\min}) - z'_{\alpha,n}\sqrt{\widehat{\text{Var}}[\widehat{\mu}_B(y_{\min})]/N},\ \widehat{\mu}_B(y_{\max}) + z'_{\alpha,n}\sqrt{\widehat{\text{Var}}[\widehat{\mu}_B(y_{\max})]/N}\right],$$

where the z-score analog $z'_{\alpha,n}$ ($\alpha = 0.95$), is set by their procedure such that the interval asymptotically has at least 95% coverage under plausible regularity conditions; for further details, see the discussion in Section 3.6.4.

3.3.4 Monotonicity and Lipschitz Smoothness

We now propose two styles of weak assumptions on the covariate-outcome mapping that can be leveraged to achieve tighter bounds on $\widehat{\mu}_B$ than the Manski bounds. In contrast to the strong modeling assumptions used in the sections on mean and model imputation, these assumptions

can be more intuitively understood and justified as conservative assumptions given particular choices of covariates.

Monotonicity. The first weak assumption we consider is a *monotonicity* condition. Intuitively, monotonicity captures the idea that we expect higher expertise for a reviewer-paper pair when some covariates are higher, all else equal. For example, in our experiments, we use the similarity component scores (bids, text similarity, subject area match) as covariates. Specifically, for covariate vectors X_i and X_j, define the *dominance* relationship $X_i \succ X_j$ to mean that X_i is greater than or equal to X_j in all components and X_i is strictly greater than X_j in at least one component. Then, the monotonicity assumption states that: if $X_i \succ X_j$, then $Y_i \geq Y_j, \forall (i,j) \in (\mathcal{R} \times \mathcal{P})^2$.

Using this assumption to restrict the range of possible values for the unobserved outcomes, we seek upper and lower bounds on μ_B. Recall that $\mathcal{O}$ is the set of all observed reviewer-paper pairs. One challenge is that the observed outcomes themselves (Y_i for $i \in \mathcal{O}$) may violate the monotonicity condition. Thus, to find an upper or lower bound, we compute *surrogate values* $\widetilde{Y}_i \in \mathbb{R}$ that satisfy the monotonicity constraint for all $i \in \mathcal{O} \cup \mathcal{I}^{Att} \cup \mathcal{I}^{Abs} \cup \mathcal{I}^{-}$ while ensuring that the surrogate values $\widetilde{Y}_i$ for $i \in \mathcal{O}$ are as close as possible to the outcomes Y_i. The surrogate values $\widetilde{Y}_i$ for $i \in \mathcal{I}^{Att} \cup \mathcal{I}^{-}$ can then be imputed as outcomes.

Inspired by isotonic regression [15], we implement a two-level optimization problem. The primary objective minimizes the ℓ_1 distance between $\widetilde{Y}_i$ and Y_i for pairs with observed outcomes $i \in \mathcal{O}$. The second objective either minimizes (for a lower bound) or maximizes (for an upper bound) the sum of Y_i for the unobserved pairs $i \in \mathcal{I}^{Att} \cup \mathcal{I}^{Abs} \cup \mathcal{I}^{-}$, weighted as in $\widehat{\mu}_B$. Define the universe of relevant pairs $\mathcal{U} = \mathcal{O} \cup \mathcal{I}^{Att} \cup \mathcal{I}^{Abs} \cup \mathcal{I}^{-}$ and define Ψ as a very large constant. This results in the following pair of optimization problems, which compute matrices $\widetilde{Y}^M_{min}, \widetilde{Y}^M_{max} \in \mathbb{R}^{m \times n}$ (leaving entries $i \notin \mathcal{U}$ undefined):

$$
\begin{aligned}
(\widetilde{Y}^M_{min}, \widetilde{Y}^M_{max}) = \underset{\widetilde{Y}_i : i \in \mathcal{U}}{\operatorname{argmin}} \quad & \Psi \sum_{i \in \mathcal{O}} |\widetilde{Y}_i - Y_i| \pm \left(\sum_{i \in \mathcal{I}^{Att} \cup \mathcal{I}^{Abs}} \widetilde{Y}_i W_i + \sum_{i \in \mathcal{I}^{-}} \widetilde{Y}_i \mathbb{P}_B(Z_i) \right), \\
\text{s.t.} \quad & \widetilde{Y}_i \geq \widetilde{Y}_j, \qquad \forall (i,j) \in \{\mathcal{U}^2 : X_i \succ X_j\}, \\
& y_{\min} \leq \widetilde{Y}_i \leq y_{\max}, \qquad \forall i \in \mathcal{U}.
\end{aligned}
$$

The sign of the second objective term depends on whether a lower (negative) or upper (positive) bound is being computed. The last set of constraints corresponds to the same constraints used to construct the Manski bounds described earlier, which is combined here with monotonicity to jointly constrain the possible outcomes. The above problem can be reformulated and solved as a linear program using standard techniques.

This procedure gives the following confidence intervals for μ_B,

$$
\widehat{\mu}^{CI}_{B|M} \in \left[\widehat{\mu}_B(\widetilde{Y}^M_{min}) - z'_{\alpha,n} \sqrt{\widehat{\text{Var}}[\widehat{\mu}_B(\widetilde{Y}^M_{min})]/N},\ \widehat{\mu}_B(\widetilde{Y}^M_{max}) + z'_{\alpha,n} \sqrt{\widehat{\text{Var}}[\widehat{\mu}_B(\widetilde{Y}^M_{max})]/N} \right],
$$

where the value $z'_{\alpha,n}$ is again set by the procedure of Imbens and Manski [70] (see discussion in Section 3.6.4).

Lipschitz Smoothness. The second weak assumption we consider is a *Lipschitz smoothness* assumption on the correspondence between covariates and outcomes. Intuitively, this captures

the idea that we expect two reviewer-paper pairs who are very similar in covariate space to have similar expertise. For covariate vectors X_i and X_j, define $d(X_i, X_j)$ as some notion of distance between the covariates. Then, the Lipschitz assumption states that there exists a constant L such that $|Y_i - Y_j| \leq Ld(X_i, X_j)$ for all $(i, j) \in (\mathcal{R} \times \mathcal{P})^2$. In practice, we can choose an appropriate value of L by studying the many pairs of observed outcomes in the data (Section 3.4.2 and Section 3.6.7), though this approach assumes that the Lipschitz smoothness of the covariate-outcome function is the same for observed and unobserved pairs.

As in the previous section, we introduce surrogate values $\widetilde{Y}_i \in \mathbb{R}$ and implement a two-level optimization problem to address Lipschitz violations within the observed outcomes (i.e., if two observed pairs are very close in covariate space but have different outcomes). Defining $\mathcal{U}$ and Ψ as above, this results in the following pair of optimization problems, which compute matrices $\widetilde{Y}^L_{min}, \widetilde{Y}^L_{max} \in \mathbb{R}^{m \times n}$ (leaving entries $i \notin \mathcal{U}$ undefined):

$$
\begin{aligned}
(\widetilde{Y}^L_{min}, \widetilde{Y}^L_{max}) = \underset{\widetilde{Y}_i : i \in \mathcal{U}}{\operatorname{argmin}} \; & \Psi \sum_{i \in \mathcal{O}} |\widetilde{Y}_i - Y_i| \pm \left(\sum_{i \in \mathcal{I}^{Att} \cup \mathcal{I}^{Abs}} \widetilde{Y}_i W_i + \sum_{i \in \mathcal{I}^-} \widetilde{Y}_i \mathbb{P}_B(Z_i) \right) \\
\text{s.t.} \quad & |\widetilde{Y}_i - \widetilde{Y}_j| \leq Ld(X_i, X_j), \quad \forall (i, j) \in \mathcal{U}, \\
& y_{min} \leq \widetilde{Y}_i \leq y_{max}, \qquad\qquad \forall i \in \mathcal{U}.
\end{aligned}
$$

As before, the sign of the second objective term depends on whether a lower (negative) or upper (positive) bound is being computed. The last set of constraints are again the same constraints used to construct the Manski bounds described earlier, which here are combined with the Lipschitz assumption to jointly constrain the possible outcomes. In the limit, as $L \to \infty$, the Lipschitz constraints become vacuous and we recover the Manski bounds. This problem can again be reformulated and solved as a linear program using standard techniques.

This procedure gives the following confidence intervals for μ_B,

$$
\widehat{\mu}^{CI}_{B|L} \in \left[\widehat{\mu}_B(\widetilde{Y}^L_{min}) - z'_{\alpha,n} \sqrt{\widehat{\text{Var}}[\widehat{\mu}_B(\widetilde{Y}^L_{min})]/N}, \; \widehat{\mu}_B(\widetilde{Y}^L_{max}) + z'_{\alpha,n} \sqrt{\widehat{\text{Var}}[\widehat{\mu}_B(\widetilde{Y}^L_{max})]/N} \right],
$$

where the value $z'_{\alpha,n}$ is again set by the procedure of Imbens and Manski [70] (see Section 3.6.4).

3.4 Experimental Setup

We apply our framework to data from two venues that used randomized paper assignments as described in Section 3.1, the 2021 Workshop on Theory and Practice of Differential Privacy (*TPDP*) and the 2022 AAAI Conference on Advancement in Artificial Intelligence (*AAAI*). In both settings, we aim to understand the effect that changing parameters of the assignment policies would have on review quality. The analyses were approved by our Institutional Review Boards.

3.4.1 Datasets

TPDP. The TPDP workshop received 95 submissions and had a pool of 35 reviewers. Each paper received exactly 3 reviews, and reviewers were assigned 8 or 9 reviews, for a total of 285 assigned reviewer-paper pairs. The reviewers were asked to bid on the papers and could place one of the following bids (the corresponding value of B_i is shown in the parenthesis):

"very low" (-1), *"low"* (-0.5), *"neutral"* (0), *"high"* (0.5), or *"very high"* (1), with *"neutral"* as the default. The similarity for each reviewer-paper pair was defined as a weighted sum of the bid score, B_i, and text-similarity scores, T_i: $S_i = w_{\text{text}} T_i + (1 - w_{\text{text}}) B_i$, with $w_{\text{text}} = 0.5$. The randomized assignment was run with an upper bound of $q = 0.5$. In their review, the reviewers were asked to assess the alignment between the paper and their *expertise* (between 1: irrelevant and 4: very relevant), and to report their review *confidence* (between 1: educated guess and 5: absolutely certain). We consider these two responses as our measures of quality. Once the assignment was generated, the organizers manually changed three reviewer-paper assignments, which we handle using the techniques discussed in Section 3.3.

AAAI. In the AAAI conference, submissions were assigned to reviewers in multiple sequential stages across two rounds of submissions. We examine the stage of the first round where the randomized assignment algorithm was used to assign all submissions to a pool of "senior reviewers." The assignment involved 8450 papers and 3145 reviewers; each paper was assigned to one reviewer, and each reviewer was assigned at most 3 or 4 papers based on their primary subject area. The similarity S_i for every reviewer-paper pair i was based on three scores: text-similarity $T_i \in [0, 1]$, subject-area score $K_i \in [0, 1]$, and bid B_i. Bids were chosen from the following list (with the corresponding value of B_i shown in the parenthesis, where $\lambda_{\text{bid}} = 1$ is a parameter scaling the impact of positive bids as compared to neutral/negative bids): *"not willing"* (0.05), *"not entered"* (1), *"in a pinch"* ($1 + 0.5\lambda_{\text{bid}}$), *"willing"* ($1 + 1.5\lambda_{\text{bid}}$), *"eager"* ($1 + 3\lambda_{\text{bid}}$). The default option was *"not entered"*. Similarities were computed as: $S_i = (w_{\text{text}} T_i + (1 - w_{\text{text}}) K_i)^{1/B_i}$, with $w_{\text{text}} = 0.75$. The actual similarities differed from this base similarity formula in a few special cases (e.g., missing data); we provide the full description of the similarity computation in Section 3.6.6. The randomized assignment was run with $q = 0.52$. Reviewers reported an *expertise* score (between 0: not knowledgeable and 5: expert) and a *confidence* score (between 0: not confident and 4: very confident), which we consider as our quality measures. After reviewers were assigned, several assignments were manually changed by the conference organizers, while several assigned reviews were also simply not submitted; we handle these cases as described in Section 3.3.

3.4.2 Assumption Suitability

For both the monotonicity and Lipschitz assumptions (as well as the model imputations), we work with the covariates X_i, a vector of the two (TPDP) or three (AAAI) component scores used in the computation of similarities. We now consider whether these assumptions are reasonable with respect to our choices of outcome variables and of covariates.

Monotonicity. Monotonicity assumes that when any component of the covariates increases, the review quality should not be lower. We can test this assumption on the observed outcomes: among all pairs of reviewer-paper pairs with both outcomes observed, 65.7% (TPDP) / 28.0% (AAAI) have a dominance relationship ($X_i \succ X_j$) and of those pairs, 79.8% (TPDP) / 76.4% (AAAI) satisfy the monotonicity condition when using expertise as an outcome and 76% (TPDP) / 78.9% (AAAI) when using confidence as an outcome. The fraction of dominant pairs for TPDP is higher since we consider only two covariates.

Lipschitz Smoothness. For the Lipschitz assumption, a choice of distance in covariate space is required. We choose the ℓ_1 distance, normalized in each dimension so that all component

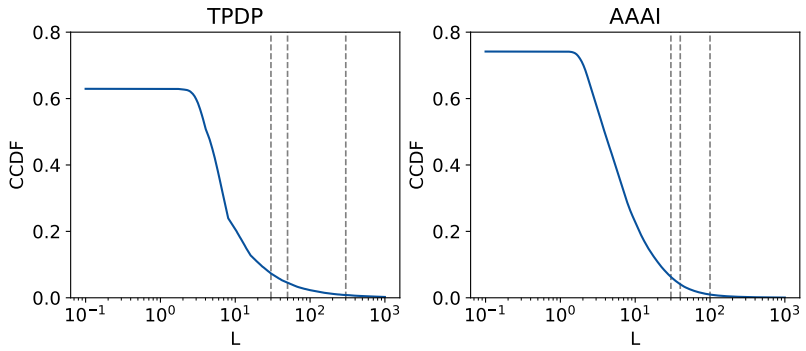


Figure 3.1: CCDF of the $L = |Y_i - Y_j|/d(X_i, X_j)$ values for all pairs of observed points, where Ys are *expertise* scores. The dashed lines denote the L values corresponding to less than 10%, 5%, and 1% violations. For TPDP, these values are $L = 30, 50, 300$, respectively; for AAAI, $L = 30, 40, 100$.

distances lie in $[0, 1]$, and divided by the number of dimensions. For AAAI, some reviewer-paper pairs are missing a covariate; if so, we impute a distance of 1 in that component. We then choose several potential Lipschitz constants L by analyzing the reviewer-paper pairs with observed outcomes. In Figure 3.1, we plot the fraction of pairs of observations that violate the Lipschitz condition for a given value of L with respect to expertise; we show the corresponding plots for confidence in Section 3.6.11. In our later experiments, we use values of L corresponding to less than 10%, 5%, and 1% violations from these plots.

With these choices, the Lipschitz assumptions correspond to beliefs that the outcome does not change too much as the similarity components change. As one example, for $L = 30$ on AAAI, when one similarity component differs by 0.1, the outcomes can differ by at most 1. Effectively, the imputed outcome of each unobserved pair is restricted to be relatively close to the outcome for the closest observed pair. In Section 3.6.7, we examine the distribution of distances between unobserved reviewer-paper pairs and their nearest observed pair, observing median distances of 0.0014 (TPDP) and 0.0011 (AAAI) across the pairs violating positivity under any of the modified similarity functions that we analyze in what follows. We conclude that most imputed pairs are very close to some observed pair, and even large values of L can significantly decrease the size of the bound when compared to the Manski bounds.

In solving the optimization problems for both the monotonicity and Lipschitz methods, we choose the constant $\Psi = 10^9$ to be large enough such that the first term of the objective dominates the second, while not causing numerical instability issues.

3.5 Results

We now present the analyses of the two datasets using the methods introduced in Section 3.3. For brevity, we report our analysis using self-reported expertise as the quality measure Y, but include the results using self-reported confidence in Section 3.6.11. When solving the LPs that output alternative randomized assignments (Section 3.6.1), we often encounter multiple unique optimal solutions and employ a persistent arbitrary tie-breaking procedure to choose amongst

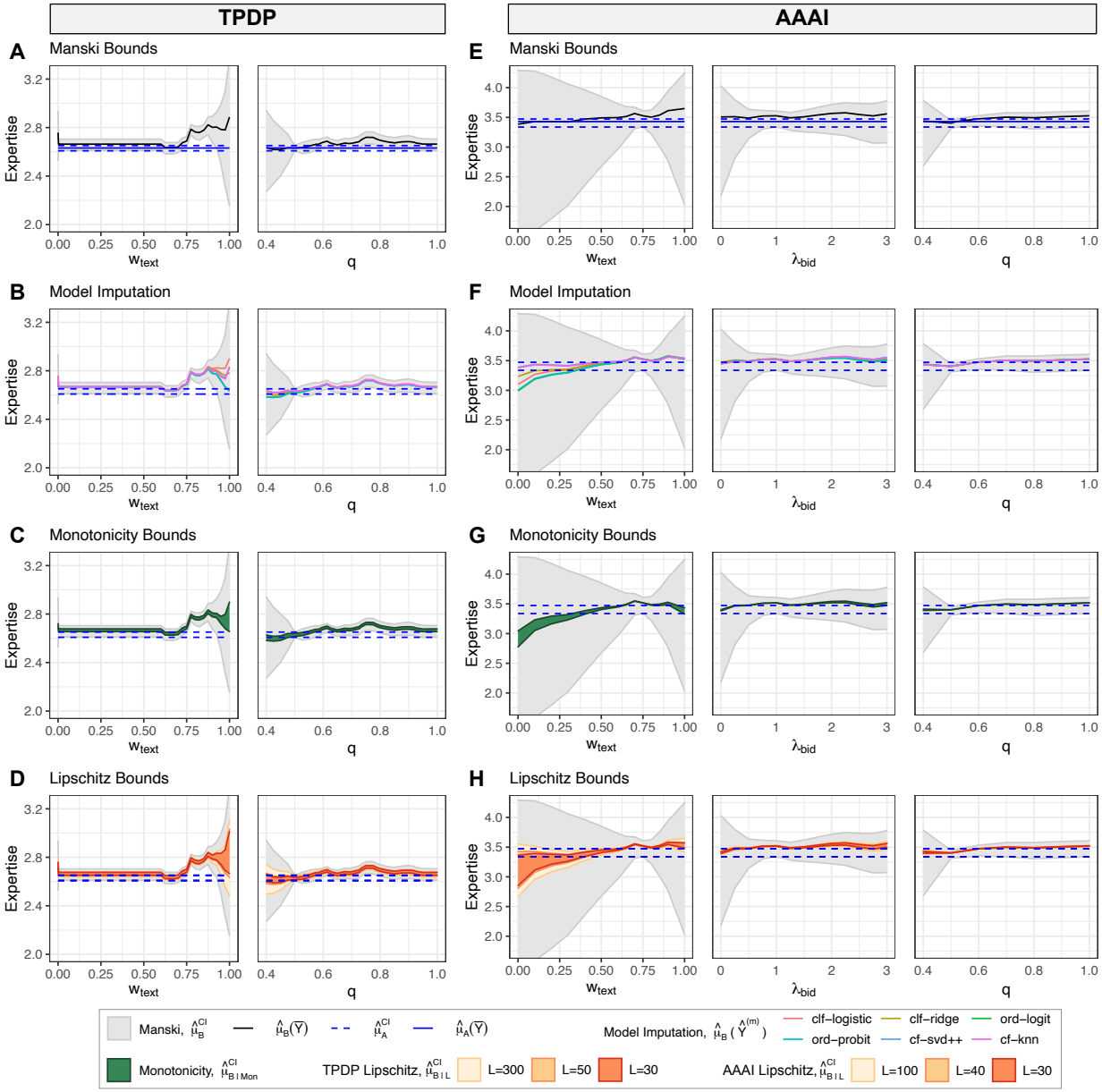


Figure 3.2: Expertise of off-policies varying w_{text} and q for TPDP and w_{text}, λ_{bid}, and q for AAAI, computed using the different estimation methods described in Section 3.3. The dashed blue lines indicate Manski bounds around the on-policy expertise, and the grey lines indicate Manski bounds around the off-policy expertise. The error bands (denoted $\widehat{\mu}_B^{CI}$ for the Manski bounds, $\widehat{\mu}_{B|M}^{CI}$ for the monotonicity bounds, and $\widehat{\mu}_{B|L}^{CI}$ for the Lipschitz bounds) represent confidence intervals that asymptotically contain the true value of μ_B with probability at least 95%, as described in Section 3.6.4. Note that to focus on the most relevant regions of the plots, the vertical axes do not start at zero. We generally see that increasing w_{text} results in higher expertise and that decreasing q leads to a very small reduction in review quality.

them (Section 3.6.8).

TPDP. We perform two analyses on the TPDP data, shown in Figure 3.2 (left). First, we analyze the choice of how to interpolate between the bids and the text similarity when computing the composite similarity score for each reviewer-paper pair. We examine a range of assignments, from an assignment based only on the bids ($w_{\text{text}} = 0$) to an assignment based only on the text similarity ($w_{\text{text}} = 1$), focusing our off-policy evaluation on deterministic assignments (i.e., policies with $q = 1$). Setting $w_{\text{text}} \in [0, 0.75]$ results in very similar assignments, each of which has Manski bounds overlapping with the on-policy (Figure 3.2A, left). Within this region, the models (Figure 3.2B, left), monotonicity bounds (Figure 3.2C, left), and Lipschitz bounds (Figure 3.2D, left) all agree that the expertise is similar to the on-policy. However, setting $w_{\text{text}} \in (0.75, 0.9)$ results in a significant improvement in average expertise, even without any additional assumptions (Figure 3.2A, left). Finally, setting $w_{\text{text}} \in (0.9, 1]$ leads to assignments that are significantly different from the assignments supported by the on-policy, which results in many positivity violations and wider confidence intervals, even under the monotonicity and Lipschitz smoothness assumptions (Figures 3.2A-D, left). Note that within this region, the models significantly disagree on the expertise (Figure 3.2B, left), indicating that the strong assumptions made by such models may not be accurate. Altogether, these results suggest that putting more weight on the text similarity (versus bids) leads to higher-expertise reviews.

Second, we investigate the "cost of randomization" to prevent fraud, measuring the effect of increasing q and thereby reducing randomness in the optimized random assignment. We consider values between $q = 0.4$ and $q = 1$ (optimal deterministic assignment). Recall the on-policy has $q = 0.5$. When varying q, we find that except for a small increase in the region around $q = 0.75$, the average expertise for policies with $q > 0.5$ is very similar to that of the on-policy (Figures 3.2A-D, right). These results suggest that using a randomized instead of a deterministic policy does not lead to a significant reduction in self-reported expertise, an observation that should be contrasted with the reduction in the expected sum-similarity objective under randomized assignments observed in Chapter 2; see further analysis in Section 3.6.9.

AAAI. We perform three analyses on the AAAI data, shown in Figure 3.2 (right). First, we examine the effect of interpolating between the text-similarity scores and the subject area scores by varying $w_{\text{text}} \in [0, 1]$, again considering only deterministic policies (i.e., $q = 1$). The on-policy sets $w_{\text{text}} = 0.75$. Due to large numbers of positivity violations, the Manski bounds are uninformative (Figure 3.2E, left), so we turn to the other estimators. The model imputation analysis indicates that policies with $w_{\text{text}} \geq 0.75$ may have slightly higher expertise than the on-policy and indicates lower expertise in the region where $w_{\text{text}} \leq 0.5$ (Figure 3.2F, left). However, the models differ somewhat in their predictions for low w_{text}, indicating that the assumptions made by these models may not be reliable. The monotonicity bounds more clearly indicate low expertise compared to the on-policy when $w_{\text{text}} \leq 0.25$, but are also slightly more pessimistic about the $w_{\text{text}} \geq 0.75$ region than the models (Figure 3.2G, left). The Lipschitz bounds indicate slightly higher than on-policy expertise for $w_{\text{text}} \geq 0.75$ and potentially suggest slightly lower than on-policy expertise for $w_{\text{text}} \leq 0.25$ (Figure 3.2H, left). Overall, all methods of analysis indicate that low values of w_{text} result in worse assignments, but the effect of considerably increasing w_{text} is unclear.

Second, we examine the effect of increasing the weight on positive bids by varying the val-

ues of λ_{bid}. Recall that $\lambda_{\text{bid}} = 1$ corresponds to the on-policy and a higher (respectively lower) value of λ_{bid} indicates greater (respectively lesser) priority given to positive bids relative to neutral/negative bids. We investigate policies that vary λ_{bid} within the range $[0, 3]$, and again consider only deterministic policies (i.e., $q = 1$). The Manski bounds are again too wide to be informative (Figure 3.2E, middle). The models all indicate similar values of expertise for all values of λ_{bid} and are all slightly more optimistic about expertise than the Manski bounds around the on-policy (Figure 3.2F, middle). The monotonicity and Lipschitz bounds both agree that the $\lambda_{\text{bid}} \geq 1$ region has slightly higher expertise as compared to the on-policy (Figure 3.2G-H, middle). Overall, our analyses provide some indication that increasing λ_{bid} may result in slightly higher levels of expertise.

Finally, we also examine the effect of varying q within the range $[0.4, 1]$. Recall that the on-policy sets $q = 0.52$. We see that the models (Figure 3.2F, right), the monotonicity bounds (Figure 3.2G, right), and the Lipschitz bounds (Figure 3.2H, right) all strongly agree that the region $q \geq 0.6$ has slightly higher expertise than the region $q \in [0.4, 0.6]$. However, the magnitude of this change is small, indicating that the "cost of randomization" is not very significant.

Power Investigation: Purposefully Bad Policies. As many of the off-policy assignments we consider have relatively similar estimated quality, we also ran additional analyses to show that our methods can discern differences between good policies (optimized toward high reviewer-paper similarity assignments) and policies intentionally chosen to have poor quality ("optimized" toward low reviewer-paper similarity assignments). We refer the reader to Section 3.6.10 for further discussion.

3.6 Supplemental Material

In this section, we present a variety of supplemental material, including omitted details and additional experimental results.

3.6.1 Linear Programs for Peer-Review Assignment

For concreteness, we state here the linear programs used to compute deterministic and randomized assignments as introduced in previous chapters.

Deterministic Assignment. Let $Z \in \{0, 1\}^{m \times n}$ be an assignment matrix where $Z_{r,p}$ denotes whether reviewer $r \in \mathcal{R}$ is assigned to paper $p \in \mathcal{P}$. Given a matrix of similarity scores $S \in \mathbb{R}_{\geq 0}^{m \times n}$, a standard objective is to find an assignment of papers to reviewers that maximizes the sum of similarities of the assigned pairs, subject to constraints that each paper is assigned to an appropriate number of reviewers ℓ_p, each reviewer is assigned no more than a maximum number of papers ℓ_r, and conflicts of interest are respected [29, 30, 52, 59, 102, 144, 145]. Denoting the set of conflict-of-interest pairs by $\mathcal{C} \subset \mathcal{R} \times \mathcal{P}$, this optimization problem can be formulated as

the following linear program:

$$\begin{aligned}
\max_{Z_{r,p}:r\in\mathcal{R},p\in\mathcal{P}} \quad & \sum_{r\in\mathcal{R},p\in\mathcal{P}} Z_{r,p}S_{r,p} & \\
\text{s.t.} \quad & \sum_{r\in\mathcal{R}} Z_{r,p} = \ell_p & \forall p\in\mathcal{P} \\
& \sum_{p\in\mathcal{P}} Z_{r,p} \le \ell_r & \forall r\in\mathcal{R} \\
& Z_{r,p} = 0 & \forall (r,p)\in\mathcal{C} \\
& 0 \le Z_{r,p} \le 1 & \forall r\in\mathcal{R}, p\in\mathcal{P}.
\end{aligned}$$

By total unimodularity conditions, this problem has an optimal solution where $Z_{r,p} \in \{0,1\}, \forall r \in \mathcal{R}, p \in \mathcal{P}$.

Although the above strategy is the primary method used for paper assignments in large-scale peer review, other variants of this method have been proposed and used in the literature. These algorithms consider various properties in addition to the total similarity, such as fairness [85, 138], strategyproofness [36, 160], envy-freeness [119] and diversity [96]. We focus on the sum-of-similarities objective here, but our off-policy evaluation framework is agnostic to the specific objective function.

Randomized Assignment. In Chapter 2, we introduce the idea of using randomization to prevent colluding reviewers and authors from being able to guarantee their assignments. Here, we restate the exact LP used to compute the randomized assignments that we analyze in this chapter. Specifically, the algorithm computes a randomized paper assignment, where the marginal probability $\mathbb{P}[Z_{r,p}]$ of assigning any reviewer r to any paper p is at most a parameter $q \in [0,1]$, chosen a priori by the program chairs. These marginal probabilities are determined by the following linear program, which maximizes the expected similarity of the assignment:

$$\begin{aligned}
\max_{\mathbb{P}[Z_{r,p}]:r\in\mathcal{R},p\in\mathcal{P}} \quad & \sum_{r\in\mathcal{R},p\in\mathcal{P}} \mathbb{P}[Z_{r,p}]S_{r,p} & \\
\text{s.t.} \quad & \sum_{r\in\mathcal{R}} \mathbb{P}[Z_{r,p}] = \ell_p & \forall p\in\mathcal{P} \\
& \sum_{p\in\mathcal{P}} \mathbb{P}[Z_{r,p}] \le \ell_r & \forall r\in\mathcal{R} \\
& \mathbb{P}[Z_{r,p}] = 0 & \forall (r,p)\in\mathcal{C} \\
& 0 \le \mathbb{P}[Z_{r,p}] \le q & \forall r\in\mathcal{R}, p\in\mathcal{P}.
\end{aligned} \tag{3.2}$$

A reviewer-paper assignment is then sampled using a randomized procedure that iteratively redistributes the probability mass placed on each reviewer-paper pair until all probabilities are either zero or one. This procedure ensures only that the desired marginal assignment probabilities are satisfied, providing no guarantees on the joint distributions of assigned pairs.

3.6.2 "No Interference" Assumption

Our estimators assume that there is no interference between the units, i.e., that the treatment of one unit does not affect the outcomes for the other units. In the causal inference literature, this

assumption is referred to as the Stable Unit Treatment Value Assumption (SUTVA) [33]. In the context of peer review, SUTVA implies that: (i) The quality $Y_{r,p}$ of the review by reviewer r reviewing paper p does not depend on what other reviewers are assigned to paper p; and (ii) the quality also does not depend on the other papers that reviewer r was assigned to review. The first assumption is quite realistic as in most peer review systems the reviewers cannot see other reviews until they submit their own. The second assumption is important to understand, as there could be "batch effects": a reviewer may feel more or less confident about their assessment (if measuring quality by confidence) depending on what other papers they were assigned to review. We do not test for batch effects or other violations of SUTVA in this work, which typically require either strong modeling assumptions or complex experimental designs [11, 122, 131, 152] specifically tailored for testing SUTVA, but consider it important future work.

3.6.3 Covariance Estimation

As described in Section 3.3, we estimate the variance of $\widehat{\mu}_B(Y^{\textit{Impute}})$ as:

$$\widehat{\text{Var}}[\widehat{\mu}_B(Y^{\textit{Impute}})] = \frac{1}{N^2} \sum_{(i,j)\in(\mathcal{R}\times\mathcal{P})^2} \text{Cov}[Z_i, Z_j] Z_i^A Z_j^A W_i W_j Y_i' Y_j',$$

$$\text{where } Y_i' = \begin{cases} Y_i & \text{if } i \in \mathcal{I}^+ \\ Y_i^{\textit{Impute}} & \text{if } i \in \mathcal{I}^{Att} \cup \mathcal{I}^- \\ \overline{Y} & \text{if } i \in \mathcal{I}^{Abs}. \end{cases}$$

However, the covariance terms (taken over $Z \sim \mathbb{P}_A$) are not known exactly. This is due to the fact that the procedure in Chapter 2 only constrains the marginal probabilities of individual reviewer-paper pairs, but pairs of pairs can be non-trivially correlated. In the absence of a closed-form expression, we use Monte Carlo methods to tightly estimate these covariances. In both our analyses of the TPDP and AAAI datasets, we sampled 1 million assignments and computed the empirical covariance. We ran an additional analysis to investigate the variability of our variance estimates. We took a bootstrap sample of 100,000 assignments (from the set of all 1 million assignments we sampled) and computed the variance based only on the (smaller) bootstrap sample. We repeated this procedure 1,000 times and computed the variance of our variance estimates. We found that the variance of our variance estimates is very small (less than 10^{-9}) even when we use 10 times fewer sampled assignments, suggesting that we have sampled enough assignments to accurately estimate the variance.

3.6.4 Coverage of Imbens-Manski Confidence Intervals

Under Manski, monotonicity, and Lipschitz assumptions, we employ a standard technique due to Imbens and Manski [70] for constructing confidence intervals for partially identified parameters. These intervals converge uniformly to the specified α-level coverage under a set of regularity assumptions on the behavior of the estimators of the upper and lower endpoints of the interval estimate: Assumption 1 from [70], establishing the coverage result in Lemma 4 there. It is difficult to verify whether Assumption 1 is satisfied for the designs (sampling reviewer-paper matchings) and interval endpoint estimators (Manski, monotonocity, Lipschitz) in this work.

A different set of assumptions, most significantly that the fraction of missing data is known

before assignment, support a different method for computing confidence intervals with the coverage result in Lemma 3 from [70], obviating the need for Assumption 1. In our setting, small amounts of attrition (relative to the number of policy-induced positivity violations) mean that the fraction of data that is missing is not exactly known before assignment, but almost. In practice, we find that the Imbens-Manski interval estimates from their Lemma 3 (assuming a known fraction of missing data) and Lemma 4 (assuming Assumption 1) are nearly identical for all three of the Manksi-, monotonicity-, and Lipschitz-based estimates, suggesting the coverage is well-behaved. A detailed theoretical analysis of whether the estimators obey the regularity conditions of Assumption 1 is beyond the scope of this work; see [82] for some theoretical developments related to the rates of convergence of Lipschitz-based estimates.

3.6.5 Model Implementation

To impute the outcomes of the unobserved reviewer-paper pairs, we train classification, ordinal regression, and collaborative filtering models. Classification models are suitable since the reviewers select their expertise and confidence scores from a set of pre-specified choices. Ordinal regression models additionally model the fact that the scores have a natural ordering. Collaborative filtering models, in contrast to the classification and ordinal regression models, do not rely on covariates and instead model the structure of the observed entries in the reviewer-paper outcome matrix, which is akin to user-item rating matrices found in recommender systems.

In the classification and regression models, we use the covariates X_i for each reviewer-paper pair as input features. In our analysis, we consider the two/three component scores used to compute the similarities: for TPDP, $X_i = (T_i, B_i)$; for AAAI, $X_i = (T_i, K_i, B_i)$. These are the primary components used by conference organizers to compute similarities, so we expect them to be usefully correlated with match quality. Although we perform our analysis with this choice of covariates, one could also include various other features of each reviewer-paper pair, e.g., some encoding of reviewer and paper subject areas, reviewer seniority, etc.

To evaluate the performance of the models, we randomly split the observed reviewer-paper pairs into train (75%) and test (25%) sets, fit the models on the train set, and measure the mean absolute error (MAE) of the predictions on the test set. To get more robust estimates of the performance, we repeat this process 10 times. In the training phase, we use 10-fold cross-validation to tune the hyperparameters, using MAE as a selection criterion, and retrain the model on the full training set with the best hyperparameters. We also consider two preprocessing decisions: (*a*) whether to encode the bids as one-hot categorical variables or continuous variables with the values described in Section 3.4.1, and (*b*) whether to standardize the features. In both cases, we used the settings that, overall, worked best (at prediction) for each model. We tested several models from each model category. To simplify the exposition, we only report the results of the two best-performing models in each category. The code repository referenced at the beginning of this chapter contains the implementation of all models, including the sets of hyperparameters considered for each model.

Figure 3.3 shows the test MAE across the 10 random train/test splits (means and 95% CIs) using expertise and confidence outcomes for both TPDP and AAAI. We note that all models perform significantly better than a baseline that predicts the mean outcome in the train set. For TPDP, we find that all models perform similarly, except for *cf-svd++*, which performs slightly better than the other models, both for expertise and confidence. For AAAI, all classification

Figure 3.3: Test performance of the imputation models described in Section 3.3.2, averaged across 10 random train/test splits of all observed reviewer-paper pairs. The error bars show 95% confidence intervals.

and regression models perform similarly, but the collaborative filtering models perform slightly worse. This difference in performance is perhaps due to the fact that we consider a larger set of covariates for AAAI than TPDP, likely making the classification and ordinal regression models more predictive.

Finally, to estimate $\widehat{\mu}_B$, we train each model on the set of all observed reviewer-paper pairs, predict the outcomes for all unobserved pairs, and impute the predicted outcomes as described in Section 3.3.2. In the training phase, we use 10-fold cross-validation to select the hyperparameters and refit the model on the full set of observed reviewer-paper pairs.

3.6.6 Details of AAAI Assignment

In Section 3.4.1, we described a simplified version of the stage of the AAAI assignment procedure that we analyze, i.e., the assignment of senior reviewers to the first round of submissions. In this section, we describe this stage of the AAAI paper assignment more precisely.

A randomized assignment was computed between 3145 senior reviewers and 8450 first-round paper submissions, independent of all other stages of the reviewer assignment. The set of senior reviewers was determined based on reviewing experience and publication record; these criteria were external to the assignment. Each paper was assigned $\ell_p = 1$ senior reviewer. Reviewers were assigned to at most $\ell_r = 4$ papers, with the exception of reviewers with a "Machine Learning" primary area or in the "AI For Social Impact" track, who were assigned to at most $\ell_r = 3$ papers. The probability limit was $q = 0.52$.

The similarities were computed from text-similarity scores T_i, subject-area scores K_i, and bids B_i. Either the text-similarity scores or the area scores could be missing for a given reviewer-paper pair, due to either a reviewer failing to provide the needed information or due to other errors in the computation of the scores. The text-similarity scores T_i were created using text-based scores from two different sources: (i) the Toronto Paper Matching System (TPMS) [29], and (ii) the ACL Reviewer Matching code [113]. The text-similarity scores T_i was set equal to the TPMS score for all pairs where this score was not missing, set equal to the ACL score for

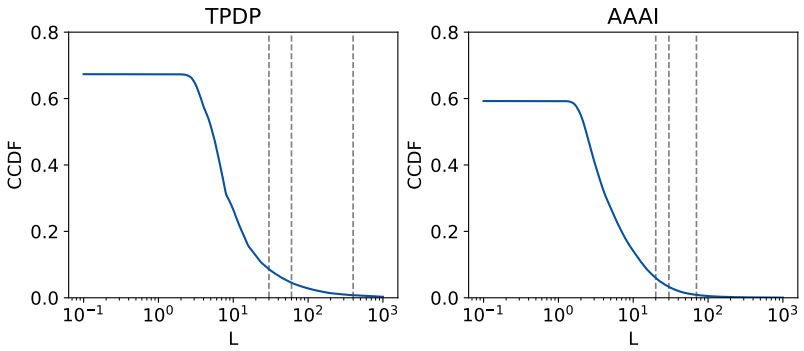


Figure 3.4: CCDF of the $L = |Y_i - Y_j|/d(X_i, X_j)$ values for all pairs of observed points, where Ys are *confidence* scores. The dashed lines denote the L values corresponding to less than 10%, 5%, and 1% violations. For TPDP, these values are $L = 30, 60, 400$, respectively; for AAAI, $L = 20, 30, 70$.

all other pairs where the ACL score was not missing, and marked as missing if both scores were missing. The subject-area scores were computed from reviewer and paper subject areas using the procedure described in Section A of [96].

Next, base scores $S'_i = w_{\text{text}} T_i + (1 - w_{\text{text}}) K_i$ were then computed with $w_{\text{text}} = 0.75$, if both T_i and K_i were not missing. If either T_i or K_i was missing, the base score was equal to the non-missing score of the two. If both were missing, the base score was set as $S'_i = 0$. For pairs where the bid was *"willing"* or *"eager"* and $K_i = 0$, the base score was set as $S'_i = T_i$.

Next, final scores were computed as $S_i = S_i'^{1/B_i}$, using the bid values *"not willing"* (0.05), *"not entered"* (1), *"in a pinch"* $(1 + 0.5\lambda_{\text{bid}})$, *"willing"* $(1 + 1.5\lambda_{\text{bid}})$, *"eager"* $(1 + 3\lambda_{\text{bid}})$; with $\lambda_{\text{bid}} = 1$. If $S_i < 0.15$ and K_i was not missing, the final score was recomputed as $S_i = \min(K_i^{1/B_i}, 0.15)$. Finally, for reviewers who did not provide their profile for use in conflict-of-interest detection, the final score was reduced by 10%.

In all of our analyses, we follow this same procedure to determine the assignment under alternative policies (varying only the parameters w_{text}, λ_{bid}, and q).

3.6.7 Details Regarding Assumption Suitability

In this section, we provide additional details on the discussion in Section 3.4.2 on the suitability of the monotonicity and Lipschitz smoothness assumptions.

First, we examine the fraction of pairs of observed reviewer-paper pairs that violate the Lipschitz condition for each value of L. Figure 3.4 shows the CCDF of L for pairs of observations (in other words, the fraction of violating observation-pairs for each value of L) with respect to confidence. The corresponding plot for expertise is shown in Figure 3.1.

Next, we examine the distances from unobserved reviewer-paper pairs to their closest observed reviewer-paper pair. In Figure 3.5, we show the CCDF of these distances for unobserved reviewer-paper pairs within a set of "relevant" pairs. We define the set of "relevant" unobserved pairs to be all pairs not supported on-policy that have positive probability in at least one policy among all off-policies with varying w_{text} with $q = 1$ for TPDP, and all off-policies varying w_{text} and λ_{bid} with $q = 1$ for AAAI.

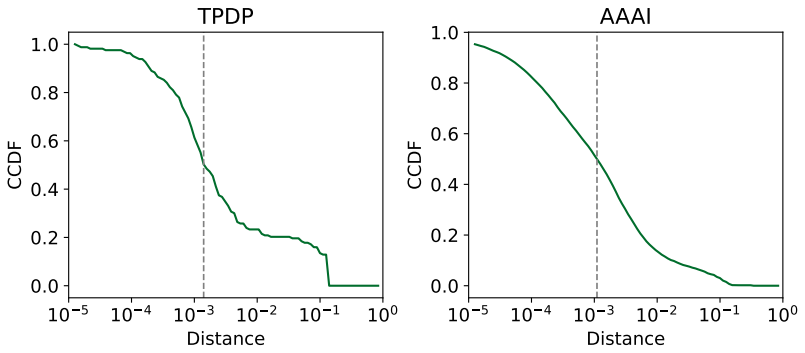


Figure 3.5: CCDF of the distances between each relevant unobserved reviewer-paper pair and its closest observed reviewer-paper pair. The dashed lines show the medians: 0.0014 (TPDP) and 0.0011 (AAAI).

3.6.8 Tie-Breaking Behavior

In Section 3.5, we specify a policy in terms of the parameters of LP (3.2) (specifically, by altering the parameters q, w_{text}, and λ_{bid} from the on-policy values). However, LP (3.2) may not have a unique solution, and thus each policy may not correspond to a unique set of assignment probabilities. Of particular concern, the on-policy specification of LP (3.2) does not uniquely identify the actual on-policy assignment probabilities.

Ideally, we could use the same tie-breaking methodology as was used in the on-policy to pick a solution in each off-policy to avoid introducing additional effects from variations in tie-breaking behavior. However, this behavior was not specified in the venues we analyze. To resolve this, we fixed arbitrary tie-breaking behaviors such that the on-policy solution to LP (3.2) matches the actual on-policy assignment probabilities; we then use these same behaviors for all off-policies.

In the TPDP analyses, we perturb all similarities by small constants such that all similarity values are unique. Specifically, we change the objective of LP (3.2) to $\sum_{i \in \mathcal{R} \times \mathcal{P}} \mathbb{P}[Z_i][(1-\lambda)S_i + \lambda \mathcal{E}_i]$, where $\lambda = 1e^{-8}$, and $\mathcal{E} \in \mathbb{R}^{m \times n}$ is the same for all policies. To choose $\mathcal{E}$, we sampled each entry uniformly at random from $[0, 1]$ and checked that the solution of the perturbed on-policy LP matches the on-policy assignment probabilities, resampling until it does. This value of $\mathcal{E}$ was then fixed for all policies.

In the AAAI analyses, the larger size of the similarity matrix meant that randomly choosing an $\mathcal{E}$ that recovers the on-policy solution was not feasible. Instead, we more directly choose how to perturb the similarities in order to achieve consistency with the on-policy. We change the objective of LP (3.2) to $\sum_{i \in \mathcal{R} \times \mathcal{P}} \mathbb{P}[Z_i](S_i - \epsilon \mathbb{I}[\mathbb{P}_A(Z_i) = 0])$, where $\epsilon \in \mathbb{R}$ is chosen for each policy by the following procedure. For each policy, ϵ is chosen to be the largest value from $\{10^{-9}, 10^{-6}, 10^{-3}\}$ such that the difference in total similarity between the solution of the original and perturbed LPs is no greater than a tolerance of 10^{-5}. We confirmed that using this procedure to perturb the on-policy LP recovers the on-policy assignment probabilities, as desired.

3.6.9 Similarity Cost of Randomization

In Chapter 2, we empirically analyze the "cost of randomization" in terms of the expected total assignment similarity, i.e., the objective value of LP (3.2), as q changes. This approach is also

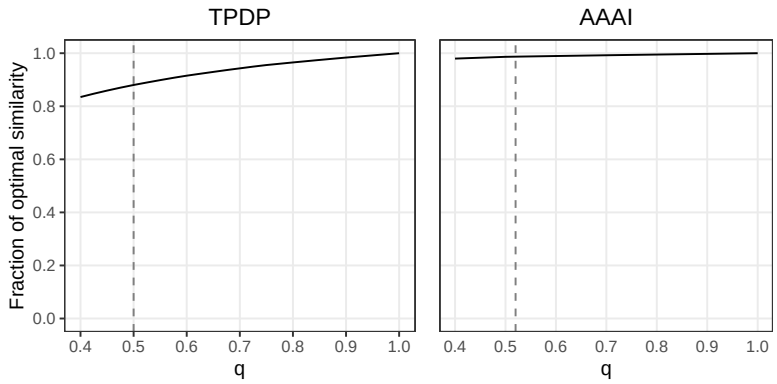


Figure 3.6: The "cost of randomization" as measured by the expected total assignment similarity. The plot shows the ratio between the sum of similarities under a randomized assignment (LP (3.2)) with ($q \leq 1$) and the sum of similarities under a deterministic assignment ($q = 1$). The dashed lines show the values of q set on-policy.

used by conference program chairs to choose an acceptable level of q in practice. In Figure 3.6, we show this trade-off between q and sum-similarity (as a ratio to the optimal deterministic sum-similarity) for both TPDP and AAAI. Note that in contrast, our approach in this work is to measure assignment quality via self-reported expertise or confidence rather than by similarity. In particular, the cost of randomization for TPDP is high in terms of sum-similarity but is revealed by our analysis to be mild in terms of expertise (Section 3.5).

3.6.10 Power Investigation: Purposefully Bad Policies

Many of the off-policy assignments we consider in Section 3.4 have shown to have relatively similar estimated quality. A possible explanation for this tendency is that most "reasonable" optimized policies are roughly equivalent in terms of quality, since our analyses only consider adjusting parameters of the (presumably reasonable) optimized on-policy. To investigate this possibility, we analyze a policy intentionally chosen to have poor quality.

Designing a "bad" policy that can be feasibly analyzed presents a challenge, as the on-policies are both optimized and thus rarely place probability on obviously bad reviewer-paper pairs. To work within this constraint, we look for bad policies where all reviewer pairs with zero on-policy probability are regarded as conflicts. We then contrast the deterministic ($q = 1$) policy that *maximizes* the total similarity score with the "bad" policy that *minimizes* it. Since the on-policy similarities are presumably somewhat indicative of expertise, we expect the minimization policy to be worse.

The results of this comparison are presented in Table 3.1. On both TPDP and AAAI, we see that our methods clearly identify the minimization policies as worse. The differences in quality between the policies becomes clearer with the addition of Lipschitz and monotonicity assumptions to address attrition. This illustrates that our methods are able to distinguish a good policy (the best of the best matches) from a clearly worse one (the worst of the best matches). Thus, it is likely that our primary analyses are simply exploring high-quality regions of the assignment-policy space, and that peer review assignment quality is often robust to the exact

Table 3.1: Expertise of bad policies (95% confidence intervals). $L = 50$ for TPDP and $L = 40$ for AAAI.

Policy	Manski	Monotonicity	Lipschitz
TPDP Max	[2.6115, 2.7045]	[2.6551, 2.6782]	[2.6498, 2.6744]
TPDP Min	[2.5521, 2.6126]	[2.5521, 2.5986]	[2.5521, 2.5937]
AAAI Max	[3.3919, 3.5213]	[3.4756, 3.4783]	[3.4764, 3.4809]
AAAI Min	[3.2591, 3.3846]	[3.3394, 3.3419]	[3.3396, 3.3443]

values of the various parameters.

3.6.11 Results for Confidence Outcomes

Figure 3.7 shows the results of our analyses using *confidence* as a quality measure (Y). We find that the results are substantively very similar to those reported in Section 3.5 using expertise.

3.7 Discussion

In this chapter, we evaluate the quality of off-policy reviewer-paper assignments in peer review using data from two venues that deployed randomized reviewer assignments. We propose new techniques for partial identification that allow us to draw useful conclusions about the off-policy review quality, even in the presence of large numbers of positivity violations and missing reviews. For a more extensive treatment of the methods proposed in this work, we refer the reader to Khan et al. [82].

Limitations. One limitation of off-policy evaluation is that our ability to make inferences inherently depends on the amount of randomness introduced on-policy. For instance, if there is a small amount of randomness, we will be able to estimate only policies that are relatively close to the on-policy unless we are willing to make some assumptions. The approaches presented in this work allow us to examine the strength of the evidence under a wide range of types and strengths of assumptions (model imputation, boundedness of the outcome, monotonicity, and Lipschitz smoothness) and to test whether these assumptions lead to converging conclusions. We emphasize that these assumptions (except for the boundedness of the outcome) are fundamentally unverifiable in our setting, given that the outcomes for unassigned pairs are unobserved. Thus, our resulting conclusions about the quality of alternative assignments cannot be directly verified without actually deploying the off-policies in question. Additionally, the measurement of review quality is a difficult question for any evaluation method: our analyses use the reviewers' self-reported expertise and confidence scores as proxies for review quality, but the self-reported nature of these measures makes them subject to various biases. Our conclusions should be interpreted in light of the possibility of misalignment between self-reported expertise/confidence and the true review quality. Finally, our substantive conclusions are based on analyses of data from two venues, and thus further work is needed to test their generalizability.

Future Work. In the context of peer review, the present work considers only a few parameterized slices of the vast space of reviewer-paper assignment policies, while there are many other

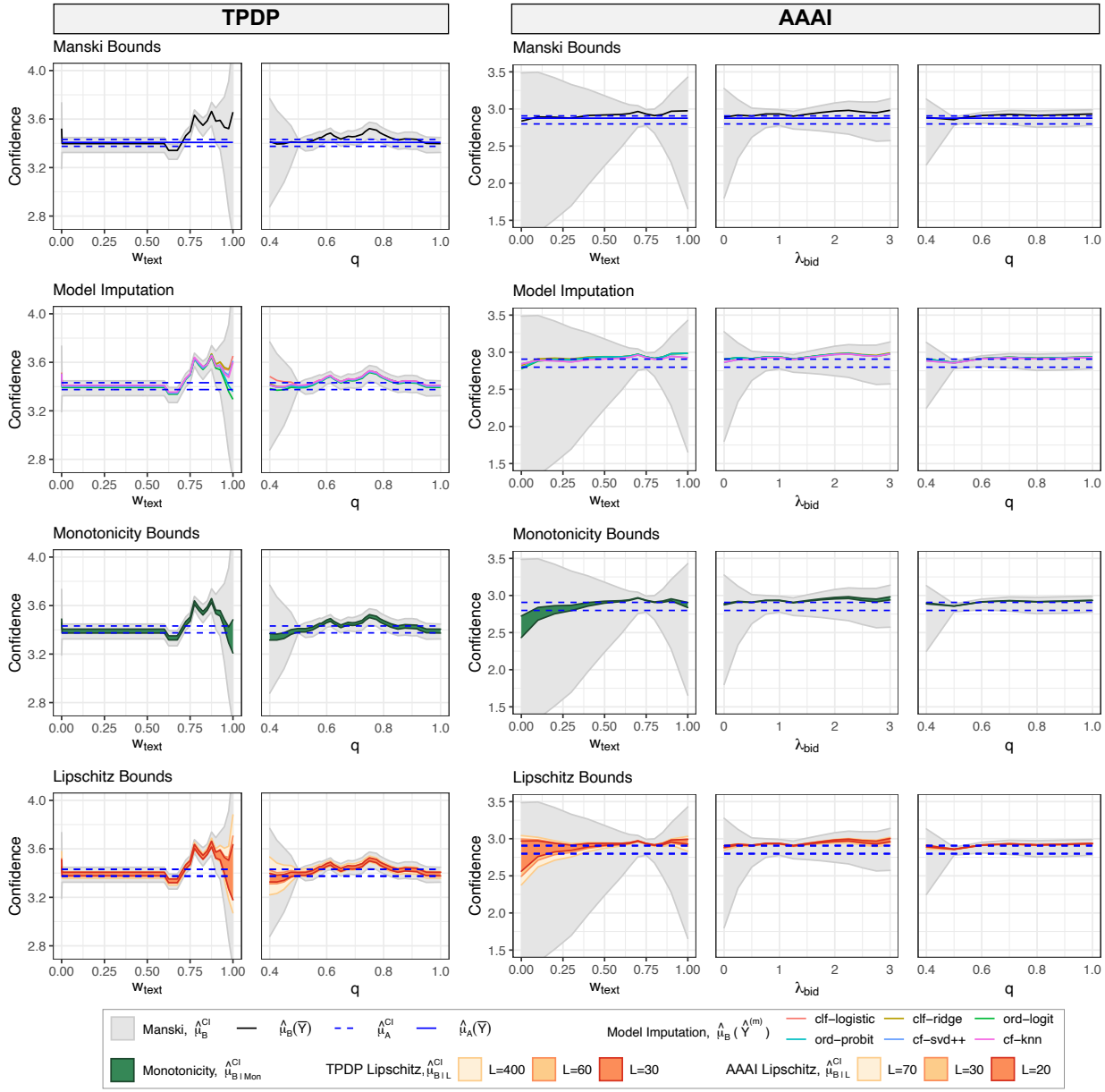


Figure 3.7: Confidence of off-policies varying w_{text} and q for TPDP and w_{text}, λ_{bid}, and q for AAAI, computed using the different estimation methods described in Section 3.3. The dashed blue lines indicate Manski bounds around the on-policy confidence, and the grey lines indicate Manski bounds around the off-policy confidence. The error bands (denoted $\widehat{\mu}_B^{CI}$ for the Manski bounds, $\widehat{\mu}_{B|M}^{CI}$ for the monotonicity bounds, and $\widehat{\mu}_{B|L}^{CI}$ for the Lipschitz bounds) represent confidence intervals that asymptotically contain the true value of μ_B with probability at least 95% as described in Section 3.3. Note that to focus on the most relevant regions of the plots, the vertical axes do not start at zero. We generally see that increasing w_{text} results in higher expertise and that decreasing q leads to a very small reduction in review quality.

substantive questions that our methodology can be used to answer. For instance, one could evaluate assignment quality under a different method of computing similarity scores (e.g., different NLP algorithms [31]), additional constraints on the assignment (e.g., based on seniority or geographic diversity [96]), or objective functions other than the sum-of-similarities (e.g., various fairness-based objectives [85, 98, 119, 138]). One interesting question in this vein is the effect of changing q on the review quality within specific sub-areas: does randomization have a disproportionate impact on the review quality within smaller subject areas? Our methodology can be used to directly answer this question. Another practical question concerns reviewer recruitment: how much would review quality have improved if additional reviewers had been recruited within certain areas? While we cannot directly estimate the quality if additional reviewers had been recruited, our methodology could be used to estimate as a proxy the quality if reviewers had been removed from the reviewer pool at random. Additional thought should also be given to the trade-offs between maximizing review quality and broader considerations of reviewer welfare: while assignments based on high text similarity may yield slightly higher-quality reviews, reviewers may be more willing to review again and less likely to submit low-effort reviews if the assignment policy follows their bids more closely. Beyond peer review, our work is applicable to off-policy evaluation in other matching problems, including school choice [7, 35], advertisement placement [21], and ridesharing [158]. The challenges we encounter in applying off-policy evaluation to randomized assignments in the peer review setting may also be significant challenges in these other matching settings; thus, the methodology of our work may be similarly helpful for drawing substantive conclusions about alternative matching policies.

Chapter 4

Robustness to Unreliable Reviewers via Two-Phase Paper Reviewing

In this chapter, we consider the problem of finding high-quality paper assignments during a two-phase paper assignment process. As we will detail further, a two-phase paper assignment process provides robustness to the issue of reviewer unreliability. However, the analysis in this chapter applies equally well to another important problem in conference peer review: the design of experiments on the review process. We first introduce both of these motivations.

Motivation 1: Two-phase paper assignment. Many computer-science conferences have adopted a two-phase review process, including the 2021 and 2022 AAAI Conferences on Artificial Intelligence (AAAI) and the 2022 International Joint Conference on Artificial Intelligence (IJCAI). After the initial reviews are submitted, a subset of papers proceed to a second phase of reviews with additional reviewers assigned. There are a variety of reasons that a two-phase reviewing process can be helpful. Crucially, a second phase of reviews can provide a conference with robustness to the various forms of reviewer unreliability that inevitably arise in large-scale conference peer review. For example, it can be used to help compensate for reviewers who were unresponsive or minimally responsive in the first phase, who can no longer review due to problems in their personal lives, who discovered conflicts they had with a paper assigned to them and recused themselves from it, etc. A second phase can also provide additional robustness against malicious behavior by allowing the assignment of additional second-phase reviewers to papers that received suspicious reviews in the first phase.

Another benefit of a two-phase reviewing process is that it can be used to triage papers based on reviews in the first phase. This can allow the conference to solicit additional reviews only on papers that obtained sufficiently high ratings in the first phase and have any chance of getting accepted (as done at AAAI 2021). The second phase can also help focus on evaluation of the papers in the "messy middle"—the papers at the borderline between acceptance and rejection. Additional reviewers could improve the evaluation of these papers to more accurately discern which should be accepted.

In all of these cases, the set of papers that will require additional review is unknown beforehand. While some venues choose to recruit new reviewers after knowing which papers proceed to phase two, the tight timeline of many conferences makes it hard to recruit new reviewers after phase one. In SIGMOD 2019 [1]: "*Additional reviews were solicited manually by the chairs*

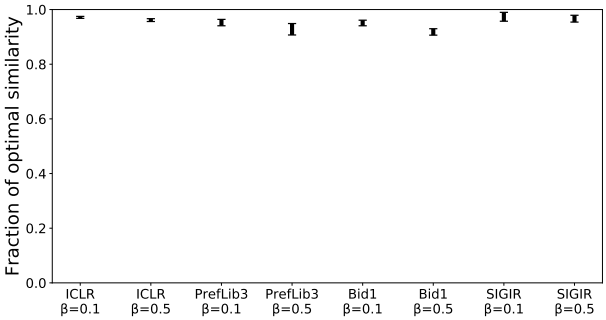
and this was a huge time sink, especially when some reviewers refused to take on the additional assignment. The additional review solicitation needs to be automated and reviewer expectations need to be set appropriately beforehand." For this reason, it is best if all the reviewers are recruited at the beginning, and a key question is then how to assign reviewers to papers in the first phase such that enough review capacity is saved for the second phase.

At a high level, the problem we study in the two-phase setting shares several common characteristics with problems in online (stochastic) matching [23, 37, 38, 47, 80], often considered in the context of ride-sharing, kidney exchange, or internet advertising. Particularly related to our results is the line of research on two-stage stochastic matching [44, 48, 81, 86, 95], which generally focuses on providing algorithms with tight approximation ratios that hold for any (i.e., worst-case) problem instances. To the best of our knowledge, the specific stochastic matching problem we consider (which arises in the context of paper assignment for peer review) has not previously been studied. Additionally, in contrast to this line of work, we aim to provide and justify simple and practical solutions based on data-dependent conditions likely to hold in real-world paper assignment instances.

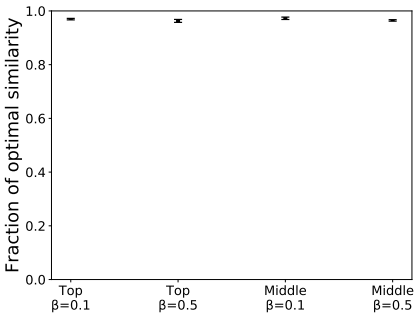
Motivation 2: Conference experiment design. Reviewers also need to be split into two groups when conferences run controlled experiments on the paper review process. Conferences often run such experiments to test changes to the review process. For example, the 2017 ACM International Conference on Web Search and Data Mining (WSDM) conducted an experiment to test the effects of single-blind versus double-blind reviewing [148]. As another example, the 2014 and 2021 Conferences on Neural Information Processing Systems (NeurIPS) ran experiments testing the consistency of acceptance decisions by providing some papers with a second set of reviews from a separate group of reviewers [17, 93, 124]. In these experiments, all papers receive reviews conducted in the usual manner (the control condition), but a random subset of papers are additionally assigned reviewers who provide reviews under an experimental condition. In the NeurIPS 2014/2021 experiments, a random 10% of papers were put in the experimental condition and received a second set of reviews. In the WSDM 2017 experiment, the subset of papers was the full paper set; that is, all papers were reviewed under both single-blind and double-blind conditions. The key question is then how to divide the reviewers between the control and experimental conditions. As in the NeurIPS 2014/2021 and WSDM 2017 experiments, this is often done randomly for statistical purposes. However, conferences still want to ensure that the resulting assignment of papers to reviewers is of high similarity.

Controlled experiments pertaining to peer review are conducted in many different scientific fields [8, 27, 118, 121, 146], including several controlled experiments recently conducted in computer science [93, 140, 142, 148]. These experiments have also led to a relatively nascent line of work on careful design of experimental methods for peer review [137, 139], and our work sheds some light in this direction in terms of trading off assignment quality with randomization in the assignment. Some other experiments in conferences [103, 105, 151] do not operate under controlled settings, but exploit certain changes in the conference policy such as a switch from single blind to double blind reviewing (i.e., natural experiments). While the methodology in Chapter 3 offers a less-costly alternative to experiments on the paper assignment policy, experiments offer important insights into many other aspects of the peer review process.

As our results apply to both the two-phase and experiment design settings, we will use the generic terminology of "stages" to refer to both phases and conditions simultaneously across the



(a) Second-stage papers drawn uniformly at random



(b) Second-stage papers chosen as the top- or middle-scoring papers from ICLR

Figure 4.1: Range of assignment similarities found over 10 random reviewer splits on real conference data, as a fraction of the oracle optimal assignment's similarity (computed after observing the second-stage papers). β indicates the fraction of papers in the second stage. The ICLR similarities [160] (911 papers, 2435 reviewers) are constructed from text-matching between papers and reviewers' past work, PrefLib3 [106] (176 papers, 146 reviewers) and Bid1 [108] (600 papers, 400 reviewers) similarities are constructed from bidding data, and SIGIR [78] similarities (73 papers, 189 reviewers) are constructed from reviewer and paper subject areas.

two settings.

Problem outline. In this chapter, we formally analyze the two-stage paper assignment problem, which encompasses both above motivations. As stated earlier, the standard paper assignment problem is to maximize the total similarity of the assignment subject to load constraints and is efficiently solvable. However, in the two-stage paper assignment problem, we must additionally decide how much of each reviewer's capacity should be saved to review papers in the second stage. We assume that the *fraction* of papers that will need additional reviews is known and that the set of second-stage papers is chosen uniformly at random.

Because of constraints present in each setting, the maximum-similarity paper assignment across the two stages cannot be achieved. In the two-phase setting, the set of papers that will need to be reviewed in the second stage is unobserved when the first-stage assignment is made, making the problem one of stochastic optimization. In the experiment design setting, reviewers are often randomized between stages for statistical purposes. We show that a simple strategy for choosing reviewers to save for the second stage performs near-optimally in terms of assignment similarity and can be used in either setting.

Contributions. Our contributions in this chapter are as follows.

First, we identify and formulate the two-stage paper assignment problem, an issue of practical importance to modern conferences, with applications to two-phase paper assignment and conference experiment design (Section 4.1).

Second, we prove that a simplified version of the problem is NP-hard, suggesting that the problem may not be efficiently solvable (Section 4.2).

Third, we empirically show that a very simple "random split" strategy, which chooses a subset

of reviewers uniformly at random to save for the second stage, gives near-optimal assignments on real conference similarity scores (Section 4.3.1). This result is summarized in Figure 4.1, which shows the assignment similarity achieved using random split as compared to the oracle optimal assignment (which views the set of second-stage papers before optimally assigning reviewers across both stages) for several datasets. We find that all random reviewer splits achieve at least 90% of the oracle optimal solution's similarity on all datasets and at least 94% on all but two experiments. These results hold across similarities constructed via a variety of methods used in practice (including text-matching, bidding, and subject areas), indicating that random split is robust across methods of similarity construction. They also hold both when the second-stage papers are drawn uniformly at random (as in Figure 4.1a) and when they are selected based on the review scores of the papers (as in Figure 4.1b). In practice, this means that program chairs planning a two-phase review process or a conference experiment can simply split reviewers across the two phases/conditions at random without concerning themselves with the potential reduction in assignment quality.

We also show that this good performance is not achieved in general: there exist similarity matrices on which random split performs very poorly (Section 4.3.2).

Fourth, we theoretically *explain* why random split performs well on our real conference similarity matrices by deriving theoretical bounds on the suboptimality of this random strategy under certain natural conditions (Sections 4.4 and 4.5). We consider two such sufficient conditions here, which are met by our datasets: if the reviewer-paper similarity matrix is low-rank, and if the similarity matrix allows for a high-value assignment (in terms of total similarity) with a large number of reviewers assigned to each paper. From these results, we give key actionable insights to conference program chairs to help them decide–well before the reviewers and/or papers are known–if random split is likely to perform well in their conference.

All of the code for our empirical results is freely available online at `https://github.com/sjecmen/multistage_reviewing_bounds`.

4.1 Problem Formulation

In this section, we formally define the two-stage paper assignment problem. Given a set of n papers $\mathcal{P} = [n]$ and a set of m reviewers $\mathcal{R} = [m]$, define $S \in [0,1]^{m \times n}$ as the similarity scores between each reviewer and paper. An assignment of papers to reviewers is represented as a matrix $Z \in \{0,1\}^{m \times n}$, where $Z_{r,p} = 1$ if reviewer r is assigned to paper p and $Z_{r,p} = 0$ otherwise. In the standard paper assignment problem, the objective is to find an assignment Z of reviewers to papers such that the total similarity $\sum_{r \in \mathcal{R}, p \in \mathcal{P}} Z_{r,p} S_{r,p}$ is maximized, subject to constraints that each paper is assigned exactly a certain load of reviewers, each reviewer is assigned to at most a certain load of papers, and any reviewer-paper pairs with a conflict of interest are not assigned [29, 30, 52, 59, 85]. In this chapter, we accommodate conflicts of interest by assuming the corresponding similarities are set to 0. This problem can be formulated as a min-cost flow problem or as a linear program, and can be efficiently solved.

In a two-stage assignment, all papers $\mathcal{P}$ require a certain number of reviewers in the first stage and a subset of papers $\mathcal{P}_2 \subseteq \mathcal{P}$ require additional review in the second stage. We assume that $\mathcal{P}_2$ consists of a fixed fraction β of papers and is drawn uniformly at random from $\mathcal{P}$. Specifically, for some $\beta \in \{\frac{1}{n}, \dots, \frac{n}{n}\}$, we assume that $\mathcal{P}_2 \sim \mathcal{U}_{\beta n}(\mathcal{P})$, the uniform distribution over all subsets

of size βn of $\mathcal{P}$. In the two-phase setting, the fraction β itself can be viewed as a parameter that the program chairs set based on available reviewer resources, or it can be estimated from past editions of the conference. Our empirical results detailed in Section 4.3.1 also cover the case where papers are chosen for the second phase based on their first-phase review scores. In the conference experiment design setting, the value of β and the uniform distribution of $\mathcal{P}_2$ are both experiment design choices. The choice of a uniform distribution for $\mathcal{P}_2$ is common, as in the NeurIPS 2014/2021 and WSDM 2017 experiments. The question we analyze is: how should reviewers be assigned to papers across the two stages?

Before continuing further, we introduce some notation. For subsets $\mathcal{R}' \subseteq \mathcal{R}$ and $\mathcal{P}' \subseteq \mathcal{P}$, desired paper load $\ell_p \in \mathbb{Z}_+$, and maximum reviewer load $\ell_r \in \mathbb{Z}_+$, define $\mathcal{M}(\mathcal{R}', \mathcal{P}'; \ell_r, \ell_p) \subseteq \{0,1\}^{m \times n}$ as the set of valid assignment matrices on $\mathcal{R}'$ and $\mathcal{P}'$. Formally, $Z \in \mathcal{M}(\mathcal{R}', \mathcal{P}'; \ell_r, \ell_p)$ if and only if $\sum_{r \in \mathcal{R}'} Z_{r,p} = \ell_p$ for all $p \in \mathcal{P}'$, $\sum_{p \in \mathcal{P}'} Z_{r,p} \leq \ell_r$ for all $r \in \mathcal{R}'$, and $Z_{r,p} = 0$ for all $(r,p) \notin \mathcal{R}' \times \mathcal{P}'$.

The two-stage paper assignment problem is to maximize the total similarity of the paper assignment across both stages. Without loss of generality, we instead consider the mean similarity so that later results will be easier to interpret. Fix a stage one paper load $\ell_p^{(1)}$, a stage two paper load $\ell_p^{(2)}$, and an overall reviewer load ℓ_r such that $\ell_p^{(1)} n + \ell_p^{(2)} \beta n \leq \ell_r m$ (i.e., the number of reviews required by papers is no greater than the number of reviews that can be supplied by reviewers). Given $\mathcal{P}_2$, the oracle optimal assignment has mean similarity

$$Q^*(\mathcal{P}_2) = \max_{\substack{A \in \mathcal{M}(\mathcal{R}, \mathcal{P}; \ell_r, \ell_p^{(1)}), \\ B \in \mathcal{M}(\mathcal{R}, \mathcal{P}_2; \ell_r, \ell_p^{(2)})}} \frac{1}{\ell_p^{(1)} n + \ell_p^{(2)} \beta n} \left[\sum_{r \in \mathcal{R}, p \in \mathcal{P}} A_{r,p} S_{r,p} + \sum_{r \in \mathcal{R}, p \in \mathcal{P}_2} B_{r,p} S_{r,p} \right]$$
$$\text{subject to } \sum_{p \in \mathcal{P}} A_{r,p} + B_{r,p} \leq \ell_r \qquad \forall r \in \mathcal{R}.$$

The last constraint ensures that each reviewer's assignment across both stages does not exceed the maximum reviewer load. Just like the standard paper assignment problem, the oracle optimal assignment for a given $\mathcal{P}_2$ can be found efficiently. However, in both the two-phase and experiment design settings, this oracle optimal assignment is either unachievable or undesirable. In the two-phase setting, the set of papers $\mathcal{P}_2$ requiring additional review is unknown until after the stage one assignment is chosen. Thus, the oracle optimal assignment cannot be computed beforehand. In the experiment design setting, the assignment of reviewers to conditions is commonly randomized in order to gain statistical power, as was done in the WSDM 2017 and NeurIPS 2014/2021 experiments. Thus, a deterministic choice of assignment may not be desirable. Additionally, depending on the experiment setup, it may not be possible for a reviewer to review papers in both conditions. In what follows, we use this oracle optimal assignment value as an unachievable baseline for comparison.

We instead consider simple strategies for the two-stage assignment problem that choose a subset $\mathcal{R}_2 \subseteq \mathcal{R}$ of reviewers to save for the second stage without observing $\mathcal{P}_2$, leaving reviewers $\mathcal{R}_1 = \mathcal{R} \setminus \mathcal{R}_2$ to be assigned to papers in the first stage. Unlike the oracle optimal assignment, such strategies are feasible to implement in both settings since they do not require knowledge of $\mathcal{P}_2$, do not split reviewer loads across conditions, and allow for a random choice of $\mathcal{R}_2$. The mean similarity of the optimal assignment when reviewers $\mathcal{R}_2$ and papers $\mathcal{P}_2$ are in the second

stage is

$$Q(\mathcal{R}_2, \mathcal{P}_2) = \frac{1}{\ell_p^{(1)} n + \ell_p^{(2)} \beta n} \left[\max_{\substack{A \in \mathcal{M}(\mathcal{R} \setminus \mathcal{R}_2, \mathcal{P}; \\ \ell_r, \ell_p^{(1)})}} \sum_{r \in \mathcal{R} \setminus \mathcal{R}_2, p \in \mathcal{P}} A_{r,p} S_{r,p} + \max_{\substack{B \in \mathcal{M}(\mathcal{R}_2, \mathcal{P}_2; \\ \ell_r, \ell_p^{(2)})}} \sum_{r \in \mathcal{R}_2, p \in \mathcal{P}_2} B_{r,p} S_{r,p} \right].$$

We require that $\ell_r |\mathcal{R}_2| \geq \ell_p^{(2)} \beta n$ and $\ell_r (m - |\mathcal{R}_2|) \geq \ell_p^{(1)} n$ for feasibility in both stages. Given $\mathcal{R}_1$, $\mathcal{R}_2$, and $\mathcal{P}_2$, the optimal paper assignment in each stage can be efficiently computed using standard methods. Thus, the the difficulty of the problem lies entirely in choosing $\mathcal{R}_2$.

The expected mean similarity of the optimal assignment when saving reviewers $\mathcal{R}_2$ for the second stage is

$$f(\mathcal{R}_2) = \mathbb{E}_{\mathcal{P}_2 \sim \mathcal{U}_{\beta n}(\mathcal{P})} \left[Q(\mathcal{R}_2, \mathcal{P}_2) \right].$$

We can also evaluate the *suboptimality* of $\mathcal{R}_2$ as compared to the oracle optimal assignment as

$$Q^*(\mathcal{P}_2) - Q(\mathcal{R}_2, \mathcal{P}_2), \qquad \text{where } \mathcal{P}_2 \sim \mathcal{U}_{\beta n}(\mathcal{P}).$$

Note that Q^* and Q are bounded in $[0, 1]$, so that both f and the suboptimality are also bounded in $[0, 1]$.

In our theoretical analysis, for simplicity, we assume that $\ell_r = \ell_p^{(1)} = \ell_p^{(2)} = 1$, leaving this implicit in f, Q, and Q^* throughout the paper. We also assume $m = (1 + \beta)n$ in our analysis unless specified otherwise, so that $|\mathcal{R}_2| = \beta n$. The intuition behind our results carries over to the cases of general loads and excess reviewers, which are covered by our empirical results in Section 4.3.1. Although we do not extend our theoretical results to formally handle these cases, we do not believe that doing so would provide any additional practical insights for program chairs. All asymptotic bounds are given as n grows.

4.2 Hardness

In the two-phase setting, the oracle optimal assignment is unachievable because $\mathcal{R}_2$ must be chosen before observing $\mathcal{P}_2$. Therefore, conferences must choose $\mathcal{R}_2$ to maximize f, the expected mean similarity of the assignment across both stages. In this section, we demonstrate that maximizing a variant of f is NP-hard, indicating that it is unlikely that f can be optimized efficiently.

First, note that evaluating $f(\mathcal{R}_2)$ requires computing an expectation over the draw of $\mathcal{P}_2$, which naively requires evaluating a sum over the optimal assignment value for $\binom{n}{\beta n}$ possible choices of $\mathcal{P}_2$. This number is exponential in the input size, so an efficient algorithm for this problem would have to either optimize f without evaluating it or compute this expectation without computing the optimal assignment for each possible $\mathcal{P}_2$.

Instead of attempting to optimize f exactly, a standard approach from two-stage stochastic optimization is to simplify the problem by sampling as follows [34, 84]. First, take some fixed number of samples $\mathcal{P}_2^{(1)}, \ldots, \mathcal{P}_2^{(K)}$ from $\mathcal{U}_{\beta n}(\mathcal{P})$. Then, rather than optimizing an average over all $\mathcal{P}_2$ in the support of $\mathcal{U}_{\beta n}(\mathcal{P})$, choose $\mathcal{R}_2$ to optimize an average over only all sampled sets:

$$\overline{f}(\mathcal{R}_2) = \frac{1}{K} \sum_{i=1}^{K} Q(\mathcal{R}_2, \mathcal{P}_2^{(i)}).$$

This is a natural simplification of the two-stage paper assignment problem, because the sum in the objective is now taken over only a constant K subsets rather than an exponential number. However, this problem is still not efficiently solvable, as the following theorem shows.

Theorem 4.1. *It is NP-hard to find $\mathcal{R}_2 \subseteq \mathcal{R}$ such that $\overline{f}(\mathcal{R}_2)$ is maximized, even when $K = 3$.*

Proof sketch. We reduce from 3-Dimensional Matching [79], which asks if there exists a way to select k tuples from a set $T \subseteq A \times B \times C$ where $|A| = |B| = |C| = k$ such that all elements of A, B, and C are selected exactly once. We construct 3 samples of second-stage papers corresponding to A, B, and C respectively, and construct reviewers corresponding to elements of T. These reviewers have 1 similarity with the papers in their tuple, and 0 similarity with all other papers. Thus, checking if there exists a choice of $\mathcal{R}_2$ which gives full expected similarity in the second stage would require solving 3-Dimensional Matching. We add additional reviewers and papers to ensure that this choice of $\mathcal{R}_2$ is optimal over both stages. □

The full proof is presented in Section 4.7.

Since it is NP-hard to find the optimal $\mathcal{R}_2$ even when estimating the objective by sampling three random choices of $\mathcal{P}_2$, this suggests that the original objective f may be hard to optimize efficiently. Therefore, in the two-phase assignment setting, we instead look for efficient approximation algorithms.

4.3 Our Approach: Random Split

Our proposed approach for finding a two-stage assignment is extremely simple: choose $\mathcal{R}_2$ uniformly at random (i.e., $\mathcal{R}_2 \sim \mathcal{U}_{\beta n}(\mathcal{R})$). We refer to this as a "random split" of reviewers into the two review stages.

In the two-phase setting, random split is an efficient approximation algorithm for the problem of optimizing f, which is likely difficult (as shown in Section 4.2). Because random split does not execute f, it produces a two-stage paper assignment without needing to estimate f by sampling.

In the conference experiment design setting, our proposed random-split strategy corresponds to a uniform random choice of reviewers for the experimental condition. Recall that in this setting, assigning reviewers to conditions uniformly at random is already a common experimental setup. The performance of random split on f therefore indicates how well this common setup performs in terms of the expected assignment similarity.

In our theoretical results, we often refer to the *suboptimality of random split*, defined as the suboptimality of $\mathcal{R}_2$ chosen via random split when $\mathcal{P}_2$ is chosen uniformly at random:

$$Q^*(\mathcal{P}_2) - Q(\mathcal{R}_2, \mathcal{P}_2), \qquad \text{where } \mathcal{P}_2 \sim \mathcal{U}_{\beta n}(\mathcal{P}), \mathcal{R}_2 \sim \mathcal{U}_{\beta n}(\mathcal{R}). \tag{4.1}$$

Recall from Section 4.1 that $Q^*(\mathcal{P}_2)$ is the mean similarity of the oracle optimal assignment given second-stage papers $\mathcal{P}_2$ and that $Q(\mathcal{R}_2, \mathcal{P}_2)$ is the mean similarity of the optimal assignment given second-stage reviewers and papers $\mathcal{R}_2, \mathcal{P}_2$. Additionally, many of our results evaluate the expected mean similarity under random split:

$$\mathbb{E}_{\mathcal{R}_2 \sim \mathcal{U}_{\beta n}(\mathcal{R})}\left[f(\mathcal{R}_2)\right] = \mathbb{E}_{\mathcal{R}_2 \sim \mathcal{U}_{\beta n}(\mathcal{R}), \mathcal{P}_2 \sim \mathcal{U}_{\beta n}(\mathcal{P})}\left[Q(\mathcal{R}_2, \mathcal{P}_2)\right].$$

In the following subsections, we first elaborate on the good performance random split displays empirically before showing that there exist cases where random split performs very poorly.

4.3.1 Empirical Performance

As introduced earlier in Figure 4.1, random split performs very well in practice on four real conference similarity matrices. The first is a similarity matrix recreated using text-matching on data from the 2018 International Conference on Learning Representations (ICLR) [160]. The second is constructed using reviewer bid data for an AI conference (conference 3) from PrefLib dataset MD-00002 [106]. The third (denoted Bid1) is a sample of the bidding data from a major computer science conference [108]. In both of these bidding datasets, we transformed "yes," "maybe," and "no response" bids into similarities of 1, 0.5, and 0.25 respectively, as is often done in practice [136]. The fourth similarity matrix is constructed from the subject areas of ACM SIGIR 2007 papers and the subject areas of the past work of their authors (assumed to be the reviewers) [78]; we set the similarity between each reviewer and paper to be equal to the number of matching subject areas out of the 25 total, normalized so that each entry of the matrix is in $[0, 1]$. In Section 4.6.1, we present further empirical results including additional datasets. In Section 4.6.2, we present additional empirical results particularly relevant to the conference experiment design setting.

We run several experiments, each corresponding to a choice of dataset and β. Each experiment consists of 10 trials, where in each trial we sample a random reviewer split and a set of second-stage papers. We then present the range of assignment values achieved across the trials as percentages of the oracle optimal assignments for each trial. The oracle optimal assignment for a trial is found by choosing the optimal assignment of reviewers across both stages after observing $\mathcal{P}_2$. We set paper loads of 2 in each stage (as done in AAAI 2021), and limit reviewer loads to be at most 6 (a realistic reviewer load [136]). Since these datasets have excess reviewers, we choose $\mathcal{R}_2$ to have size $\frac{\beta}{1+\beta}m$ so that the proportions of reviewers and papers in the second stage are equal.

In Figure 4.1a, $\mathcal{P}_2$ is drawn uniformly at random in each trial (as in the problem formulation). We see that all trials of random split achieve at least 90% of the oracle optimal solution's similarity on all datasets, with all trials on all but two experiments achieving at least 94%. We see additionally that the randomness of the reviewer choice does not cause much variance in the value of the assignment, as there is at most a 5% difference between the minimum and maximum similarity (as a percentage of oracle optimal) for each experiment. Note that this is true despite the fact that the similarity matrices of the different datasets are constructed in several different ways, indicating that random split is robust across methods of similarity construction.

In Figure 4.1b, $\mathcal{P}_2$ is chosen as a fixed set for all trials based on the actual review scores received by the papers at ICLR 2018 [63] (as review scores were not available for other datasets). We run trials where either the top-scoring papers or the messy-middle papers are given additional reviews. Since about 37% of papers were accepted, we define the messy middle as the range of $\frac{\beta}{1+\beta}m$ papers centered on the 63rd-percentile paper when ordered by score. These are sets of papers that a conference may potentially want to assign additional reviewers to. In all cases, random split shows consistently good performance, similar to when $\mathcal{P}_2$ was drawn uniformly at random. All trials achieve at least 95% of the oracle optimal similarity, with at most a 2% difference between the minimum and maximum for each experiment. This suggests that the good performance of random reviewer split naturally holds in these practical cases.

4.3.2 A Counterexample

The good results random split shows in practice may be somewhat surprising because random split does not perform well in all settings. The following theorem shows that for any β, there exist instances of the two-stage paper assignment problem where the suboptimality of random split (4.1) is $\Omega(1)$ in expectation.

Theorem 4.2. *For any constant $\beta \in [0, 1]$, there exists n_0 such that for all $n \geq n_0$ where $\beta n \in \mathbb{Z}_+$, there exist instances of the two-stage paper assignment problem where the suboptimality of random split is at least $\frac{\beta^4}{(1+\beta)^3}$ in expectation.*

Proof sketch. Consider $\beta = 1$. We construct a similarity matrix where every reviewer has similarity 1 with only 1 paper, and all papers have similarity 1 with only 2 reviewers. The optimal reviewer split puts the two good reviewers for each paper in separate stages and always achieves a mean similarity of 1. Random split puts both good reviewers in the same stage with at least constant probability for each paper, giving a constant mean similarity < 1. □

The full proof is presented in Section 4.7.

Note that the above lower bound on the objective value of random split holds even in the easy case of $\beta = 1$, where the problem could be solved simply through standard paper assignment methods. This case is particularly relevant in the conference experiment setting, where all papers are commonly reviewed under both conditions (as in the WSDM 2017 experiment).

Although the above lower bound demonstrates that random split cannot hope to do well in general, the constructed example is unrealistic for real conferences. However, program chairs may understandably want some guarantee that a random reviewer split will work well for their conference before deciding to use it. Ideally, this guarantee should be given before the precise similarity matrix for the conference is known, since the similarities may not be known in full until late in the planning process.

In the following sections, we provide such guarantees, thereby showing that the good performance of random split is not just an artifact of our specific datasets. We focus our attention on two sufficient conditions on the similarity matrix under which we show random split performs well. These conditions are natural for real similarity matrices, implying that random split will perform well for many real conferences, whether in the context of a two-phase review process or a conference experiment. Using these conditions, we provide actionable insights to program chairs based on simple properties of their conference's similarities that they may have intuition about. These insights are designed to be useful well before the full paper and reviewer sets are known.

4.4 Condition 1: Low-Rank Similarity Matrix

The first condition we consider is that the similarity matrix S has low rank k. This condition naturally arises in practice when reviewer-paper similarities are calculated from the number of subject area agreements between reviewers and papers; in such cases, the rank is no greater than the number of subject areas. For example, the SIGIR similarity matrix used in Figure 4.1 is constructed in this way and thus has rank no greater than 25 (the number of subject areas). In this section, we provide asymptotic upper and lower bounds on the suboptimality of random split for constant-rank similarity matrices.

4.4.1 Theoretical Bounds

We first provide an upper bound on the suboptimality of random split (4.1). This shows that random reviewer splits perform well on constant-rank similarity matrices, including the SIGIR similarity matrix examined earlier. More precisely, the following theorem shows that if the similarity matrix S has constant rank k, the suboptimality of random split is at most $\widetilde{O}(n^{-\frac{1}{2}})$ when $k = 1$, $\widetilde{O}(n^{-\frac{1}{2}+o(1)})$ when $k = 2$, and $\widetilde{O}(n^{-\frac{1}{k}+o(1)})$ when $k \geq 3$ with high probability.

Theorem 4.3. *Consider any constants $\beta \in [0, 1]$ and $k \in \mathbb{Z}_+$. There exists n_0 and constants C, η such that, for any $n \geq n_0$ where $\beta n \in \mathbb{Z}_+$ and for any similarity matrix $S \in [0, 1]^{(1+\beta)n \times n}$ of rank k, the suboptimality of random split is at most:*

- $C(\log n)^{\eta} n^{-\frac{1}{2}}$ *if* $k = 1$
- $C(\log n)^{\eta} n^{-\frac{1}{k}+\frac{1}{\log \log n}}$ *if* $k \geq 2$

with probability at least $1 - \frac{1}{n}$ (where $\log$ indicates the base-2 logarithm).

Proof sketch. By Lemma 4 of [128], a rank k similarity matrix $S \in [0, 1]^{m \times n}$ can be factored into vectors $u_r \in \mathbb{R}^k$ for each $r \in \mathcal{R}$ and $v_p \in \mathbb{R}^k$ for each $p \in \mathcal{P}$ such that $S_{r,p} = \langle u_r, v_p \rangle$, $||u_r||_2 \leq k^{1/4}$, and $||v_p||_2 \leq k^{1/4}$. We cover the k-dimensional ball containing all paper vectors with smaller cells, and consider a reviewer to be in one of these cells if the oracle optimal assignment (given $\mathcal{P}_2$) assigns it to a paper in that cell. Using a concentration inequality on the number of reviewers and papers in each cell in each stage, we can upper bound the number of reviewers that we cannot match to papers within the same cell. We then increase the size of the cells and attempt to match the remaining reviewers in this way, continuing until all reviewers are matched. We upper bound the suboptimality of the resulting assignment by the L2 distance between a reviewer's assigned paper and the paper they are assigned by the oracle optimal assignment. □

The constants C and η may depend on k, which is itself assumed to be constant. The full proof is presented in Section 4.7.

For constant-rank similarity matrices, the suboptimality diminishes as n grows, unlike when the rank of the similarity matrix is unrestricted. Conceptually, our proof technique of finding a minimum-distance matching between two samples of points resembles the optimal transport problem solved when finding the Wasserstein distance between a probability distribution and its empirical measure. Thus, our upper bounds nearly match those found in the literature on the expected empirical 1-Wasserstein distance for continuous measures (see [115] and references therein).

We now complement the above upper bound with lower bounds on the suboptimality of random split (4.1) for constant rank similarity matrices. The following theorem shows that, for similarity matrices of constant rank k, the suboptimality of random split is $\Omega(n^{-1/2})$ in expectation and $\Omega(n^{-2/k})$ with high probability.

Theorem 4.4. *Suppose $\beta = 1$. For any constant $k \in \mathbb{Z}_+$, there exists n_0 and constants C, ζ such that for all $n \geq n_0$:*

(a) There exist instances of the two-stage paper assignment problem with similarity matrices $S \in [0, 1]^{2n \times n}$ of rank k such that the suboptimality of random split is at least $Cn^{-1/2}$ in expectation.

(b) There exist instances of the two-stage paper assignment problem with similarity matrices $S \in [0,1]^{2n \times n}$ of rank k such that the suboptimality of random split is at least $Cn^{-2/k}$ with probability $1 - \zeta e^{-n/10}$.

Proof sketch. (a) We construct k groups of reviewers and papers, where all reviewers and papers in the same group have similarity 1 with each other and similarity 0 with all other reviewers/papers. The first group contains $\frac{n}{2}$ papers and n reviewers. The optimal reviewer split puts half of each group's reviewers in each stage and assigns all reviewers to papers with similarity 1. By an anti-concentration inequality, with constant probability, at least $\Omega(\sqrt{n})$ reviewers in the first group cannot be assigned to a paper in their group under random split.

(b) We construct a vector in $\mathbb{R}^k$ for each reviewer and each paper and set the similarity between that reviewer and that paper to be the inner product of their corresponding vectors. We place one paper vector and two reviewer vectors at each point in an evenly-spaced grid throughout the cube $[0, 1/\sqrt{k}]^k$. The resulting similarity matrix has rank k. The optimal assignment assigns the two reviewers at each point to the paper at that point. With high probability, random split places $\Omega(n)$ pairs of reviewer vectors into the same stage. One of each of these reviewer pairs must be assigned to a paper at a different point, which is at least $\Omega(n^{-1/k})$ away in L2 distance. The suboptimality of the resulting assignment can be written in terms of the total squared L2 distance between each reviewer and their assigned paper, giving the stated bound. □

The constants C and ζ may depend on k, which is itself assumed to be constant. The full proof is presented in Section 4.7.

4.4.2 Interpretation of Results

As discussed earlier in this section, certain methods of constructing similarities (such as counting subject area agreements) may inherently lead to low-rank similarity matrices. If a conference is using such a method, the results in this section provide guarantees to the program chairs that random split will perform well, particularly if the rank of the matrix is low compared to the number of papers and reviewers. Alternatively, program chairs may be able to estimate that their reviewers and papers can be grouped into a small number of communities with little variation within them, in which case the similarity matrix may also be low rank.

4.5 Condition 2: High-Value, Large-Load Assignment

A natural condition on the similarity matrix to consider is that each paper has a large number μ of reviewers with high similarity for that paper. It turns out that this condition is insufficient for guaranteeing good performance of random split, since the same group of μ reviewers could have high similarity with all papers, thus satisfying this condition without changing the assignment value by much (since we can only assign these reviewers to a few papers). In this section, we consider a condition on the similarity matrix that is similar in spirit: the existence of a high-value assignment (in terms of total similarity) on the full reviewer and paper sets where each paper is assigned a large number $(1 + \beta)\mu$ of reviewers. Our proposed condition handles the issue with the naive "large number of reviewers" condition by requiring that the high-value reviewers for each paper can all be simultaneously assigned.

In the following subsections, we first provide theoretical guarantees about the performance of random split under this condition. We then demonstrate that this condition helps to explain the good performance of random split on the real similarity matrices presented earlier.

4.5.1 Theoretical Bounds

The first result of this section gives a lower bound on the expected value of random split in terms of the value of a single, large-load assignment. All results in this section still hold if there are excess reviewers (i.e., if $m \geq (1+\beta)n$ and $\mathcal{R}_2 \sim \mathcal{U}_{\frac{\beta}{1+\beta}m}(\mathcal{R})$).

Theorem 4.5. *Consider any $\mu \in [10^4]$ and $\beta \in \left\{\frac{1}{100}, \ldots, \frac{100}{100}\right\}$ such that $\beta\mu \in \mathbb{Z}_+$. If there exists an assignment $Z^{(\mu)} \in \mathcal{M}(\mathcal{R}, \mathcal{P}; \mu, (1+\beta)\mu)$ with mean similarity $s^{(\mu)}$, choosing $\mathcal{R}_2$ via random split gives that*

$$\mathbb{E}_{\mathcal{R}_2}\left[f(\mathcal{R}_2)\right] \geq s^{(\mu)}\left[1 - \sqrt{\frac{\beta}{2\pi(1+\beta)^2\mu}}\left(2\sqrt{\frac{1}{1+\beta}} + \sqrt{1-\beta}\right)\right].$$

A similar bound holds when $\beta\mu$ is not integral, with some additional small terms due to rounding.

Proof sketch. We construct assignments with paper and reviewer loads of at most μ in stage one and at most $\beta\mu$ in stage two using the reviewer-paper pairs assigned by $Z^{(\mu)}$. We drop any extra assignments at random so that no reviewers and papers are overloaded, and assume any pairs that must be assigned from outside of $Z^{(\mu)}$ have similarity 0. From within each of these larger assignments, we can find an assignment with paper and reviewer loads of 1 with at least the same mean similarity. The expected mean similarity of these assignments can be written as the expectation of a function of binomial random variables. Approximating these by normal random variables and checking via simulation that this is in fact a lower bound for the stated values of β and μ, we get the stated bound. □

The more general version of the bound and the full proof are stated in Section 4.7.

The above bound works well when the reviewer-paper pairs in the large-load assignment are all nearly equally valuable. However, it cannot take advantage of the fact that certain reviewers may be extremely valuable for a certain paper and can be prioritized for assignment to that paper when possible. The next result uses additional information about the value of an assignment with smaller loads, along with a large-load assignment disjoint from the small assignment, to make use of these highly valuable reviewer-paper pairs in the case where $\beta = 1$. Recall from Section 4.3.2 that $\beta = 1$ is still not an easy case for random split in general and is particularly relevant for the conference experiment setting.

Theorem 4.6. *Suppose $\beta = 1$, and consider any $\mu \in [10^4]$ such that $\frac{\mu}{4} \in \mathbb{Z}_+$. Suppose there exists an assignment $Z^{(1)} \in \mathcal{M}(\mathcal{R}, \mathcal{P}; 1, 2)$ with mean similarity $s^{(1)}$. Suppose there also exists an assignment $Z^{(\mu)} \in \mathcal{M}(\mathcal{R}, \mathcal{P}; \mu, 2\mu)$ with mean similarity $s^{(\mu)}$ that does not contain any of the pairs assigned in $Z^{(1)}$. Then, choosing $\mathcal{R}_2$ via random split gives that*

$$\mathbb{E}_{\mathcal{R}_2}\left[f(\mathcal{R}_2)\right] \geq \frac{3}{4}s^{(1)} + \left(1 - \frac{1.44}{\sqrt{\mu}}\right)\frac{1}{4}s^{(\mu)}.$$

A similar bound holds when $\frac{\mu}{4}$ is not integral, with some additional small terms due to rounding.

Proof sketch. We first attempt to assign as many pairs as possible from within $Z^{(1)}$; in expectation we can assign $\frac{3}{4}$ of them. Among the remaining reviewers and papers, we attempt to construct assignments with paper and reviewer loads of $\frac{\mu}{4}$ in both stages from within the reviewer-paper pairs assigned by $Z^{(\mu)}$. This is done in a similar way as in Theorem 4.5. □

The more general version of the bound and the full proof are stated in Section 4.7.

If we consider $Z^{(1)}$ as the optimal assignment and assume that μ is divisible by 4, we get an approximation ratio (between the random split assignment and oracle optimal assignment's similarities) of $\frac{3}{4} + \frac{\gamma_\mu}{4}\left(1 - \frac{1.44}{\sqrt{\mu}}\right)$ where $\gamma_\mu = \frac{s^{(\mu)}}{s^{(1)}}$. With $\mu = 8$, we achieve an approximation ratio of at least $\frac{3}{4} + \frac{\gamma_8}{8}$. Additionally, if $\gamma_\mu \to 1$ as n grows for any $\mu = \omega(1)$, the suboptimality of random split (4.1) approaches 0. For example, this means that the suboptimality of random split approaches 0 as n grows if the mean similarity of an assignment with paper loads of $\log n$ improves faster than the mean similarity of the optimal assignment.

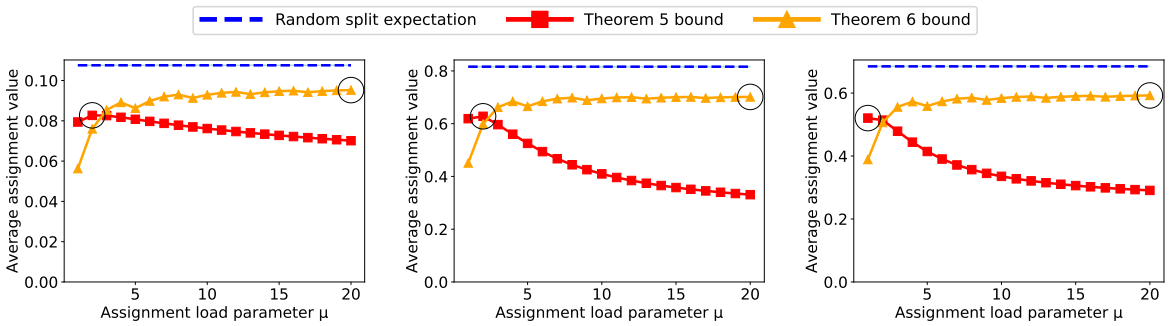
4.5.2 Empirical Evaluation

We now show the performance of these bounds on our real conference datasets in order to evaluate the extent to which they explain the good performance of random split. We use three of the conference datasets introduced earlier with $\beta = 1$. In Section 4.6.1, we evaluate the bounds on additional datasets (including the SIGIR dataset). On PrefLib3 and Bid1, the problem is infeasible with paper and reviewer load constraints of 1 since $m < 2n$, so we modify the datasets by splitting each reviewer into 3 copies as follows. For each paper, we arbitrarily give one of the copies the same similarity as the original reviewer and give the other copies similarity 0. In this way, the similarity of the optimal assignment on this modified dataset is no greater than the similarity of the optimal assignment on the original dataset.

In Figure 4.2, we vary the value of the parameter μ (indicating the loads of the assignment $Z^{(\mu)}$) and show the bounds of Theorem 4.5 and Theorem 4.6 as compared to the estimated expected value of random split. The estimated expected value is averaged over 10 trials with the standard error of the mean shaded, although it is sometimes not visible because it is small. We see that on these datasets, the bound of Theorem 4.5 performs best for low values of μ and not very well for higher values, likely due to the presence of a few "star" reviewers for each paper which hold a lot of the value. By making use of extra information about the values of these reviewers, the bound of Theorem 4.6 achieves a high fraction of the actual random split value. Although this bound is maximized at large values of μ on these datasets, it is close to its maximum even with reasonably low values of μ. For example, on ICLR, the lower bound achieves 86% of the estimated expected value of random split with $\mu = 8$. This indicates the good performance of random split is explained well by the presence of just a few good reviewers per paper that can be simultaneously assigned.

4.5.3 Interpretation of Results

Although our results in this section are stated in terms of the precise values of high-load assignments, they can be interpreted by program chairs in a simple and practical way. Roughly, our results indicate that if several good reviewers can be *simultaneously* assigned to each paper (as was the case for the three conference similarity matrices in Figure 4.2), random split will



(a) ICLR (b) PrefLib3 (c) Bid1

Figure 4.2: Performance of the "high-value large-load" bounds on real conference datasets, $\beta = 1$. On the x-axis we vary the parameter μ, which determines the loads of the assignment $Z^{(\mu)}$ used in the bound. The best setting of μ for each bound is circled.

perform well. When considering the potential performance of randomly splitting reviewers, program chairs should consider the reviewer and paper pools they expect to have at their conference and make a judgement about how many good-quality reviewers they think could be assigned to each paper (if the reviewer loads are scaled up proportionately). For example, the program chairs of a large AI conference might be confident that the top several reviewers for each paper are about equally valuable (due to the depth of the reviewer pool) and could be assigned to each paper with only a modest loss in average review quality; this would imply that random split would perform very well for this conference.

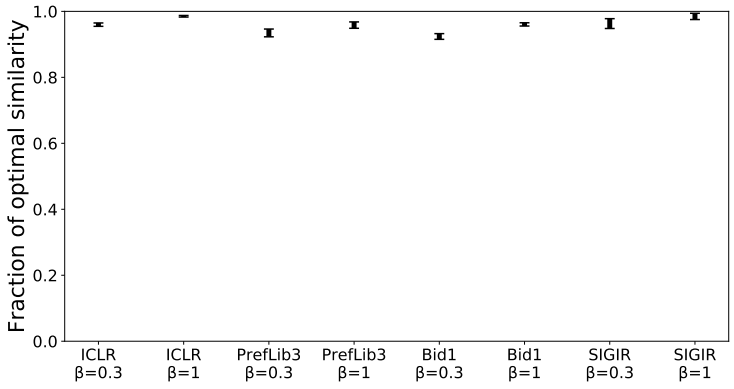
4.6 Supplemental Material

In this section, we present additional empirical results and analysis.

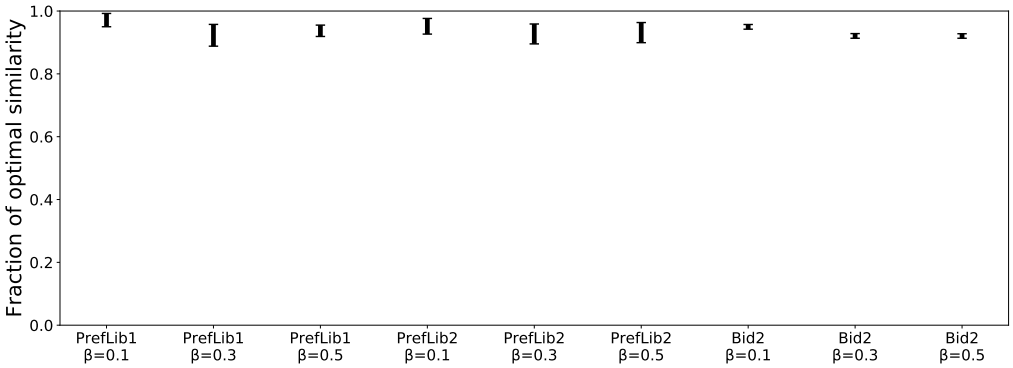
4.6.1 Additional Empirical Results

We first present empirical evaluations showing the performance of random split on additional values of β for the similarity matrices used in Section 4.3.1 (Figure 4.3a), as well as on additional similarity matrices constructed from bidding data (Figure 4.3b). Two of these additional similarity matrices were constructed using bid data for the two other AI conferences (conferences 1 and 2) from PrefLib dataset MD-00002 [106] with sizes $(n = 54, m = 31)$ and $(n = 52, m = 24)$ respectively. Another additional similarity matrix (marked Bid2) was constructed from another sample of the bidding data from a major computer science conference [108] with size $(n = 1200, m = 300)$. As in the bidding datasets shown earlier, we transformed "yes," "maybe," and "no response" bids into similarities of 1, 0.5, and 0.25 respectively.

We run several experiments, each corresponding to a choice of dataset and β. Each experiment consists of 10 trials, where in each trial we sample a random reviewer split and a set of second-stage papers, and report the range of assignment values found as percentages of the oracle optimal assignments for each trial. We set paper loads of 2 in each stage, and limit reviewer loads to be at most 6 for all datasets except PrefLib2 and Bid2, which limit reviewer loads to be at most 12 (for feasibility). As in Section 4.3.1, we draw $\mathcal{R}_2$ uniformly at random with size $\frac{\beta}{1+\beta}m$



(a) Additional values of β



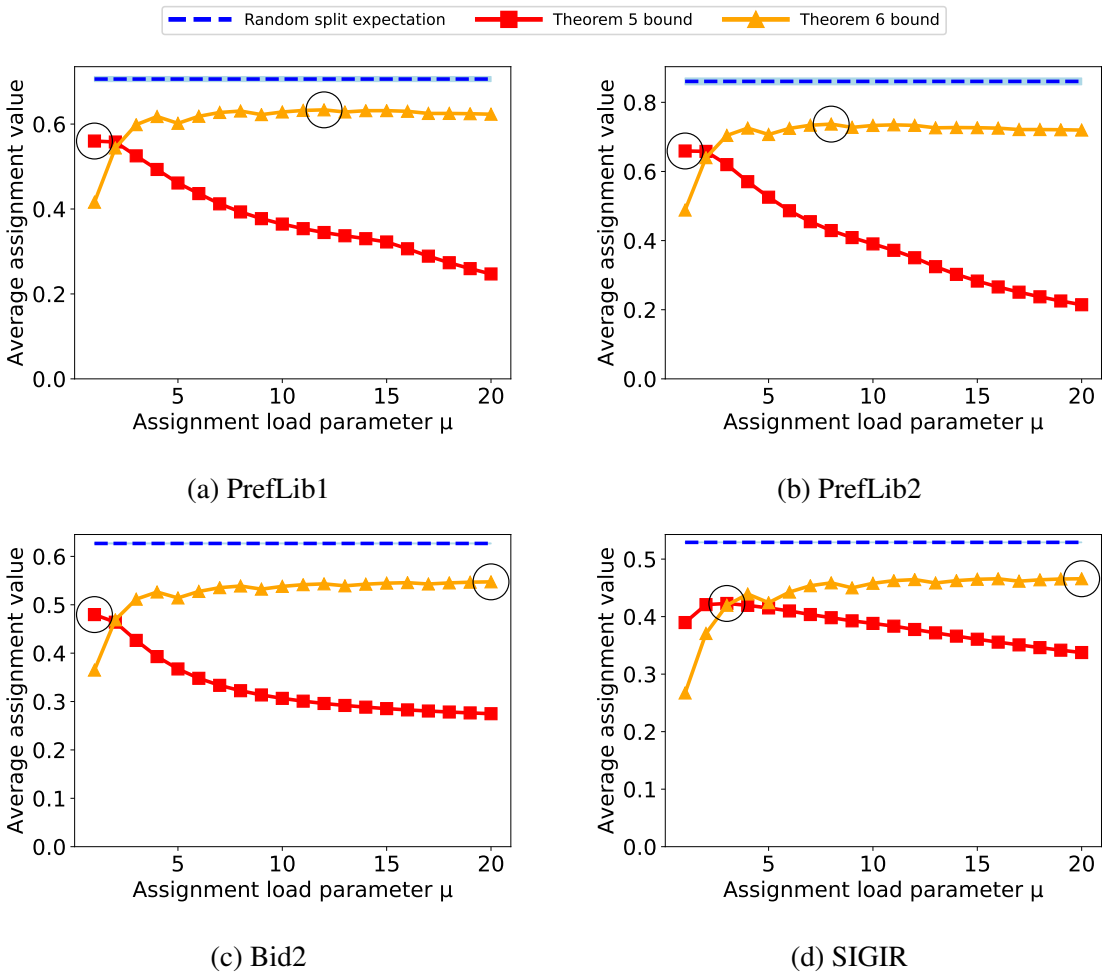
(b) Additional similarity matrices

Figure 4.3: Additional results showing ranges of values found over 10 random reviewer splits.

and draw $\mathcal{P}_2$ uniformly at random with size βn. In general, we see that random split performs very well on these datasets as well. We see that all trials of random split achieve at least 88% of the oracle optimal solution's similarity on all datasets, with all trials on all but three experiments achieving at least 94%. The range of values on each experiment is generally small (at most 7%), with the largest ranges occuring on the small PrefLib datasets.

We additionally test the bounds of Section 4.5 on these datasets as well as the SIGIR dataset to evaluate how well they explain the performance of random split. On PrefLib1, Preflib2, and Bid2, we scale up the number of reviewers by 4, 5, and 8 respectively for feasibility, as described in Section 4.5.2. On the x-axis we vary the parameter μ, which determines the loads of the assignment $Z^{(\mu)}$ used in the bound.

In Figure 4.4, we see similar results to those shown earlier. The Theorem 4.5 bound performs best at low values of μ. The Theorem 4.6 bound performs better at higher values of μ, although for some datasets a more moderate value of μ is better since the assignment value $s^{(\mu)}$ drops too quickly at higher μ. From the Theorem 4.6 bound, we see that the good performance of random split on these datasets is generally explained fairly well by the large-load, high-similarity assignment.



(a) PrefLib1

(b) PrefLib2

(c) Bid2

(d) SIGIR

Figure 4.4: Performance of Theorem 4.5 and 4.6 bounds on additional real conference datasets, $\beta = 1$. The best setting of μ shown for each bound is circled.

All empirical evaluations in this paper were run on a computer with 8 cores and 16 GB of RAM, running Ubuntu 18.04 and solving the LPs with Gurobi 9.0.2 [101].

4.6.2 Empirical Results for Paper-Split Variant

In this section, we provide some additional empirical results that are particularly relevant to the conference experiment design setting. Sometimes, conferences may not have the reviewing resources to provide a significant number of papers with two sets of reviews as part of an experiment. Instead, they may want to provide reviews to each paper under only one of the conditions. If the papers and reviewers are both split between conditions uniformly at random, this can be seen as a variant of our standard two-stage paper assignment problem where only papers in $\mathcal{P}_1 = \mathcal{P} \setminus \mathcal{P}_2$ are assigned reviewers in stage one.

To test whether such experiments will still give high-similarity assignments in practice, we conduct additional empirical evaluations. The results of these experiments are shown in Figure 4.5. For each dataset of those introduced in Section 4.3.1, we take 10 samples of random

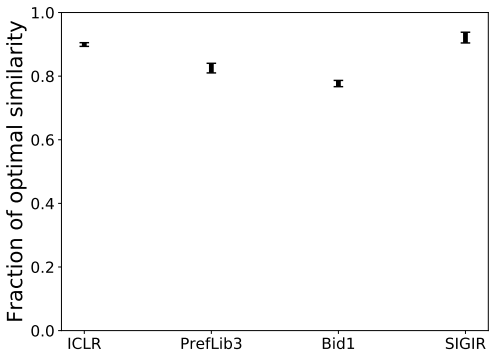


Figure 4.5: Range of values found over 10 random reviewer splits when papers split between stages.

reviewer and paper splits where $|\mathcal{R}_2| = m/2$ and $|\mathcal{P}_2| = n/2$ so that half of the reviewers and papers are in each stage (i.e., each condition). We then find assignments in each stage with paper loads of 3 and reviewer loads of at most 6 (standard conference loads), and display the range of assignment values found as a fraction of the oracle optimal assignment's value. On all datasets, all trials of random reviewer split achieve over 75% of the oracle optimal assignment's total similarity with low variation (at most 4%). On ICLR and SIGIR, all trials achieve over 90% of the oracle optimal similarity. Overall, the average assignment quality is slightly worse than in the standard model (where all papers are in stage one). This is likely because it is more difficult for reviewers to be assigned to their optimal papers when each paper is in only one of the two stages.

4.6.3 Submodularity of Objective Function

In this section, we show that the problem of optimizing f (or $\overline{f}$) is actually an instance of submodular optimization. For simplicity, we consider the case where $m = (1+\beta)n$ and $\ell_r = \ell_p^{(1)} = \ell_p^{(2)} = 1$.

For some set $\mathcal{N}$, a function $g : 2^{\mathcal{N}} \to \mathbb{R}$ is submodular if $g(A \cup \{u\}) - g(A) \geq g(B \cup \{u\}) - g(B)$ for all $A \subseteq B \subseteq \mathcal{N}$ and all $u \in \mathcal{N} \setminus B$. Since f and $\overline{f}$ are defined only for $\mathcal{R}_2$ where $|\mathcal{R}_2| = \beta n$, we modify them to be defined over $2^{\mathcal{R}}$.

Recall that for subsets $\mathcal{R}' \subseteq \mathcal{R}$ and $\mathcal{P}' \subseteq \mathcal{P}$, desired paper load ℓ_p, and maximum reviewer load ℓ_r, $\mathcal{M}(\mathcal{R}', \mathcal{P}'; \ell_r, \ell_p)$ is the set of assignment matrices that assign a load of exactly ℓ_p to all papers in $\mathcal{P}'$ (and a load of at most ℓ_r to all reviewers in $\mathcal{R}'$). Define $\mathcal{M}'(\mathcal{R}', \mathcal{P}'; \ell_r, \ell_p)$ as the set of assignment matrices that assign a load of at most ℓ_p to all papers in $\mathcal{P}'$ (and a load of at most ℓ_r to all reviewers in $\mathcal{R}'$). Formally, $Z \in \mathcal{M}'(\mathcal{R}', \mathcal{P}'; \ell_r, \ell_p)$ if and only if $\sum_{r \in \mathcal{R}'} Z_{r,p} \leq \ell_p$ for all $p \in \mathcal{P}'$, $\sum_{p \in \mathcal{P}'} Z_{r,p} \leq \ell_r$ for all $r \in \mathcal{R}'$, and $Z_{r,p} = 0$ for all $(r,p) \notin \mathcal{R}' \times \mathcal{P}'$.

Consider the modified version of Q

$$Q'(\mathcal{R}_2, \mathcal{P}_2) = \frac{1}{(1+\beta)n} \left[\max_{A \in \mathcal{M}'(\mathcal{R} \setminus \mathcal{R}_2, \mathcal{P}; 1, 1)} \sum_{r \in \mathcal{R} \setminus \mathcal{R}_2, p \in \mathcal{P}} A_{r,p} S_{r,p} + \max_{B \in \mathcal{M}'(\mathcal{R}_2, \mathcal{P}_2; 1, 1)} \sum_{r \in \mathcal{R}_2, p \in \mathcal{P}_2} B_{r,p} S_{r,p} \right]$$

which allows papers to be underloaded and so is defined for all $\mathcal{R}_2$ and $\mathcal{P}_2$. Define $f_{sub}(\mathcal{R}_2) =$

$\mathbb{E}_{\mathcal{P}_2}[Q'(\mathcal{R}_2, \mathcal{P}_2)]$ and $\overline{f}_{sub}(\mathcal{R}_2) = \frac{1}{K}\sum_{i=1}^{K} Q'(\mathcal{R}_2, \mathcal{P}_2^{(i)})$ as modifications of f and $\overline{f}$. Since $S \geq 0$, there exists a maximum-similarity assignment from within $\mathcal{M}'(\mathcal{R}', \mathcal{P}'; 1, 1)$ that meets all paper load constraints with equality when $|\mathcal{R}'| \geq |\mathcal{P}'|$ and thus is contained in $\mathcal{M}(\mathcal{R}', \mathcal{P}'; 1, 1)$. Also, $\mathcal{M}(\mathcal{R}', \mathcal{P}'; 1, 1) \subseteq \mathcal{M}'(\mathcal{R}', \mathcal{P}'; 1, 1)$. Thus, when $|\mathcal{R}_2| = \beta n$, $Q(\mathcal{R}_2, \mathcal{P}_2) = Q'(\mathcal{R}_2, \mathcal{P}_2)$. Therefore, subject to the constraint $|\mathcal{R}_2| = \beta n$, maximizing f_{sub} (or $\overline{f}_{sub}$) is equivalent to maximizing f (or $\overline{f}$).

Proposition 4.1. *f_{sub} and $\overline{f}_{sub}$ are submodular in $\mathcal{R}_2$.*

Proof. Note that $\max_{Z \in \mathcal{M}'(\mathcal{R}', \mathcal{P}'; 1, 1)} \sum_{r \in \mathcal{R}', p \in \mathcal{P}'} Z_{r,p} S_{r,p}$ is a submodular function of the reviewer set $\mathcal{R}'$ when the paper set $\mathcal{P}'$ is held fixed [89]. Submodularity in $\mathcal{R}_2$ is equivalent to submodularity in $\mathcal{R}_1 = \mathcal{R} \setminus \mathcal{R}_2$, so $Q'(\mathcal{R}_2, \mathcal{P}_2)$ is submodular in $\mathcal{R}_2$. As sums over terms submodular in $\mathcal{R}_2$, f_{sub} and $\overline{f}_{sub}$ are submodular in $\mathcal{R}_2$. □

Therefore, the two-stage paper assignment problem is an instance of maximizing a non-monotone submodular function subject to a cardinality constraint $|\mathcal{R}_2| = \beta n$. However, no value oracle for the function f is available due to the expectation over $\mathcal{P}_2$. Since a polynomial-time value oracle is available for $\overline{f}$, the paper [24] gives an approximation algorithm achieving an approximation ratio of no greater than 0.5 (depending on β). This guarantee does not imply much about the assignment quality, since it can be trivially achieved by maximizing assignment similarity in the first stage only. Furthermore, it is known that achieving an approximation ratio of greater than 0.5 requires an exponential number of queries to the value oracle; this holds true even without the cardinality constraint [46, 55]. Thus, generic algorithms for submodular maximization are not helpful for our problem.

4.7 Omitted Proofs

In this section, we present the proofs omitted from the previous sections.

Proof of Theorem 4.1

We will show that it is NP-hard to determine if there exists a choice of $\mathcal{R}_2$ with value $\overline{f}(\mathcal{R}_2) = 1$ when $K = 3$, for some instance of $\mathcal{P}_2^{(1)}, \mathcal{P}_2^{(2)}, \mathcal{P}_2^{(3)}$. If such a choice exists, it would be the optimal solution. Therefore, any algorithm to optimize $\overline{f}$ would be able to determine if there exists a solution with value 1, solving an NP-hard problem. This implies the NP-hardness of optimizing $\overline{f}$.

We reduce from 3-Dimensional Matching, an NP-complete problem [79]. An instance of 3-Dimensional Matching consists of three sets A, B, C of size s, and a collection of tuples $T \subseteq A \times B \times C$. It asks whether there exists a selection of s tuples from T that includes each element of A, B, and C exactly once.

Given such an instance of 3-Dimensional Matching, we construct an instance of the two-stage paper assignment problem with $n = |T| + 2s$, $m = |T| + 3s$, and $\beta = \frac{m}{n} - 1$ ($\ell_r = \ell_p^{(1)} = \ell_p^{(2)} = 1$). $\beta n = s$ papers and reviewers will be in stage two. The first s papers correspond to elements of A, the next s to elements of B, and the next s to elements of C; the remaining $|T| - s$ papers are "dummy papers" that all reviewers can review. The first $3s$ reviewers are "specialty reviewers" corresponding to each of the first $3s$ papers, and the remaining $|T|$ reviewers correspond to each of the elements of T. We construct the $K = 3$ sampled subsets $\mathcal{P}_2^{(1)} = \{1, \ldots, s\}, \mathcal{P}_2^{(2)} =$

$\{s+1,\ldots,2s\}, \mathcal{P}_2^{(3)} = \{2s+1,\ldots,3s\}$, where the elements of these sets correspond to the elements of A, B, and C respectively. We then construct S as follows. For $i \in [3s]$, set $S_{i,i} = 1$ and $S_{i,j} = 0$ for all $j \in [3s], j \neq i$. For the remaining reviewers $i \in \{3s+1,\ldots,3s+|T|\}$ and for papers $j \in [3s]$, set $S_{i,j} = 1$ if the element corresponding to j in $A \cup B \cup C$ is included in the tuple corresponding to i in T. Finally, for the remaining papers $j \in \{3s+1,\ldots,|T|+2s\}$, set $S_{i,j} = 1$ for all reviewers i.

Suppose we have a "yes" instance of 3-Dimensional Matching, so there exists a choice of s tuples from T that cover each element of A, B, and C. Choose the corresponding s reviewers as $\mathcal{R}_2$ and the remaining reviewers as $\mathcal{R}_1$. In stage one, we can assign each specialty reviewer to each of their corresponding papers and each of the remaining $|T|-s$ reviewers in $\mathcal{R}_1$ to dummy papers. In stage two, for each of the three possible samples, there exists one reviewer that has similarity 1 with each paper since the corresponding choice of tuples from T cover A, B, and C. Therefore, this partition achieves $\overline{f}(\mathcal{R}_2) = 1$.

Suppose we have a "no" instance of 3-Dimensional Matching, so no choice of s tuples from T covers each element of A, B, and C. We claim that no choice of $\mathcal{R}_2$ will achieve s total similarity in the second stage. First, suppose we include a speciality reviewer in $\mathcal{R}_2$. This reviewer has similarity 1 with only one paper, so there exists a sample of stage two papers $\mathcal{P}_2^{(i)}$ such that this reviewer must be assigned to a paper with which it has similarity 0. Therefore, $\overline{f}(\mathcal{R}_2)$ cannot be 1 when a specialty reviewer is in $\mathcal{R}_2$ and so $\mathcal{R}_2$ must be chosen from the reviewers corresponding to elements of T. However, no choice of s tuples covers each element of A, B, and C. Therefore, for every choice of $\mathcal{R}_2$, some reviewer must be assigned to a paper with which they have similarity 0 for at least one of the sampled sets of stage two papers. This means that $\overline{f}(\mathcal{R}_2) = 1$ is unachievable.

Proof of Theorem 4.2

For any $\beta \in [0,1]$, choose any n such that $\beta n \in \mathbb{Z}_+$. We construct the following similarity matrix. Paper i has similarity 1 with reviewer i, and also with reviewer $n+i$ if $i \leq \beta n$. All other similarities are 0.

On this example, the oracle optimal assignment for any $\mathcal{P}_2$ is to assign reviewers $\{1,\ldots,n\}$ to papers in the first stage, since this maximizes the similarity across both stages. This choice gives a total similarity of n in stage one and an expected similarity of $\beta^2 n$ in stage two (since each reviewer's matching paper is in stage two with probability β), for a total similarity of $n(1+\beta^2)$. Since there are $(1+\beta)n$ total assignments, the expected mean similarity is $\frac{1+\beta^2}{1+\beta}$.

Now consider the assignment after randomly splitting reviewers. Any paper $p \leq \beta n$ has two reviewers a, b with similarity 1. For sufficiently large $n \geq \frac{1+4\beta}{1+\beta}$, the expected value of this paper's assignment is

$$
\begin{aligned}
&(\mathbb{P}[a \in \mathcal{R}_1 \wedge b \in \mathcal{R}_2] + \mathbb{P}[b \in \mathcal{R}_1 \wedge a \in \mathcal{R}_2])(1 + \mathbb{P}[p \in \mathcal{P}_2]) \\
&\qquad + \mathbb{P}[a,b \in \mathcal{R}_1] + \mathbb{P}[a,b \in \mathcal{R}_2]\mathbb{P}[p \in \mathcal{P}_2] \\
&= \left(2\frac{n}{(1+\beta)n}\frac{\beta n}{(1+\beta)n-1}\right)(1+\beta) + \frac{n}{(1+\beta)n}\frac{n-1}{(1+\beta)n-1} + \frac{\beta n}{(1+\beta)n}\frac{\beta n-1}{(1+\beta)n-1}\beta \\
&\leq 2\frac{\beta}{(1+\beta)-\frac{1}{n}} + \frac{1}{(1+\beta)^2} + \frac{\beta^3}{(1+\beta)^2}
\end{aligned}
$$

$$\leq \frac{1+4\beta}{2(1+\beta)} + \frac{1}{(1+\beta)^2} + \frac{\beta^3}{(1+\beta)^2}.$$

There are βn of these papers.

Any of the remaining papers $p > \beta n$ has only one reviewer a with similarity 1. The expected value of this paper's assignment is

$$\begin{aligned}&\mathbb{P}[a \in \mathcal{R}_1] + \mathbb{P}[a \in \mathcal{R}_2]\mathbb{P}[p \in \mathcal{P}_2] \\ &= \frac{1+\beta^2}{1+\beta}.\end{aligned}$$

There are $(1-\beta)n$ of these papers.

Totalling over all papers and dividing by the total number of assignments, the mean expected similarity of random split is at most

$$\left(\frac{1+4\beta}{2(1+\beta)} + \frac{1}{(1+\beta)^2} + \frac{\beta^3}{(1+\beta)^2}\right)\frac{\beta}{1+\beta} + \frac{(1+\beta^2)(1-\beta)}{(1+\beta)^2}.$$

The suboptimality is therefore at least

$$\begin{aligned}&\frac{1+\beta^2}{1+\beta} - \left(\frac{1+4\beta}{2(1+\beta)} + \frac{1}{(1+\beta)^2} + \frac{\beta^3}{(1+\beta)^2}\right)\frac{\beta}{1+\beta} - \frac{(1+\beta^2)(1-\beta)}{(1+\beta)^2} \\ &= \frac{(1+\beta^2)2\beta}{(1+\beta)^2} - \left(\frac{1+4\beta}{2(1+\beta)} + \frac{1}{(1+\beta)^2} + \frac{\beta^3}{(1+\beta)^2}\right)\frac{\beta}{1+\beta} \\ &= \left(2(1+\beta^2)(1+\beta) - \frac{1}{2}(1+4\beta)(1+\beta) - 1 - \beta^3\right)\frac{\beta}{(1+\beta)^3} \\ &= \left(\frac{1}{2} - \frac{1}{2}\beta + \beta^3\right)\frac{\beta}{(1+\beta)^3} \\ &\geq \frac{\beta^4}{(1+\beta)^3}.\end{aligned}$$

Proof of Theorem 4.3

By Lemma 4 of [128], a rank k similarity matrix $S \in [0,1]^{(1+\beta)n \times n}$ can be factored into vectors $u_r \in \mathbb{R}^k$ for each reviewer r and $v_p \in \mathbb{R}^k$ for each paper p such that $S_{r,p} = \langle u_r, v_p \rangle$, $||u_r||_2 \leq k^{1/4}$, and $||v_p||_2 \leq k^{1/4}$.

Consider the ball of radius $k^{1/4}$ in $\mathbb{R}^k$ in which the paper vectors v_p lie. We cover this ball with smaller "cells" by dividing the containing k-dimensional hypercube with side length $2k^{1/4}$ along each dimension to create some number of smaller hypercubes. If we divide the containing hypercube into t equal-sized segments along each dimension, there are t^k cells in total and the maximum L2 distance between two points in a cell is $\frac{2k^{3/4}}{t}$.

We construct L layers of cells in this way, where the cells increase in size between layers. Denote by t_i the number of divisions along each dimension at layer i. We choose $t_i = 2^{A_i}$ for some integer A_i for all layers i so that each cell at layer i is fully contained within a single cell at each higher layer. Denote by s_i the desired maximum within-cell distance at layer i. This

distance is achieved if t_i is at least $\frac{2k^{3/4}}{s_i}$, so the minimum such t_i that is also a power of two is at most $\frac{4k^{3/4}}{s_i}$. This gives that there are at most $z_i = \left(\frac{4k^{3/4}}{s_i}\right)^k$ cells in layer i.

In what follows, we say that a paper p is in some cell if its vector v_p is in the cell. (Papers on the border of multiple cells at layer 1 are considered to be in an arbitrary one of the bordering cells so that each paper is in exactly one cell. At higher layers, such papers are considered to be in the cell containing their layer 1 cell.) We say that a reviewer is in a cell if it is assigned to a paper in that cell by the oracle optimal paper assignment (given $\mathcal{P}_2$).

Given $\mathcal{P}_2$ and $\mathcal{R}_2$ produced by random split, we proceed through layers from 1 to L in order to match reviewers to papers in the same stage. We match as many reviewers as possible to papers that are within the same cell at each layer i, and then continue to layer $i+1$. Define n_i as an upper bound on the number of reviewers unmatched before matching within layer i; $n_1 = (1+\beta)n$. The difference in value between the assignment Z produced in this way and the oracle optimal assignment Z^* (which we call the "value gap") is

$$\begin{aligned}\sum_{r\in\mathcal{R},p\in\mathcal{P}} (Z^*_{r,p} - Z_{r,p})\langle u_r, v_p\rangle &= \sum_{r\in\mathcal{R},p\in\mathcal{P},p^*\in\mathcal{P}} Z_{r,p}Z^*_{r,p^*}\langle u_r, v_{p^*} - v_p\rangle \\ &\le \sum_{r\in\mathcal{R},p\in\mathcal{P},p^*\in\mathcal{P}} Z_{r,p}Z^*_{r,p^*}||u_r||_2||v_{p^*} - v_p||_2 \\ &\le k^{1/4} \sum_{r\in\mathcal{R},p\in\mathcal{P},p^*\in\mathcal{P}} Z_{r,p}Z^*_{r,p^*}||v_{p^*} - v_p||_2.\end{aligned}$$

Consider some cell containing x papers. All x of these papers are in stage one. Define $Hyp(N,K,M)$ as the hypergeometric distribution where N is the population size, M is the number of draws, and K is the number of successes in the population; by symmetry $Hyp(N,K,M)$ is equivalent to $Hyp(N,M,K)$. The number of stage two papers has distribution $Hyp(n,x,\beta n)$. With probability $1-2\delta$, by Hoeffding's inequality [64] and using the symmetry property, within $\beta x \pm \sqrt{\frac{x}{2}\ln(1/\delta)}$ of the papers in this cell are also in stage two. (In this section, $\ln$ indicates the logarithm with base e and $\log$ indicates the logarithm with base 2.) Call y the total number of reviewers in the cell. There are exactly the same number of reviewers as total stage one and two papers in this cell, so y is within $(1+\beta)x \pm \sqrt{\frac{x}{2}\ln(1/\delta)}$ and is at most $2x$. Since $\mathcal{R}_2$ is produced by random split, the number of reviewers in this cell in stage one has distribution $Hyp((1+\beta)n, y, n)$ and the number of reviewers in this cell in stage two has distribution $Hyp((1+\beta)n, y, \beta n)$. By Hoeffding's inequality and again using symmetry, the number of reviewers in the cell in stage one is at most $\frac{y}{1+\beta} + \sqrt{\frac{y}{2}\ln(1/\delta)} \le x + \sqrt{\frac{x}{2}\ln(1/\delta)} + \sqrt{x\ln(1/\delta)} \le x + \sqrt{3x\ln(1/\delta)}$ with probability $1-2\delta$ (conditioned on the earlier event concerning the number of stage-two papers). By this argument, with probability $1-4\delta$ (again conditioned on the earlier event), there are within $x \pm \sqrt{3x\ln(1/\delta)}$ reviewers in stage one in the cell and within $\beta x \pm \sqrt{3x\ln(1/\delta)}$ reviewers in stage two in the cell. In total, there are at most nL cells with a non-zero number of papers across all layers and so the total probability of error in any of the bounds is at most $6\delta Ln$.

Assume that this high probability event occurs. In layer i, in any cell j with x_j papers (all of which are in stage one), the number of stage one reviewers is within $x_j \pm \sqrt{3x_j\ln(1/\delta)}$. Any reviewers in this cell matched at earlier layers must have been matched to papers also in this cell. Therefore, the number of unmatched stage one reviewers after matching within this cell

is at most $\sqrt{3x_j \ln(1/\delta)}$. The number of stage two reviewers is within $\beta x_j \pm \sqrt{3x_j \ln(1/\delta)}$ and the number of stage two papers is within $\beta x_j \pm \sqrt{\frac{x_j}{2} \ln(1/\delta)}$. Therefore, the number of unmatched stage two reviewers after matching within this cell is at most $\sqrt{6x_j \ln(1/\delta)}$. In total over both stages, the total number of unmatched reviewers after matching in layer i is at most $n_{i+1} = \sum_{j=1}^{z_i} \sqrt{18x_j \ln(1/\delta)} \leq \sqrt{18 z_i n \ln(1/\delta)}$. All of the reviewers matched at layer i are matched to papers at most s_i away from their optimal paper assignment. Across all layers, the value gap is therefore bounded by $k^{1/4}\left(\sum_{i=1}^{L-1} n_i s_i + 2n_L k^{1/4}\right)$, since everything at layer L is matched to whatever remains regardless of s_L.

We now determine how to set s_i for all layers i. We choose $s_1 = s$ and set other s_i such that $n_i s_i = (1+\beta)ns$ for all i. This leads to the recursively-defined values of $s_i = \frac{(1+\beta)ns}{n_i}$, $z_i = (4k^{3/4})^k s_i^{-k}$, $n_{i+1} = \sqrt{z_i n 18 \ln(1/\delta)}$ with initial values $n_1 = (1+\beta)n$ and $s_1 = s$. Unrolling the iteration, we see that

$$\begin{aligned} s_i &= s_{i-1}^{\frac{k}{2}} n^{\frac{1}{2}} s (4k^{3/4})^{-\frac{k}{2}} (18\ln(1/\delta))^{-\frac{1}{2}} (1+\beta) \\ &= n^{\frac{1}{2}\sum_{j=0}^{i-2}\left(\frac{k}{2}\right)^j} s^{\sum_{j=0}^{i-1}\left(\frac{k}{2}\right)^j} (4k^{3/4})^{-\frac{k}{2}\sum_{j=0}^{i-2}\left(\frac{k}{2}\right)^j} (18\ln(1/\delta))^{-\frac{1}{2}\sum_{j=0}^{i-2}\left(\frac{k}{2}\right)^j} (1+\beta)^{\sum_{j=0}^{i-2}\left(\frac{k}{2}\right)^j} \end{aligned}$$

for $i \geq 2$. Defining ϵ such that $s = \left(\frac{(1+\beta)^2 n}{18\ln(1/\delta)}\right)^{\epsilon}$,

$$s_i = \left(\frac{(1+\beta)^2 n}{18\ln(1/\delta)}\right)^{\frac{1}{2}\sum_{j=0}^{i-2}\left(\frac{k}{2}\right)^j + \epsilon\sum_{j=0}^{i-1}\left(\frac{k}{2}\right)^j} (4k^{3/4})^{-\frac{k}{2}\sum_{j=0}^{i-2}\left(\frac{k}{2}\right)^j}$$

for $i \geq 2$. This gives a value gap of at most $k^{1/4}(1+\beta)^{1+2\epsilon} n^{1+\epsilon} (18\ln(1/\delta))^{-\epsilon}\left((L-1) + 2k^{1/4} s_L^{-1}\right)$. We now continue in cases on the value of k.

Case $k = 1$**.** Note that $\sum_{j=0}^{i-1}\left(\frac{k}{2}\right)^j = 2\left(1 - \frac{1}{2^i}\right)$, so

$$s_i = \left(\frac{(1+\beta)^2 n}{18\ln(1/\delta)}\right)^{1 - \frac{1}{2^{i-1}} + \epsilon 2\left(1-\frac{1}{2^i}\right)} (4k^{3/4})^{-1+\frac{1}{2^{i-1}}}.$$

Choose $\epsilon = -\frac{1}{2} + \frac{1}{2(2^L-1)}$ so that $s_L = (4k^{3/4})^{-1+\frac{1}{2^{L-1}}}$. Setting $\delta = (2n)^{-3}$ and $L = \log\log n$, for sufficiently large n, the value gap is bounded by

$$\begin{aligned} &k^{1/4}(1+\beta)^{\frac{1}{2^L-1}} n^{\frac{1}{2}+\frac{1}{2(2^L-1)}} (18\ln(1/\delta))^{\frac{1}{2}-\frac{1}{2(2^L-1)}} \left((L-1) + 2k^{1/4}(4k^{3/4})^{1-\frac{1}{2^L-1}}\right) \\ &\leq (1+\beta) n^{\frac{1}{2}} 2^{\log(n)\frac{1}{2(\log n - 1)}} (54\ln(2n))^{\frac{1}{2}} \left((\log\log n - 1) + 8\right) \\ &\leq 2(1+\beta) n^{\frac{1}{2}} (54\ln(2n))^{\frac{1}{2}} \left((\log\log n - 1) + 8\right) \\ &\leq C(\log n)^{\eta} n^{\frac{1}{2}} \end{aligned}$$

for some constants C, η with probability $1 - \frac{6\log\log n}{8n^2} \geq 1 - \frac{1}{n}$.

Case $k = 2$**.** Note that $\sum_{j=0}^{i-1}\left(\frac{k}{2}\right)^j = i$, so

$$s_i = \left(\frac{(1+\beta)^2 n}{18\ln(1/\delta)}\right)^{\frac{1}{2}(i-1)+\epsilon i} (4k^{3/4})^{-i+1}.$$

Choose $\epsilon = -\frac{1}{2} + \frac{1}{2L}$ so that $s_L = (4k^{3/4})^{-L+1}$. Setting $\delta = (2n)^{-3}$ and $L = \log\log n$, for sufficiently large n, the value gap is bounded by

$$\begin{aligned}
&k^{1/4}(1+\beta)^{\frac{1}{L}} n^{\frac{1}{2}+\frac{1}{2L}} (18\ln(1/\delta))^{\frac{1}{2}-\frac{1}{2L}} \left((L-1) + 2k^{1/4}(4k^{3/4})^{L-1}\right) \\
&\leq k^{1/4}(1+\beta) n^{\frac{1}{2}+\frac{1}{2\log\log n}} (54\ln(2n))^{\frac{1}{2}} \left((\log\log n - 1) + 2k^{1/4}(\log n)^{\log(4k^{3/4})}\right) \\
&\leq C(\log n)^{\eta} n^{\frac{1}{2}+\frac{1}{\log\log n}}
\end{aligned}$$

for some constants C, η with probability $1 - \frac{6\log\log n}{8n^2} \geq 1 - \frac{1}{n}$.

Case $k \geq 3$. Note that $\sum_{j=0}^{i-1} \left(\frac{k}{2}\right)^j = \frac{\left(\frac{k}{2}\right)^i - 1}{\frac{k}{2}-1}$, so

$$s_i = \left(\frac{(1+\beta)^2 n}{18\ln(1/\delta)}\right)^{\left(\frac{1}{2}+\epsilon\right)\left(\frac{(k/2)^i-1}{(k/2)-1}\right) - \frac{1}{2}\left(\frac{k}{2}\right)^{i-1}} (4k^{3/4})^{-\frac{k}{2}\left(\frac{(k/2)^i-1}{(k/2)-1}\right) + \left(\frac{k}{2}\right)^i}.$$

Choose $\epsilon = -\frac{1}{k} + \frac{\left(\frac{1}{2}-\frac{1}{k}\right)}{(k/2)^L - 1}$ so that $s_L = (4k^{3/4})^{-\frac{k}{2}\left(\frac{(k/2)^L-1}{(k/2)-1}\right) + \left(\frac{k}{2}\right)^L}$. Setting $\delta = (2n)^{-3}$ and $L = \frac{\log\log\log n}{\log(k/2)}$, for sufficiently large n, the value gap is bounded by

$$\begin{aligned}
&k^{1/4}(1+\beta)^{1-\frac{2}{k}+\frac{2\left(\frac{1}{2}-\frac{1}{k}\right)}{(k/2)^L-1}} n^{1-\frac{1}{k}+\frac{\left(\frac{1}{2}-\frac{1}{k}\right)}{(k/2)^L-1}} (18\ln(1/\delta))^{\frac{1}{k}-\frac{\left(\frac{1}{2}-\frac{1}{k}\right)}{(k/2)^L-1}} \\
&\qquad \left((L-1) + 2k^{1/4}(4k^{3/4})^{\frac{k}{2}\left(\frac{(k/2)^L-1}{(k/2)-1}\right) - \left(\frac{k}{2}\right)^L}\right) \\
&\leq k^{1/4}(1+\beta) n^{1-\frac{1}{k}+\frac{\left(\frac{1}{2}-\frac{1}{k}\right)}{\log\log n - 1}} (54\ln(2n))^{\frac{1}{k}} \left(\frac{\log\log\log n}{\log(k/2)} - 1 + 2k^{1/4}(4k^{3/4})^{2\log\log n}\right) \\
&\leq k^{1/4}(1+\beta) n^{1-\frac{1}{k}+\frac{\left(\frac{1}{2}-\frac{1}{k}\right)}{\log\log n - 1}} (54\ln(2n))^{\frac{1}{k}} \left(\frac{\log\log\log n}{\log(k/2)} - 1 + 2k^{1/4}(\log n)^{2\log(4k^{3/4})}\right) \\
&\leq C(\log n)^{\eta} n^{1-\frac{1}{k}+\frac{1}{\log\log n}}
\end{aligned}$$

for some constants C, η with probability $1 - \frac{6\log\log\log n}{8\log(k/2)n^2} \geq 1 - \frac{1}{n}$.

To get the suboptimality, divide the value gap by $(1+\beta)n \leq 2n$.

Proof of Theorem 4.4

(a) Choose n large enough such that $k \leq \frac{n}{2}$. We define k groups of reviewers and papers such that all papers have similarity 1 with all reviewers within the same group and similarity 0 with all other reviewers. Group 1 contains all papers $p \leq \lceil \frac{n}{2} \rceil$ and all reviewers $r \leq 2\lceil \frac{n}{2} \rceil$. Each other group $2, \ldots, k$ contains 2 reviewers and 1 paper. All papers and reviewers not in any group have all similarities 0. This similarity matrix has rank k.

The oracle optimal assignment for any $\mathcal{P}_2$ will split the reviewers in each group evenly between stages, so all papers in any group can be assigned a similarity-1 reviewer in both stages. This gives a total similarity of at least $n + 2(k-1)$.

Define X as the random variable representing the number of reviewers from group 1 selected to be in $\mathcal{R}_2$. $X \sim Hyp(2n, 2\lceil n/2 \rceil, n)$, the hypergeometric distribution corresponding to the

number of successes when n items are sampled without replacement from a population of $2n$ items where $2\lceil n/2 \rceil$ of them are successes. By Lemma 2.1 of [18], $\mathbb{P}[X = t] \leq \frac{C}{\sigma}$ for any $t \geq 0$ where $\sigma^2 = \frac{\lceil n/2 \rceil}{2}\left(1 - \frac{\lceil n/2 \rceil}{n}\right)$ and C is an absolute constant. Since $\sigma^2 \geq \frac{n}{4}\left(1 - \frac{1}{2} - \frac{1}{n}\right) \geq \frac{n}{16}$ for $n \geq 4$, $\mathbb{P}[X = t] \leq \frac{4C}{\sqrt{n}}$ for sufficiently large n. Therefore, $\mathbb{P}\left[\frac{n}{2} - \frac{\sqrt{n}}{16C} + 1 \leq X \leq \frac{n}{2} + \frac{\sqrt{n}}{16C}\right] \leq \frac{1}{2}$. With probability at least $\frac{1}{2}$, at least $\frac{\sqrt{n}}{16C}$ of the reviewers in group 1 cannot be matched to an optimal paper in their stage. Therefore, the total expected similarity is no greater than $n - \frac{\sqrt{n}}{32C} + 2(k-1)$ and the expected difference in value from the oracle optimal assignment is at least $\frac{\sqrt{n}}{32C}$. Since there are $2n$ assignments, the suboptimality is at least $\frac{1}{64C\sqrt{n}}$.

(b) We construct a similarity matrix by creating a vector in $\mathbb{R}^k$ for each reviewer and each paper, and setting the similarity between that reviewer and that paper to be the inner product of their corresponding vectors. Consider the cube in $\mathbb{R}^k$ contained in $[0, 1/\sqrt{k}]^k$. We construct a grid of points within this cube by evenly spacing $z = \lceil n^{1/k} \rceil$ along each axis and filling in the remaining points so that there are $z^k \geq n$ grid points in total. Place the n paper vectors at arbitrary (unique) points on this grid, so that each vector is at least $\frac{1}{\sqrt{k}\lceil n^{1/k} \rceil} \geq \frac{1}{2\sqrt{k}n^{1/k}}$ away from any other paper vector. Place the $2n$ reviewer vectors such that 2 are at each grid point with a paper vector. The inner product of any two vectors is in $[0, 1]$, so this is a valid similarity matrix. The $2n \times k$ and $n \times k$ matrices where the rows are the reviewer and paper vectors respectively have linearly independent columns and so have rank k; thus, the similarity matrix has rank k.

We claim that the oracle optimal matching across both stages chooses one reviewer from each grid point and matches it to the paper at the same point. Suppose we have a matching where this is not the case. There must exist a cycle of matched reviewer and paper pairs where the corresponding vectors are not paired with themselves and are instead paired $(x_1, x_2), (x_2, x_3), \ldots, (x_K, x_1)$. This cycle has a total similarity of (using x_{K+1} to refer to x_1)

$$\begin{aligned}\sum_{i=1}^{K} \langle x_i, x_{i+1} \rangle &\leq \sum_{i=1}^{K} ||x_i||_2 ||x_{i+1}||_2 \\ &\leq \sum_{i=1}^{K} ||x_i||_2^2 \\ &= \sum_{i=1}^{K} \langle x_i, x_i \rangle\end{aligned}$$

so the matching value can be improved by changing the cycle so that reviewers and papers at the same grid point are matched. The second inequality is because $2ab \leq a^2 + b^2$ for any $a, b \in \mathbb{R}$. Therefore, the claimed matching is indeed optimal.

Now, consider the sample of n reviewers in stage one produced by a random split of reviewers. The following lemma shows that with probability $1 - O(e^{-n/10})$, $\Theta(n)$ grid points have both reviewers present in stage one under random split.

Lemma 4.1. *There exists n_0 and a constant ζ such that for all $n \geq n_0$, the probability that less than $n/100$ grid points have both reviewers in stage one after a random reviewer split is at most $\zeta e^{-n/10}$.*

Proof. There are at most $\binom{n}{a}3^{n-a}$ ways to assign reviewers to stages such that a pairs of reviewers at the same grid point are in stage one. For all n and a such that $n+1 \geq 4a$, $\binom{n}{a}3^{n-a} = \binom{n}{a-1}\frac{n+1-a}{a}3^{n-a} \geq \binom{n}{a-1}3^{n-a+1}$. Setting $a = n/100$, $\binom{n}{n/100} \leq (100e)^{n/100} \leq \exp(0.06n)$ and $3^{n-(n/100)} \leq \exp(1.09n)$. Therefore, the number of ways to assign reviewers to stages such that less than $n/100$ pairs are in stage one is at most $\sum_{b=0}^{(n/100)-1}\binom{n}{b}3^{n-b} \leq (n/100)\binom{n}{n/100}3^{n-(n/100)} \leq \exp(1.15n + \ln(0.01n))$. Using Sterling inequalities [126], the total number of ways to assign reviewers to stages is $\binom{2n}{n} \geq \frac{2\sqrt{\pi}}{e^2\sqrt{n}}2^{2n} \geq \frac{2\sqrt{\pi}}{e^2}\exp(1.35n - 0.5\ln(n))$. Therefore, the probability that less than $n/100$ grid points have both reviewers in stage one is at most $\frac{e^2}{2\sqrt{\pi}}\exp(-0.2n + \ln(0.01n) + 0.5\ln(n)) \leq \frac{e^2}{2\sqrt{\pi}}\exp(-0.1n)\exp(-0.1n + \ln(0.01n) + 0.5\ln(n)) \leq \frac{e^2}{2\sqrt{\pi}}\exp(-0.1n)$ for sufficiently large n. □

Therefore, with high probability, at least $n/100$ reviewers must be assigned to a paper at a different grid point.

Consider the assignments produced in each stage after random split, and consider the reviewers not assigned to their optimal papers by these assignments. From the set of vectors corresponding to these suboptimally-assigned reviewers, we can construct some number K of disjoint cycles $C_j = \{x_1^{(j)}, \ldots, x_{K_j}^{(j)}\}$, where a reviewer with vector $x_i^{(j)}$ is assigned to the paper with vector $x_{i+1}^{(j)}$ when the optimal assignment would assign them to the paper with vector $x_i^{(j)}$. By Lemma 4.1, $\sum_{j=1}^{K}|C_j| \geq \frac{n}{100}$. The difference in value between the random-split assignments and the optimal assignment is

$$\begin{aligned}\sum_{j=1}^{K}\sum_{i=1}^{K_j}\langle x_i^{(j)}, x_i^{(j)} - x_{i+1}^{(j)}\rangle &= \frac{1}{2}\sum_{j=1}^{K}\sum_{i=1}^{K_j}||x_i^{(j)} - x_{i+1}^{(j)}||_2^2 \\ &\geq \frac{1}{8k}n^{-2/k}\sum_{j=1}^{K}|C_j| \\ &\geq \frac{1}{8k}n^{-2/k}\left(\frac{n}{100}\right)\end{aligned}$$

with probability at least $1 - \zeta e^{-n/10}$ for sufficiently large n. Dividing by $2n$, the suboptimality is at least $\frac{1}{1600k}n^{-2/k}$.

Proof of Theorem 4.5

In this section, we state and prove a more general version of the bound in Theorem 4.5 that does not require that $\beta\mu$ be integral. This result immediately implies the result of Theorem 4.5.

In the proof, we use the following lemma. We prove this lemma following the proof of the main theorem. For some set $\mathcal{N}$ and some constant $p \in [0,1]$, define distribution $\mathcal{I}_p(\mathcal{N})$ as the distribution over all subsets $A \subseteq \mathcal{N}$ induced by choosing each item $x \in \mathcal{N}$ to be in A independently with probability p. Recall from Section 4.6.3 the definition of Q', a modified version of Q that allows papers to be underloaded (i.e., assigned fewer reviewers than their load).

Lemma 4.2. *Consider the modified version of f: $f'(\mathcal{R}_2) = \mathbb{E}_{\mathcal{P}_2 \sim \mathcal{I}_\beta(\mathcal{P})}[Q'(\mathcal{R}_2, \mathcal{P}_2)]$. f' draws*

$\mathcal{P}_2 \sim \mathcal{I}_\beta(\mathcal{P})$ rather than $\mathcal{P}_2 \sim \mathcal{U}_{\beta n}(\mathcal{P})$ and allows papers to be underloaded. Then,

$$\mathbb{E}_{\mathcal{R}_2 \sim \mathcal{I}_{\beta/(1+\beta)}(\mathcal{R})}\left[f'(\mathcal{R}_2)\right] \leq \mathbb{E}_{\mathcal{R}_2 \sim \mathcal{U}_{(\beta/(1+\beta))m}(\mathcal{R})}[f(\mathcal{R}_2)].$$

This lemma shows that when attempting to lower bound the expected value of random split, we can analyze as if the second-stage reviewers and papers were drawn independently.

We now state and prove the main theorem. We abuse notation slightly by defining $\mathcal{M}(\mathcal{R}, \mathcal{P}; \ell_r, \ell_p)$ to include all assignments where papers are assigned either $\lfloor \ell_p \rfloor$ or $\lceil \ell_p \rceil$ reviewers when ℓ_p is not integral; i.e., $Z \in \mathcal{M}(\mathcal{R}', \mathcal{P}'; \ell_r, \ell_p)$ if and only if $\lfloor \ell_p \rfloor \leq \sum_{r \in \mathcal{R}'} Z_{r,p} \leq \lceil \ell_p \rceil$ for all $p \in \mathcal{P}'$, $\sum_{p \in \mathcal{P}'} Z_{r,p} \leq \ell_r$ for all $r \in \mathcal{R}'$, and $Z_{r,p} = 0$ for all $(r,p) \notin \mathcal{R}' \times \mathcal{P}'$.

Theorem 4.5 (Generalized)**.** *Consider any $\mu \in [10^4]$ and $\beta \in \left\{\frac{1}{100}, \ldots, \frac{100}{100}\right\}$. Let $\epsilon = \lceil \beta\mu \rceil - \lfloor \beta\mu \rfloor$. If there exists an assignment $Z^{(\mu)} \in \mathcal{M}(\mathcal{R}, \mathcal{P}; \mu, (1+\beta)\mu)$ with mean similarity $s^{(\mu)}$, choosing $\mathcal{R}_2$ via random split gives that*

$$\mathbb{E}_{\mathcal{R}_2}\left[f(\mathcal{R}_2)\right] \geq$$
$$s^{(\mu)}\left[1 - \sqrt{\frac{\beta}{2\pi(1+\beta)\lfloor(1+\beta)\mu\rfloor}}\left(2\sqrt{\frac{1}{1+\beta}} + \sqrt{1-\beta}\right) - \frac{(1+2\beta)}{(1+\beta)\lceil\beta\mu\rceil}\epsilon\right]\left[1 - \frac{\epsilon}{\lceil(1+\beta)\mu\rceil}\right].$$

Proof. By Lemma 4.2, we can consider drawing $\mathcal{P}_2 \sim \mathcal{I}_\beta(\mathcal{P})$ and $\mathcal{R}_2 \sim \mathcal{I}_{\beta/(1+\beta)}(\mathcal{R})$ and allowing papers to be underloaded. For all reviewers $r \in \mathcal{R}$, define the random variables $U_r = \begin{cases} 1 \text{ w.p. } 1/(1+\beta) \\ 2 \text{ w.p. } \beta/(1+\beta) \end{cases}$ representing the stage that reviewer r is randomly chosen to be in. Define the random variables $V_p = \begin{cases} 1 \text{ w.p. } \beta \\ 0 \text{ w.p. } 1-\beta \end{cases}$ representing whether $p \in \mathcal{P}_2$. All of these random variables are independent. Also, denote by $v^{(\mu)} = s^{(\mu)}(1+\beta)n\mu$ the total similarity value of assignment $Z^{(\mu)}$, and denote by $v_p^{(\mu)}$ and $v_r^{(\mu)}$ the total similarity value of the assignments for paper p and reviewer r respectively in assignment $Z^{(\mu)}$.

The proof works as follows. We form an assignment $B^{(1)}$ in stage one with paper loads of at most μ and reviewer loads of at most μ, and form an assignment $B^{(2)}$ in stage two with paper loads of at most $\lceil\beta\mu\rceil$ and reviewer loads of at most $\lceil\beta\mu\rceil$. We do this by initially assigning all reviewer-paper pairs from $Z^{(\mu)}$ that are present in the same stage, and then randomly removing assignments from each paper or reviewer that is overloaded. We then find "final assignments" (i.e., assignments that are feasible solutions for the two-stage assignment problem) from within $B^{(1)}$ and $B^{(2)}$.

Stage One: First, consider stage one. Define $Binom(N,p)$ as the binomial distribution with N trials and p probability of success; denote by f the Binomial pmf. The number of reviewers assigned by $Z^{(\mu)}$ to paper p and present in stage one is a $Binom\left(\lambda_p, \frac{1}{1+\beta}\right)$ variable, where $\lambda_p \in \{\lfloor(1+\beta)\mu\rfloor, \lceil(1+\beta)\mu\rceil\}$. Suppose that we observe the set of such reviewers, randomly remove reviewers from this set until its size is at most μ, and then assign these reviewers to p in our stage one assignment $B^{(1)}$. Since each reviewer has at most μ assigned papers in $Z^{(\mu)}$, $B^{(1)}$ satisfies the desired load constraints on both sides. The expected total value of the assigned

reviewers after we drop reviewers from each paper at random is

$$\begin{aligned}\mathbb{E}\left[\sum_{r\in\mathcal{R}} B^{(1)}_{r,p} S_{r,p}\right] &= \sum_{x=0}^{\mu} f\left(x;\lambda_p,\frac{1}{1+\beta}\right) v_p^{(\mu)} \frac{x}{\lambda_p} + \sum_{x=\mu+1}^{\lambda_p} f\left(x;\lambda_p,\frac{1}{1+\beta}\right) v_p^{(\mu)} \frac{\mu}{\lambda_p} \\ &= \frac{v_p^{(\mu)}}{\lambda_p} \mathbb{E}_{X\sim Binom\left(\lambda_p,\frac{1}{1+\beta}\right)}\left[\min(X,\mu)\right] \\ &\geq \frac{v_p^{(\mu)}}{\lceil(1+\beta)\mu\rceil} \mathbb{E}_{X\sim Binom\left(\lfloor(1+\beta)\mu\rfloor,\frac{1}{1+\beta}\right)}\left[\min(X,\mu)\right].\end{aligned}$$

Summing over all papers,

$$\mathbb{E}\left[\sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}} B^{(1)}_{r,p} S_{r,p}\right] \geq \frac{v^{(\mu)}}{\lceil(1+\beta)\mu\rceil} \mathbb{E}_{X\sim Binom\left(\lfloor(1+\beta)\mu\rfloor,\frac{1}{1+\beta}\right)}\left[\min(X,\mu)\right].$$

Due to the loads, the matrix $\frac{1}{\mu}B^{(1)}$ has row sums at most 1 and column sums at most 1. By a generalization of the Birkhoff-von Neumann theorem [25], this can be written as a convex combination of matrices with all entries in $\{0,1\}$, all row sums at most 1, and all column sums at most 1. Each of these matrices represents an assignment obeying the reviewer and paper load constraints for the final assignment (since we allow papers to be underloaded), so they are all valid final assignments in stage one. At least one of these assignments must have a total value at least $\frac{1}{\mu}$ of the value of $B^{(1)}$.

Stage Two: Now, consider stage two. The number of reviewers assigned by $Z^{(\mu)}$ to paper p present in stage two is a $Binom\left(\lambda_p, \frac{\beta}{1+\beta}\right)$ variable, where $\lambda_p \in \{\lfloor(1+\beta)\mu\rfloor, \lceil(1+\beta)\mu\rceil\}$. The number of papers assigned by $Z^{(\mu)}$ to a reviewer r present in stage two is a $Binom(\mu,\beta)$ random variable. We first calculate the total expected value of all assignments in $Z^{(\mu)}$ and present in stage two (without dropping assignments from overloaded reviewers/papers):

$$\mathbb{E}\left[\sum_{r\in\mathcal{R}_2, p\in\mathcal{P}_2} Z^{(\mu)}_{r,p} S_{r,p}\right] = \frac{\beta^2}{1+\beta} v^{(\mu)}.$$

We then construct assignment $B^{(2a)}$ from the pairs assigned in $Z^{(\mu)}$ and present in stage two by dropping reviewers from each paper at random until all papers have a load of at most $\lceil\beta\mu\rceil$, with a value on paper p (if present in stage two) of

$$\begin{aligned}\mathbb{E}\left[\sum_{r\in\mathcal{R}} B^{(2a)}_{r,p} S_{r,p} \middle| p\in\mathcal{P}_2\right] &= \sum_{x=0}^{\lceil\beta\mu\rceil} f\left(x;\lambda_p,\frac{\beta}{1+\beta}\right) v_p^{(\mu)} \frac{x}{\lambda_p} + \sum_{x=\lceil\beta\mu\rceil+1}^{\lambda_p} f\left(x;\lambda_p,\frac{\beta}{1+\beta}\right) v_p^{(\mu)} \frac{\lceil\beta\mu\rceil}{\lambda_p} \\ &= \frac{v_p^{(\mu)}}{\lambda_p} \mathbb{E}_{X\sim Binom\left(\lambda_p,\frac{\beta}{1+\beta}\right)}\left[\min\left(X,\lceil\beta\mu\rceil\right)\right] \\ &\geq \frac{v_p^{(\mu)}}{\lceil(1+\beta)\mu\rceil} \mathbb{E}_{X\sim Binom\left(\lfloor(1+\beta)\mu\rfloor,\frac{\beta}{1+\beta}\right)}\left[\min\left(X,\lceil\beta\mu\rceil\right)\right].\end{aligned}$$

Each paper is present in stage two with probability β, so the overall value is

$$\mathbb{E}\left[\sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}} B^{(2a)}_{r,p} S_{r,p}\right] \geq \frac{\beta v^{(\mu)}}{\lceil(1+\beta)\mu\rceil}\mathbb{E}_{X\sim Binom\left(\lfloor(1+\beta)\mu\rfloor,\frac{\beta}{1+\beta}\right)}\left[\min\left(X,\lceil\beta\mu\rceil\right)\right].$$

We separately construct assignment $B^{(2b)}$ from the pairs assigned in $Z^{(\mu)}$ and present in stage two by dropping papers from each reviewer at random until all reviewers have a load of at most $\lceil\beta\mu\rceil$, with a value on reviewer r (if present in stage two) of

$$\begin{aligned}\mathbb{E}\left[\sum_{p\in\mathcal{P}} B^{(2b)}_{r,p} S_{r,p}\middle| r\in\mathcal{R}_2\right] &= \sum_{x=0}^{\lceil\beta\mu\rceil} f\left(x;\mu,\beta\right) v^{(\mu)}_r \frac{x}{\mu} + \sum_{x=\lceil\beta\mu\rceil+1}^{\mu} f\left(x;\mu,\beta\right) v^{(\mu)}_r \frac{\lceil\beta\mu\rceil}{\mu}\\ &= \frac{v^{(\mu)}_r}{\mu}\mathbb{E}_{X\sim Binom(\mu,\beta)}\left[\min\left(X,\lceil\beta\mu\rceil\right)\right].\end{aligned}$$

Totalling across all reviewers, since each reviewer is present in stage two with probability $\frac{\beta}{1+\beta}$,

$$\mathbb{E}\left[\sum_{r\in\mathcal{R}}\sum_{p\in\mathcal{P}} B^{(2b)}_{r,p} S_{r,p}\right] = \frac{\beta v^{(\mu)}}{(1+\beta)\mu}\mathbb{E}_{X\sim Binom(\mu,\beta)}\left[\min\left(X,\lceil\beta\mu\rceil\right)\right].$$

Define $B^{(2)}$ as the intersection of the assigned pairs in $B^{(2a)}$ and $B^{(2b)}$; $B^{(2)}$ satisfies the desired load constraints on both sides. Its expected value is lower-bounded by the total expected value of $B^{(2a)}$ and $B^{(2b)}$ less the expected value of the pairs assigned in $Z^{(\mu)}$ and present in stage two, since the pairs assigned in $B^{(2a)}$ and $B^{(2b)}$ are subsets of the stage two pairs assigned in $Z^{(\mu)}$.

$$\begin{aligned}\mathbb{E}\left[\sum_{r\in\mathcal{R},p\in\mathcal{P}} B^{(2)}_{r,p} S_{r,p}\right] \geq{}& \frac{\beta v^{(\mu)}}{\lceil(1+\beta)\mu\rceil}\mathbb{E}_{X\sim Binom\left(\lfloor(1+\beta)\mu\rfloor,\frac{\beta}{1+\beta}\right)}\left[\min\left(X,\lceil\beta\mu\rceil\right)\right]\\ &+ \frac{\beta v^{(\mu)}}{(1+\beta)\mu}\mathbb{E}_{X\sim Binom(\mu,\beta)}\left[\min\left(X,\lceil\beta\mu\rceil\right)\right].\\ &- \frac{\beta^2}{1+\beta} v^{(\mu)}.\end{aligned}$$

By the same Birkhoff-von Neumann argument as used in stage one, there exists a valid final assignment in stage two with paper loads of at most 1, reviewer loads of at most 1, and value at least $\frac{1}{\lceil\beta\mu\rceil}$ of the value of $B^{(2)}$.

Total: Sum the total value of the 1-load assignment in both stages and divide by $(1+\beta)n$ to get a lower bound on the expected mean similarity:

$$\begin{aligned}
&s^{(\mu)}\Bigg[\frac{\mu}{\lceil(1+\beta)\mu\rceil}\mathbb{E}_{X\sim Binom\left(\lfloor(1+\beta)\mu\rfloor,\frac{1}{1+\beta}\right)}\left[\min\left(\frac{X}{\mu},1\right)\right] \\
&\quad+\frac{\beta\mu}{\lceil(1+\beta)\mu\rceil}\mathbb{E}_{X\sim Binom\left(\lfloor(1+\beta)\mu\rfloor,\frac{\beta}{1+\beta}\right)}\left[\min\left(\frac{X}{\lceil\beta\mu\rceil},1\right)\right] \\
&\quad+\frac{\beta}{(1+\beta)}\mathbb{E}_{X\sim Binom(\mu,\beta)}\left[\min\left(\frac{X}{\lceil\beta\mu\rceil},1\right)\right]-\frac{\beta^2\mu}{(1+\beta)\lceil\beta\mu\rceil}\Bigg] \\
&\geq s^{(\mu)}\left(\frac{\mu}{\lceil(1+\beta)\mu\rceil}\right)\Bigg[\mathbb{E}_{X\sim Binom\left(\lfloor(1+\beta)\mu\rfloor,\frac{1}{1+\beta}\right)}\left[\min\left(\frac{X}{\mu},1\right)\right] \\
&\quad+\beta\Bigg(\mathbb{E}_{X\sim Binom\left(\lfloor(1+\beta)\mu\rfloor,\frac{\beta}{1+\beta}\right)}\left[\min\left(\frac{X}{\lceil\beta\mu\rceil},1\right)\right] \\
&\qquad\qquad+\mathbb{E}_{X\sim Binom(\mu,\beta)}\left[\min\left(\frac{X}{\lceil\beta\mu\rceil},1\right)\right]-1\Bigg)\Bigg]. \qquad (4.2)
\end{aligned}$$

Since the above bound is a function of the binomial pmf, we search for a simpler approximation. Say that $X\sim Binom(N,p)$ and $q=1-p$. The above bound is a function of $\mathbb{E}\left[\min\left(\frac{X}{\ell},1\right)\right]$ for three binomial random variables where $Np\leq\ell$. We approximate these binomials as if they were normals $G\sim\mathcal{N}(Np,Npq)$, since $\frac{X-Np}{\sqrt{Npq}}$ converges in distribution to a standard normal. We use f_G as the pdf of G, F_G as the cdf of G, and Φ as the standard normal cdf.

$$\begin{aligned}
\mathbb{E}\left[\min\left(\frac{X}{\ell},1\right)\right]&\approx\mathbb{E}\left[\min\left(\frac{G}{\ell},1\right)\right] \\
&=\mathbb{E}\left[\min\left(\frac{G}{\ell},1\right)|G\leq\ell\right]\mathbb{P}[G\leq\ell]+\mathbb{E}\left[\min\left(\frac{G}{\ell},1\right)|G>\ell\right]\mathbb{P}[G>\ell] \\
&=\frac{1}{\ell}\left(Np-Npq\frac{f_G(\ell)}{F_G(\ell)}\right)F_G(\ell)+1-F_G(\ell) \\
&=1-\frac{Npq}{\ell}f_G(\ell)-F_G(\ell)\left(1-\frac{Np}{\ell}\right) \\
&\geq 1-\sqrt{\frac{q}{2\pi Np}}-\Phi\left(\frac{\ell-Np}{\sqrt{Npq}}\right)\left(1-\frac{Np}{\ell}\right) \\
&\geq 1-\sqrt{\frac{q}{2\pi Np}}-\left(1-\frac{Np}{\ell}\right).
\end{aligned}$$

In total, defining $\epsilon^+=\lceil\beta\mu\rceil-\beta\mu$, $\epsilon^-=\beta\mu-\lfloor\beta\mu\rfloor$, and $\epsilon=\epsilon^++\epsilon^-$, this approximation to the lower bound gives

$$s^{(\mu)}\left(\frac{\mu}{\lceil(1+\beta)\mu\rceil}\right)\left[1-\sqrt{\frac{\beta}{2\pi\lfloor(1+\beta)\mu\rfloor}}-\Phi\left(\frac{\mu-\frac{\lfloor(1+\beta)\mu\rfloor}{1+\beta}}{\sqrt{\frac{\lfloor(1+\beta)\mu\rfloor\beta}{(1+\beta)^2}}}\right)\left(1-\frac{\lfloor(1+\beta)\mu\rfloor}{(1+\beta)\mu}\right)\right.$$

$$+\beta\left(1-\sqrt{\frac{1}{2\pi\lfloor(1+\beta)\mu\rfloor\beta}}-\sqrt{\frac{1-\beta}{2\pi\mu\beta}}\right.$$
$$\left.-\Phi\left(\frac{\lceil\beta\mu\rceil-\frac{\lfloor(1+\beta)\mu\rfloor\beta}{1+\beta}}{\sqrt{\frac{\lfloor(1+\beta)\mu\rfloor\beta}{(1+\beta)^2}}}\right)\left(1-\frac{\beta\lfloor(1+\beta)\mu\rfloor}{\lceil\beta\mu\rceil(1+\beta)}\right)-\Phi\left(\frac{\lceil\beta\mu\rceil-\beta\mu}{\sqrt{(1-\beta)\beta\mu}}\right)\left(1-\frac{\beta\mu}{\lceil\beta\mu\rceil}\right)\right)\Bigg] \tag{4.3}$$

$$\geq s^{(\mu)}\left[1-\frac{2}{(1+\beta)}\sqrt{\frac{\beta}{2\pi\lfloor(1+\beta)\mu\rfloor}}-\frac{1}{(1+\beta)}\sqrt{\frac{\beta(1-\beta)}{2\pi\mu}}\right.$$
$$\left.-\frac{1+2\beta}{1+\beta}\left(1-\frac{\beta\lfloor(1+\beta)\mu\rfloor}{\lceil\beta\mu\rceil(1+\beta)}\right)\right]\left(\frac{(1+\beta)\mu}{\lceil(1+\beta)\mu\rceil}\right)$$
$$\geq s^{(\mu)}\left[1-\sqrt{\frac{\beta}{2\pi(1+\beta)\lfloor(1+\beta)\mu\rfloor}}\left(2\sqrt{\frac{1}{1+\beta}}+\sqrt{1-\beta}\right)\right.$$
$$\left.-\frac{(1+2\beta)}{(1+\beta)\lceil\beta\mu\rceil}\left(\frac{\beta}{1+\beta}\epsilon^{-}+\epsilon^{+}\right)\right]\left(1-\frac{\epsilon^{+}}{\lceil(1+\beta)\mu\rceil}\right)$$
$$\geq s^{(\mu)}\left[1-\sqrt{\frac{\beta}{2\pi(1+\beta)\lfloor(1+\beta)\mu\rfloor}}\left(2\sqrt{\frac{1}{1+\beta}}+\sqrt{1-\beta}\right)\right.$$
$$\left.-\frac{(1+2\beta)}{(1+\beta)\lceil\beta\mu\rceil}\epsilon\right]\left(1-\frac{\epsilon}{\lceil(1+\beta)\mu\rceil}\right).$$

Via simulation, we confirm that the approximation in (4.3) is in fact a lower bound on the expression in (4.2) for all $\mu \in [10^4]$ and $\beta \in \left\{\frac{1}{100}, \ldots, \frac{100}{100}\right\}$. In Figure 4.2 of Section 4.5, we plot the more precise bound of (4.3). □

Proof of Lemma 4.2

It remains to prove Lemma 4.2. We first prove a supplementary lemma, from which the main lemma follows.

Lemma 4.3. *Consider a set $\mathcal{N}$ of N items and a submodular function $g : 2^{\mathcal{N}} \to \mathbb{R}$. Then, $E_{A\sim\mathcal{I}_p(\mathcal{N})}[g(A)] \leq E_{A\sim\mathcal{U}_{pN}(\mathcal{N})}[g(A)]$.*

Proof. Consider the following randomized procedure h, which takes in a set $D \subseteq \mathcal{N}$ and constructs a set containing exactly pN items. If $|D| = pN$, return $h(D) = D$. If $x = |D| - pN > 0$, then choose a subset $B \subseteq D$ uniformly at random such that $|B| = x$ and return $h(D) = D \setminus B$. If $x = pN - |D| > 0$, choose a subset $C \subseteq \mathcal{N} \setminus D$ uniformly at random such that $|C| = x$ and return $h(D) = D \cup C$.

If $D \sim \mathcal{I}_p(\mathcal{N})$ then $h(D) \sim \mathcal{U}_{pN}(\mathcal{N})$, since all subsets of size pN have an equal chance to be created. We will show that $\mathbb{E}_{D\sim\mathcal{I}_p(\mathcal{N})}[g(h(D)) - g(D)] \geq 0$, proving that $\mathbb{E}_{D\sim\mathcal{I}_p(\mathcal{N})}[g(D)] \leq \mathbb{E}_{A\sim\mathcal{U}_{pN}(\mathcal{N})}[g(A)]$. More specifically, we show that for each $x > 0$,

$$\mathbb{E}_{D\sim\mathcal{I}_p(\mathcal{N})}[g(h(D)) - g(D) \mid |D| = pN + x] + \mathbb{E}_{D\sim\mathcal{I}_p(\mathcal{N})}[g(h(D)) - g(D) \mid |D| = pN - x] \geq 0.$$

Since g is submodular, for any subsets $A \subseteq \mathcal{N}$, $C \subseteq A$, $B \subseteq \mathcal{N} \setminus A$, we have that $g((A \setminus C) \cup B) - g(A \setminus C) \geq g(A \cup B) - g(A)$.

$$
\begin{aligned}
&\mathbb{E}_{D \sim \mathcal{I}_p(\mathcal{N})}[g(h(D)) - g(D) \mid |D| = pN + x] \\
&\quad + \mathbb{E}_{D \sim \mathcal{I}_p(\mathcal{N})}[g(h(D)) - g(D) \mid |D| = pN - x] \\
&= \frac{1}{\binom{N}{pN+x}\binom{pN+x}{x}} \sum_{D \subseteq \mathcal{N}: |D| = pN+x} \sum_{B \subseteq D: |B| = x} g(D \setminus B) - g(D) \\
&\quad + \frac{1}{\binom{N}{pN-x}\binom{N-pN+x}{x}} \sum_{D \subseteq \mathcal{N}: |D| = pN-x} \sum_{C \subseteq \mathcal{N} \setminus D: |C| = x} g(D \cup C) - g(D) \qquad (4.4) \\
&= \frac{1}{\binom{N}{pN}\binom{N-pN}{x}} \sum_{A \subseteq \mathcal{N}: |A| = pN} \sum_{B \subseteq \mathcal{N} \setminus A: |B| = x} g(A) - g(A \cup B) \\
&\quad + \frac{1}{\binom{N}{pN}\binom{pN}{x}} \sum_{A \subseteq \mathcal{N}: |A| = pN} \sum_{C \subseteq A: |C| = x} g(A) - g(A \setminus C) \qquad (4.5) \\
&= \frac{1}{\binom{N}{pN}\binom{pN}{x}\binom{N-pN}{x}} \\
&\quad \sum_{A \subseteq \mathcal{N}: |A| = pN} \sum_{B \subseteq \mathcal{N} \setminus A: |B| = x} \sum_{C \subseteq A: |C| = x} g(A) - g(A \cup B) + g(A) - g(A \setminus C) \\
&= \frac{1}{\binom{N}{pN}\binom{pN}{x}\binom{N-pN}{x}} \\
&\quad \sum_{A \subseteq \mathcal{N}: |A| = pN} \sum_{B \subseteq \mathcal{N} \setminus A: |B| = x} \sum_{C \subseteq A: |C| = x} g(A) - g(A \cup B) + g((A \setminus C) \cup B) - g(A \setminus C) \qquad (4.6) \\
&\geq 0.
\end{aligned}
$$

(4.4) writes out the expected value as a sum over all choices of D and all sets sampled by the procedure h. (4.5) rewrites the sums using $A = D \setminus B$ and $A = D \cup C$; each choice of D, B in the original sum corresponds to exactly one choice of A, B in the new sum. (4.6) re-arranges the sum to exchange each $g(A)$ term for a $g((A \setminus C) \cup B)$ term; in both cases each set of size pN is counted $\binom{pN}{x}\binom{N-pN}{x}$ times in the sum (exactly once for each choice of B, C). □

We also show that f' and Q' are submodular.

Proposition 4.2. *$Q'(\mathcal{R}_2, \mathcal{P}_2)$ is submodular in $\mathcal{R}_2$ and $\mathcal{P}_2$. Further, f' is submodular in $\mathcal{R}_2$.*

Proof. Note that $\max_{Z \in \mathcal{M}'(\mathcal{R}', \mathcal{P}'; 1, 1)} \sum_{r \in \mathcal{R}', p \in \mathcal{P}'} Z_{r,p} S_{r,p}$ is a submodular function of the reviewer set $\mathcal{R}'$ when the paper set $\mathcal{P}'$ is held fixed and of the paper set $\mathcal{P}'$ when the reviewer set is held fixed [89]. Submodularity in $\mathcal{R}_2$ is equivalent to submodularity in $\mathcal{R}_1 = \mathcal{R} \setminus \mathcal{R}_2$, so $Q'(\mathcal{R}_2, \mathcal{P}_2)$ is submodular in $\mathcal{R}_2$ and $\mathcal{P}_2$. As a sum over terms submodular in $\mathcal{R}_2$, f' is submodular in $\mathcal{R}_2$. □

We now prove the main lemma. Since $S \geq 0$, there exists a maximum-similarity assignment from within $\mathcal{M}'(\mathcal{R}', \mathcal{P}'; 1, 1)$ that meets all paper load constraints with equality when $|\mathcal{R}'| \geq |\mathcal{P}'|$, and thus is contained in $\mathcal{M}(\mathcal{R}', \mathcal{P}'; 1, 1)$. Also, $\mathcal{M}(\mathcal{R}', \mathcal{P}'; 1, 1) \subseteq \mathcal{M}'(\mathcal{R}', \mathcal{P}'; 1, 1)$. Thus, when $|\mathcal{R}_2| \geq \beta n$ and $m - |\mathcal{R}_2| \geq n$, $Q(\mathcal{R}_2, \mathcal{P}_2) = Q'(\mathcal{R}_2, \mathcal{P}_2)$. Further, by Proposition 4.2, Q'

is submodular in $\mathcal{P}_2$. Therefore, by Lemma 4.3, $f(\mathcal{R}_2) \geq f'(\mathcal{R}_2)$ whenever $|\mathcal{R}_2| = \frac{\beta}{1+\beta}m$ (since $m \geq (1+\beta)n$). This shows that

$$\mathbb{E}_{\mathcal{R}_2 \sim \mathcal{U}_{(\beta/(1+\beta))m}(\mathcal{R})}[f(\mathcal{R}_2)] \geq \mathbb{E}_{\mathcal{R}_2 \sim \mathcal{U}_{(\beta/(1+\beta))m}(\mathcal{R})}[f'(\mathcal{R}_2)].$$

By Proposition 4.2, f' is submodular in $\mathcal{R}_2$. Therefore, by Lemma 4.3,

$$\mathbb{E}_{\mathcal{R}_2 \sim \mathcal{U}_{(\beta/(1+\beta))m}(\mathcal{R})}[f'(\mathcal{R}_2)] \geq \mathbb{E}_{\mathcal{R}_2 \sim \mathcal{I}_{\beta/(1+\beta)}(\mathcal{R})}\left[f'(\mathcal{R}_2)\right].$$

Proof of Theorem 4.6

In this section, we state and prove a more general version of the bound in Theorem 4.6 that does not require that $\frac{\mu}{4}$ be integral. This result immediately implies the result of Theorem 4.6.

Theorem 4.6 (Generalized)**.** *Suppose $\beta = 1$, and consider any $\mu \in [10^4]$. Define $\epsilon = \lceil \frac{\mu}{4} \rceil - \frac{\mu}{4}$. Suppose there exists an assignment $Z^{(1)} \in \mathcal{M}(\mathcal{R}, \mathcal{P}; 1, 2)$ with mean similarity $s^{(1)}$. Suppose there also exists an assignment $Z^{(\mu)} \in \mathcal{M}(\mathcal{R}, \mathcal{P}; \mu, 2\mu)$ with mean similarity $s^{(\mu)}$ that does not contain any of the pairs assigned in $Z^{(1)}$. Then, choosing $\mathcal{R}_2$ via random split gives that*

$$\mathbb{E}_{\mathcal{R}_2}\left[f(\mathcal{R}_2)\right] \geq \frac{3}{4}s^{(1)} + \frac{s^{(\mu)}}{4}\left[1 - \frac{\sqrt{7}+\sqrt{6}}{2\sqrt{\pi\mu}} - \frac{3\epsilon}{\lceil \mu/4 \rceil}\right].$$

Proof. We attempt to construct an assignment in each stage in two rounds. We first match all available pairs from $Z^{(1)}$ (tiebreaking randomly between the two reviewers if both are available), and then attempt to construct a larger assignment from $Z^{(\mu)}$.

By Lemma 4.2, we can consider drawing $\mathcal{P}_2 \sim \mathcal{I}_\beta(\mathcal{P})$ and $\mathcal{R}_2 \sim \mathcal{I}_{\beta/(1+\beta)}(\mathcal{R})$ and allowing papers to be underloaded. For all reviewers $r \in \mathcal{R}$, define the random variables $U_r = \begin{cases} 1 \text{ w.p. } 1/2 \\ 2 \text{ w.p. } 1/2 \end{cases}$ representing the stage that reviewer r is randomly chosen to be in. For each pair of reviewers (i, j) that are matched to the same paper in $Z^{(1)}$, define the random variables $F_{i,j} = \begin{cases} i \text{ w.p. } 1/2 \\ j \text{ w.p. } 1/2 \end{cases}$ representing the reviewer that will be assigned in round one if both are in the same stage. All of these random variables are independent. Define the total similarity value of the assignments as $v^{(1)} = 2ns^{(1)}$ and $v^{(\mu)} = 2n\mu s^{(\mu)}$. For $Z^{(\mu)}$, define the total similarity value assigned to paper p and reviewer r respectively as $v_p^{(\mu)}$ and $v_r^{(\mu)}$.

Round One: We first match all available pairs from $Z^{(1)}$. For any paper $p \in \mathcal{P}$, call a, b the two reviewers assigned to p by $Z^{(1)}$. The value assigned to paper p across both stages is represented by a random variable $V_p = \mathbb{I}[U_a \neq U_b](S_{a,p}+S_{b,p})+\mathbb{I}[U_a = U_b](S_{a,p}\mathbb{I}[F_{a,b} = a]+S_{b,p}\mathbb{I}[F_{a,b} = b])$. $\mathbb{E}[V_p] = \frac{3}{4}(S_{a,p} + S_{b,p})$, so $\mathbb{E}[\sum_{p\in\mathcal{P}} V_p] = \frac{3}{4}v^{(1)}$ is the total expected value assigned in round 1.

Round Two: Fixing the round one assignments, we now attempt to find a matching for all remaining papers and reviewers by matching pairs from within $Z^{(\mu)}$. We first attempt to find an assignment with paper and reviewer loads of at most $\theta = \lceil \mu/4 \rceil$ among the remaining reviewers and papers in each stage. We start with the pairs from $Z^{(\mu)}$ that both are present in this stage and were not matched in round one, and randomly drop entries from each reviewer and paper until they are no longer overloaded. This argument mirrors the one made in the proof of Theorem 4.5.

We consider stage one without loss of generality. We start by constructing an assignment C to include all pairs assigned in $Z^{(\mu)}$ where the reviewer and paper both were unmatched in round one and are in stage one. Each reviewer-paper pair in $Z^{(\mu)}$ can be assigned in C with probability $\frac{1}{32}$, so $\mathbb{E}\left[\sum_{r\in\mathcal{R},p\in\mathcal{P}} C_{r,p}S_{r,p}\right] = \frac{v^{(\mu)}}{32}$.

We then construct an assignment $B^{(1a)}$ from C by removing assigned reviewers from each paper at random until each paper has load at most θ. Fix some paper p, and define W_p as the event that paper p was not assigned in round one. The number of reviewers assigned to p in $Z^{(\mu)}$ that are in stage one and not assigned in round one is a $Binom(2\mu, 1/8)$ random variable. The expected value assigned to p in this assignment is (using f as the Binomial pmf),

$$\begin{aligned}\mathbb{E}\left[\sum_{r\in\mathcal{R}} B^{(1a)}_{r,p}S_{r,p}\middle| W_p\right] &= \sum_{x=0}^{\theta} f(x;2\mu,1/8)v_p^{(\mu)}\frac{x}{2\mu} + \sum_{x=\theta+1}^{2\mu} f(x;2\mu,1/8)v_p^{(\mu)}\frac{\theta}{2\mu}\\ &= \frac{v_p^{(\mu)}}{2\mu}\mathbb{E}_{X\sim Binom(2\mu,1/8)}\left[\min(X,\theta)\right].\end{aligned}$$

Summing over all papers, since each paper has a $1/4$ change of being unmatched in round one,

$$\mathbb{E}\left[\sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}} B^{(1a)}_{r,p}S_{r,p}\right] = \frac{v^{(\mu)}}{8\mu}\mathbb{E}_{X\sim Binom(2\mu,1/8)}\left[\min(X,\theta)\right].$$

We separately construct an assignment $B^{(1b)}$ from C by removing assigned papers from each reviewer at random until each reviewer has load at most θ. Fix some reviewer r, and define W_r as the event that reviewer r was not assigned in round one. The number of papers assigned to r in $Z^{(\mu)}$ that are not assigned in round one is a $Binom(\mu, 1/4)$ random variable. The expected value assigned to r in this assignment is,

$$\begin{aligned}\mathbb{E}\left[\sum_{p\in\mathcal{P}} B^{(1b)}_{r,p}S_{r,p}\middle| W_r\right] &= \sum_{x=0}^{\theta} f(x;\mu,1/4)v_r^{(\mu)}\frac{x}{\mu} + \sum_{x=\theta+1}^{\mu} f(x;\mu,1/4)v_r^{(\mu)}\frac{\theta}{\mu}\\ &= \frac{v_r^{(\mu)}}{\mu}\mathbb{E}_{X\sim Binom(\mu,1/4)}\left[\min(X,\theta)\right].\end{aligned}$$

Summing over all reviewers, since each reviewer has a $1/8$ change of being both unmatched in round one and present in stage one,

$$\mathbb{E}\left[\sum_{r\in\mathcal{R}}\sum_{p\in\mathcal{P}} B^{(1b)}_{r,p}S_{r,p}\right] = \frac{v^{(\mu)}}{8\mu}\mathbb{E}_{X\sim Binom(\mu,1/4)}\left[\min(X,\theta)\right].$$

Finally, we construct $B^{(1)}$ to include all pairs assigned in both $B^{(1a)}$ and $B^{(1b)}$. It has value at least equal to the total value of $B^{(1a)}$ and $B^{(1b)}$ less the value of C, since the assigned pairs in $B^{(1a)}$ and $B^{(1b)}$ are subsets of the assigned pairs in C.

$$\begin{aligned}&\mathbb{E}\left[\sum_{r\in\mathcal{R},p\in\mathcal{P}} B^{(1)}_{r,p}S_{r,p}\right]\\ &\quad\geq \frac{v^{(\mu)}}{8\mu}\left[\mathbb{E}_{X\sim Binom(2\mu,1/8)}\left[\min(X,\theta)\right] + \mathbb{E}_{X\sim Binom(\mu,1/4)}\left[\min(X,\theta)\right] - \frac{\mu}{4}\right].\end{aligned}$$

By construction this assignment has paper loads of at most θ and reviewer loads of at most θ (among all reviewers and papers unmatched in round one and present in stage one).

By a generalization of the Birkhoff-von Neumann theorem [25], there exists an assignment with paper loads of at most 1 and reviewer loads of at most 1 among all reviewers and papers unmatched in round one and present in stage one, with value at least $\frac{1}{\theta}$ of the value of $B^{(1)}$. Totalling over both stages and dividing by $2n$, the round two assignments contribute at least

$$\frac{s^{(\mu)}}{4}\left[\mathbb{E}_{X\sim Binom(2\mu,1/8)}\left[\min\left(\frac{X}{\theta},1\right)\right]+\mathbb{E}_{X\sim Binom(\mu,1/4)}\left[\min\left(\frac{X}{\theta},1\right)\right]-\frac{\mu}{4\theta}\right] \tag{4.7}$$

to the mean assignment value.

If $X \sim Binom(N,p)$, the above bound is a function of $\mathbb{E}\left[\min\left(\frac{X}{\ell},1\right)\right]$ for two binomial random variables where $Np \leq \ell$. Using the normal approximation presented in the proof of Theorem 4.5, we get the following approximation to the above bound (defining $\epsilon = \lceil \mu/4 \rceil - (\mu/4)$):

$$\begin{aligned}
&\frac{s^{(\mu)}}{4}\left[1-\frac{\sqrt{7}+\sqrt{6}}{2\sqrt{\pi\mu}}-\left(1-\frac{\mu/4}{\lceil\mu/4\rceil}\right)\left(\Phi\left(\frac{\epsilon}{\sqrt{\frac{7}{32}\mu}}\right)+\Phi\left(\frac{\epsilon}{\sqrt{\frac{3}{16}\mu}}\right)+1\right)\right] \qquad (4.8)\\
&\geq \frac{s^{(\mu)}}{4}\left[1-\frac{\sqrt{7}+\sqrt{6}}{2\sqrt{\pi\mu}}-3\left(1-\frac{\mu/4}{\lceil\mu/4\rceil}\right)\right]\\
&= \frac{s^{(\mu)}}{4}\left[1-\frac{\sqrt{7}+\sqrt{6}}{2\sqrt{\pi\mu}}-\frac{3\epsilon}{\lceil\mu/4\rceil}\right].
\end{aligned}$$

Via simulation, we confirm that the approximation in (4.8) is in fact a lower bound on the expression in (4.7) for all $\mu \in [10^4]$. In Figure 4.2 of Section 4.5, we plot the more precise bound of (4.8). □

4.8 Discussion

We showed that randomly splitting reviewers between two reviewing phases or two reviewing conditions produces near-optimal assignments on realistic conference similarity matrices. Our analysis of this phenomenon can help future program chairs make decisions about whether random split will work well for their conference's two-phase review process, based on their assessment of whether a few simple conditions are applicable to their case. In the setting of conference experiment design, our analysis allows program chairs to understand if running an experiment on their review process will significantly impact their assignment quality.

In addition, our results can potentially be further generalized to related reviewing models such as those of academic journals (which accept submissions on a rolling basis), or to other multi-stage resource allocation problems that involve matching resources based on similarities. For example, datacenters receiving a large batch of jobs may have to select some to run on various servers immediately and some to run later when additional servers have been freed, or hospitals may want to assign nurses to shifts based on expertise but without knowledge of which expertise will be most applicable in later shifts.

One limitation of our work is that while our empirical results demonstrate the effectiveness of the random-split strategy with real conference data, our theoretical results make the simplifying assumption that paper and reviewer loads are 1, which is unrealistic for real conferences. However, we believe that incorporating this detail would not change our explanations for the good performance of random split. Another limitation is that we assume the set of papers requiring reviews in the second stage is drawn uniformly at random. Although this is a reasonable belief without further information in the two-phase setting, one direction for future work is to consider non-uniform distributions of second-stage papers and analyze if a form of random split still performs well there.

Our work could potentially produce negative outcomes in the form of worse paper assignments if program chairs decide to use random split on an incorrect belief that their conference will fit our conditions. However, program chairs are required to make such decisions about how to perform the paper assignment anyway, so this is not a significant increase in risk. The use of random reviewer splits, as opposed to some alternate strategy where reviewers can self-select their stage, could also negatively impact reviewers with strong preferences over which stage they review in (e.g., due to schedule constraints). These preferences should ideally be taken into account along with the similarity of the resulting assignment when choosing the reviewer split; we leave this as an interesting direction for future work.

Chapter 5

Individually Strategyproof Paper Assignments

In this chapter, we address a form of undesirable behavior in conference peer review introduced in Chapter 1 as "strategic reviewing". Conference peer review is competitive, meaning that the eventual outcome of a submitted paper is impacted by the evaluations of other papers. Only a fixed fraction of the papers are accepted as posters, accepted for oral presentations, or given "best paper" awards. As a result, malicious reviewers may behave strategically in the following way: a reviewer may give low scores to the papers they evaluate, in the hope that by hurting the chances of those papers, they increase the relative chance of a good outcome for their own paper.

A controlled experiment [14] found that people indeed behave in such a strategic manner in competitive peer assessment. Furthermore, the work [147] shows that even a small fraction of reviewers behaving strategically in peer review can significantly lower the average quality of the accepted papers. It is thus vital to ensure the fairness and integrity of the process by developing mechanisms to prevent such strategic behavior. In fact, the National Science Foundation briefly experimented with a method (introduced by [109]) that attempts to prevent strategic behavior in the peer review of research proposals [112], but this method does not come with theoretical guarantees.

Henceforth, we use the term "strategyproofness" to refer to individual strategyproofness, meaning that no single reviewer can improve the outcome of their own paper by providing untruthful reviews. By far the most well-studied way of ensuring strategyproofness is the partitioning method introduced in [5] and studied further in [12, 13, 22, 51, 65, 77, 107, 160]. Under the partitioning method, papers are partitioned into some number of subsets, and no reviewer is assigned a paper from the same subset as their own. The individual reviewer evaluations are then aggregated separately for each subset, so that any reviewer's evaluations cannot influence the final outcome for their own paper.

Apart from strategyproofness, another key aspect in assigning reviewers to papers is matching based on expertise. For instance, in peer review of papers or proposals, not all reviewers have expertise for all papers or proposals. In the standard assignment algorithm for conference peer review (Chapter 1), this expertise is represented by similarities. Since the goal of peer assessment is to evaluate each paper as competently as possible, it is important to ensure that each paper is assigned reviewers with suitable expertise, or in other words, to maximize the quality of the

assignment of reviewers to papers.

As both strategyproofness and assignment quality are crucial, *our work studies the problem of finding a strategyproof assignment with maximum assignment quality*. The key question we ask is: *what is the price paid by strategyproofing in terms of the assigned reviewers' expertise?* As a metric of evaluation, we use the ratio of the quality we obtain with strategyproofness to the maximum quality achievable without the strategyproofness constraint.

Our work contributes to the body of literature on analyzing the price of strategyproofness in various settings [10, 41, 77, 88, 125]. This includes a line of work on impartial peer nomination/selection [5, 12, 13, 22, 51, 65, 91, 107], which focuses on selecting the best k papers in a strategyproof manner given an profile of evaluations. In contrast, we optimize the *assignment* of reviewers to papers subject to a strategyproofness constraint and characterize the price of strategyproofness in terms of the assigned reviewers' expertise. Further, our setting generalizes the standard peer selection setting, since evaluations may be used for various relative grading schemes other than best-k selection. The prior work closest to ours is [160], which considers the partitioning mechanism specifically for conference peer review. They provide an algorithm that utilizes partitioning and conduct empirical analysis on its quality. However, they provide no theoretical guarantees on their algorithm's assignment quality.

Conference peer review is only one example of a peer assessment setting, where the individuals making papers are each asked to evaluate papers made by their peers. Peer assessment can occur in a variety of disciplines when the number of papers is large enough to make independent expert evaluations of all of them infeasible. In education, peer grading of homeworks has become increasingly prevalent in Massive Open Online Courses (MOOCs) [39, 120, 135] and conventional classrooms. In the workplace, peer evaluation is frequently used to assess employee performance and determine employee promotions and bonuses [50, 156]. In scientific research, peer review is used for grant proposals (in addition to conference paper papers) [134, 136, 148]. As in the case of conference peer review, these settings are also competitive and may require evaluators to have suitable expertise for their assigned submissions. In peer assessment within an organization, the peer assessors for any employee must be chosen to have a suitable understanding of that employee's work. In peer grading of essays or projects, the assessors must have the relevant background to do a suitable evaluation. While we remain focused on the conference peer review setting for clarity, the analysis in this chapter applies similarly to these other settings.

With that background, we now list **our main contributions** in this chapter:

1. We present polynomial-time computable algorithms that are optimal in the worst case.
2. We show that the problem of instance-wise optimal strategyproof assignment via partitioning is NP-hard.
3. We conduct experimental evaluations on data from the peer-review process of the 2018 International Conference on Learning Representations (ICLR), where we find that our algorithms achieve high-expertise assignments while producing fair partitions of papers.

This chapter is based on the joint work [36]. The results in Sections 5.2.1-5.2.3 and Section 5.2.6 were established by my collaborators prior to my contributions, and are included here purely to provide context.

All of the code for our algorithms and our empirical results is freely available online at `https://github.com/sjecmen/optimal_strategyproof_assignment`.

5.1 Background and Problem Formulation

We consider a setting of peer assessment between reviewers, where each reviewer first submits some work for evaluation and is then assigned to evaluate other reviewers' papers. After evaluations have been completed, papers can be compared based on the evaluation scores in order to determine any competitive outcomes, such as relative grades (in a classroom setting), accept/reject decisions (in conference peer review), or employee bonuses and promotions (in an organization).

5.1.1 Preliminaries

Let $\mathcal{R} = \{r_1, \ldots, r_m\}$ be the set of reviewers and let $\mathcal{P} = \{p_1, \ldots, p_n\}$ be the set of submitted papers from the reviewers. We assume that each reviewer r_i ($i \in [m]$) authors exactly one paper p_i. (This is equivalent to common settings in the strategyproofing literature [12, 22, 51, 65, 77]. Furthermore, we handle arbitrary authorships in Section 5.4.)

A key focus of our work is the assignment of reviewers to papers for review. Constructing a high-quality assignment for peer assessment (in the absence of strategyproofing requirements) is a well-studied problem, and is conducted in two phases. The first phase involves computing a "similarity" between every reviewer-paper pair, a number between 0 and 1 where a higher value indicates a better match in terms of expertise. Similarities are computed in various ways [29, 49, 108, 110]. Our work is agnostic to the method used to compute similarity scores. We assume we are given a matrix $S \in [0, 1]^{m \times m}$ of 'similarity scores' for each reviewer-paper pair that capture the expertise of each reviewer to evaluate each paper. For any $i \in [m], j \in [n]$, the $(i, j)^{\text{th}}$ entry of matrix S, denoted by $S_{i,j}$, represents the similarity between reviewer r_i and paper p_j, where a higher value means that one expects a better quality of evaluation.

5.1.2 Assignments

The second phase of the assignment process then uses the similarities to assign papers to reviewers. For a predefined value $k \in \mathbb{Z}_+$, an assignment with loads of k is defined as a set $\mathcal{Z} \subseteq \mathcal{R} \times \mathcal{P}$ of assigned reviewer-paper pairs where each paper is assigned exactly k reviewers, each reviewer is assigned to exactly k papers, and no reviewer is assigned to their own paper. It is important to note that in our applications of interest, the "load" k is typically a small constant independent of m, and we will assume so throughout this chapter.

The assignment is chosen by maximizing a specified objective subject to the load constraints. By far the most common choice of objective is to maximize the sum of the assigned similarities [29, 30, 60, 97, 144, 145], and this approach is widely used in practice. Formally, for any assignment $\mathcal{Z} \subseteq \mathcal{R} \times \mathcal{P}$, the total similarity is given by $\sum_{(r_i, p_j) \in \mathcal{Z}} S_{i,j}$. Fixing some k, define

$\mathcal{Z}_S^*$ as the maximum-similarity assignment

$$\mathcal{Z}_S^* = \arg\max_{\mathcal{Z}\subseteq\mathcal{R}\times\mathcal{P}} \sum_{(r_i,p_j)\in\mathcal{Z}} S_{i,j} \tag{5.1a}$$

$$\text{subject to } \sum_{r_i\in\mathcal{R}} \mathbb{I}[(r_i,p_j)\in\mathcal{Z}] = k \qquad \forall p_j \in \mathcal{P} \tag{5.1b}$$

$$\sum_{p_j\in\mathcal{P}} \mathbb{I}[(r_i,p_j)\in\mathcal{Z}] = k \qquad \forall r_i \in \mathcal{R} \tag{5.1c}$$

$$(r_i,p_i)\notin\mathcal{Z} \qquad \forall r_i \in \mathcal{R}. \tag{5.1d}$$

The optimal assignment (without strategyproofness) $\mathcal{Z}_S^*$ can be found efficiently via standard methods such as min-cost flow algorithms or linear programming. Let Opt_S be the similarity of $\mathcal{Z}_S^*$ (leaving dependence on k implicit in the notation); that is, Opt_S is the maximum value of the aforementioned objective under the stated constraints. When unambiguous, the subscript S may be omitted.

5.1.3 Strategyproofness via Partitioning

Our goal in this paper is to find maximum-similarity *strategyproof* assignments. A strategyproof assignment is one in which no reviewer can improve the outcome of their own paper by changing the evaluation they provide.

As introduced earlier, a standard method for constructing strategyproof assignments begins by partitioning the reviewers into two subsets. An assignment of reviewers to papers is then found, where reviewers can only be assigned to papers authored by reviewers in the other subset. After evaluations are completed, any relative grading (e.g., classroom grading or accept/reject decisions) is done independently within each subset. Thus, the evaluation provided by any reviewer cannot influence the final outcome of their own paper.

In this paper, we use the term "strategyproof-via-partitioning" specifically to describe assignments produced in this way.

Definition 5.1. *An assignment $\mathcal{Z}$ is* **strategyproof-via-partitioning** *if there exists a partition of $\mathcal{R}$ into two subsets $\mathcal{R}_1, \mathcal{R}_2$ such that*

$$(r_i,p_j)\notin\mathcal{Z} \qquad \forall r_i, r_j \in \mathcal{R}_t; \forall t \in \{1,2\} \tag{5.2a}$$

$$\mathcal{R}_1 \cup \mathcal{R}_2 = \mathcal{R}; \quad \mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset. \tag{5.2b}$$

In Section 5.2.6, we extend this definition to allow for partitioning into more than two subsets. Our goal is to find a maximum-similarity strategyproof-via-partitioning assignment

$$\arg\max_{\mathcal{R}_1,\mathcal{R}_2\subseteq\mathcal{R};\mathcal{Z}\subseteq\mathcal{R}\times\mathcal{P}} \sum_{(r_i,p_j)\in\mathcal{Z}} S_{i,j}$$

$$\text{subject to (5.1b)} - \text{(5.1d)}, \text{(5.2a)}, \text{(5.2b)}.$$

If an assignment satisfies (5.2a) for some partition, we say that assignment "respects" the partition; we say that a pair (r_i, p_j) respects the partition if r_i and r_j are in different subsets. Note that

Algorithm 5.1 Random Partition

Input: $\mathcal{R}$, $\mathcal{P}$, S, k

1: Sample $\mathcal{R}_1$ uniformly at random from $\{\mathcal{R}' : \mathcal{R}' \subseteq \mathcal{R}, |\mathcal{R}'| = |\mathcal{R}|/2\}$
2: $\mathcal{R}_2 \leftarrow \mathcal{R} \setminus \mathcal{R}_1$
3: $\mathcal{Z} \leftarrow$ max-similarity assignment with loads k respecting $(\mathcal{R}_1, \mathcal{R}_2)$
4: **return** assignment $\mathcal{Z}$ and partition $(\mathcal{R}_1, \mathcal{R}_2)$

the load constraints imply $|\mathcal{R}_1| = |\mathcal{R}_2|$ for any feasible solution, so we assume that m is even in all of our results; we also assume that $k \leq \frac{m}{2}$ for feasibility.

Given a partition $(\mathcal{R}_1, \mathcal{R}_2)$, finding the maximum-similarity assignment can be done via standard methods by additionally disallowing any pairs violating constraint (5.2a). Thus, the primary question we consider in this paper is how to optimally choose the partition in order to maximize the similarity of the resulting assignment.

5.1.4 Evaluation Metric

We evaluate a strategyproof-via-partitioning assignment algorithm in terms of the ratio between the similarity of the assignment it produces and Opt_S, the similarity of the optimal non-strategyproof assignment. Specifically, consider any assignment algorithm that, given input similarities S, produces a strategyproof-via-partitioning assignment denoted by $\mathcal{Z}_S$. We evaluate its performance in terms of the worst-case input similarities as:

$$\min_{S:\mathsf{Opt}_S>0} \frac{\sum_{(r_i,p_j)\in\mathcal{Z}_S} S_{i,j}}{\mathsf{Opt}_S}.$$

5.2 Theoretical Results

In this section, we present our main theoretical results.

5.2.1 Baseline: Random Partitioning

We begin with a result that provides a simple baseline for comparison: Algorithm 5.1 chooses a partition uniformly at random. This is the approach taken by most prior literature on partitioning-based mechanisms. It is easy to show that such a uniformly random partition can attain at least half of the optimal similarity.

Proposition 5.1. *For any k and any S, Algorithm 5.1 finds a strategyproof-via-partitioning assignment with similarity at least $\frac{1}{2}\mathsf{Opt}_S$ in expectation.*

Proof. Since it is feasible to assign all pairs in $\mathcal{Z}_S^*$ that respect the partition, Algorithm 5.1

achieves expected similarity

$$\begin{aligned}\mathbb{E}_{\mathcal{Z}}\left[\sum_{(r_i,p_j)\in\mathcal{Z}} S_{i,j}\right] &\geq \sum_{(r_i,p_j)\in\mathcal{Z}_S^*} S_{i,j}(\mathbb{P}[r_i\in\mathcal{R}_1, r_j\in\mathcal{R}_2]+\mathbb{P}[r_j\in\mathcal{R}_1, r_i\in\mathcal{R}_2])\\ &= \sum_{(r_i,p_j)\in\mathcal{Z}_S^*} S_{i,j}\left(\frac{m}{m-1}\right)\frac{1}{2}\\ &\geq \frac{1}{2}\mathsf{Opt}_S.\end{aligned}$$

□

Note that this bound on the expected performance of random partitioning is tight in the limit as m grows: in the worst-case over similarities, Algorithm 5.1 achieves exactly $\left(\frac{m}{m-1}\right)\frac{1}{2}\mathsf{Opt}$ similarity. This occurs when all reviewer-paper pairs assigned by $\mathcal{Z}^*$ have similarity 1, and all other pairs have similarity 0.

5.2.2 Worst-Case Upper Bound

Since $\frac{1}{2}\mathsf{Opt}$ is easily attainable, the next natural question is: how much better is achievable? We establish an upper bound of $\frac{k+1}{2k+1}\mathsf{Opt}$ on the worst-case performance of any strategyproof-via-partitioning assignment algorithm.

Theorem 5.1. *For any k and any m, there exist similarities S for m reviewers such that no strategyproof-via-partitioning assignment has similarity greater than $\frac{k+1}{2k+1}\mathsf{Opt}_S$.*

Proof. Place the reviewers into groups of size $2k+1$, leaving any remaining reviewers out. Within each complete group, number the reviewers from 0 to $2k$. For all i from 0 to $2k$, set the similarity of r_i and $p_{i+1},\ldots,p_{(i+1+k) \mod 2k+1}$ to 1. Set all other similarities to 0. On these similarities, $\mathcal{Z}^*$ can assign every similarity-1 pair, for a total of $k(2k+1)$ per group. The optimal partition splits each group into subsets of size k and $k+1$, allowing at most $k(k+1)$ similarity-1 pairs to be assigned in each group. □

5.2.3 Cycle-Breaking Algorithm

In this section, we present a simple algorithm that meets the upper bound of Theorem 5.1 when $k=1$.

Define a "cycle" γ of length ℓ in an assignment as an ordered list of indices $\gamma_1,\ldots,\gamma_\ell$ such that reviewer r_{γ_i} is assigned to paper $p_{\gamma_{i+1}}$ (defining $\gamma_{\ell+1}=\gamma_1$). In any assignment with loads $k=1$, the full set of indices $[m]$ can be uniquely partitioned into such cycles, since each reviewer is assigned to one paper and each paper is assigned one reviewer.

Algorithm 5.2 works by splitting each cycle in the optimal $k=1$ assignment across the partition in the way that maximizes similarity. The following theorem shows a lower bound on the similarity of the strategyproof-via-partitioning assignment produced by this algorithm when $k=1$.

Theorem 5.2. *When $k=1$, for any S, Algorithm 5.2 finds a strategyproof-via-partitioning assignment with similarity at least $\frac{2}{3}\mathsf{Opt}_S$ in polynomial time.*

Algorithm 5.2 Cycle-Breaking Algorithm

Input: reviewers $\mathcal{R}$, papers $\mathcal{P}$, similarities S, load k

$\widetilde{\mathcal{Z}}_S^* \leftarrow$ max-similarity assignment with loads 1
$\mathcal{R}_1 \leftarrow \emptyset$; $\mathcal{R}_2 \leftarrow \emptyset$
for cycle γ of length ℓ in $\widetilde{\mathcal{Z}}_S^*$ **do**
    $y \leftarrow \min_{i \in [\ell]} S_{\gamma_i, \gamma_{i+1}}$
    $A \leftarrow \emptyset$; $B \leftarrow \emptyset$
    for $i \in [\ell]$ **do**
        $j \leftarrow y + i \mod \ell$
        if i odd **then**
            $A \leftarrow A \cup \{r_{\gamma_j}\}$
        else
            $B \leftarrow B \cup \{r_{\gamma_j}\}$
        end if
    end for
    if $|\mathcal{R}_1| \leq |\mathcal{R}_2|$ **then**
        $\mathcal{R}_1 \leftarrow \mathcal{R}_1 \cup A$; $\mathcal{R}_2 \leftarrow \mathcal{R}_2 \cup B$
    else
        $\mathcal{R}_1 \leftarrow \mathcal{R}_1 \cup B$; $\mathcal{R}_2 \leftarrow \mathcal{R}_2 \cup A$
    end if
end for
$\mathcal{Z} \leftarrow$ max-similarity assignment with loads k respecting $(\mathcal{R}_1, \mathcal{R}_2)$
return assignment $\mathcal{Z}$ and partition $(\mathcal{R}_1, \mathcal{R}_2)$

Proof. $(\mathcal{R}_1, \mathcal{R}_2)$ is a partition of $\mathcal{R}$ since each reviewer is included in exactly one cycle in $\widetilde{\mathcal{Z}}_S^*$. Further, $|\mathcal{R}_1| = |\mathcal{R}_2|$ since reviewers are added to the partition to keep it as balanced as possible and we assume m is even.

We bound the value of the returned assignment $\mathcal{Z}$ when $k = 1$. By construction, at most one reviewer-paper pair in each cycle of $\widetilde{\mathcal{Z}}_S^*$ does not respect the partition. Any cycle containing such a disallowed pair must be of length at least three, and the disallowed pair must have the minimum similarity among all assigned pairs in the cycle. Since it is feasible to assign all pairs in $\widetilde{\mathcal{Z}}_S^*$ that respect the partition, the value of the strategyproof-via-partitioning assignment must be at least $\frac{2}{3}\mathsf{Opt}_S$.

The partitioning step can be done in $O(m)$ time, since each reviewer is considered once, and finding the two maximum-similarity matchings can be done with high probability in $\widetilde{O}(m^3)$ time [153]. □

5.2.4 Coloring Algorithm

In this section, we present another algorithm for strategyproof peer assessment, which meets the upper bound of Theorem 5.1 for any k. The algorithm begins by constructing a directed graph $G_{\mathcal{Z}^*}$ representing the optimal assignment $\mathcal{Z}^*$. This graph contains one vertex v_i for all $i \in [m]$, and an edge (v_i, v_j) if $(r_i, p_j) \in \mathcal{Z}^*$. We then find an equitable coloring of this graph, which is

Algorithm 5.3 Coloring Algorithm

Input: reviewers $\mathcal{R}$, papers $\mathcal{P}$, similarities S, load k

1: $\mathcal{Z}_S^* \leftarrow$ max-similarity assignment with loads k
2: $G_{\mathcal{Z}^*} \leftarrow$ directed graph representing $\mathcal{Z}_S^*$
3: $f \leftarrow$ equitable $(2k+2)$-coloring of $G_{\mathcal{Z}^*}$
4: **for** $T \in \{T : T \subseteq [2k+2], |T| = k+1\}$ **do**
5: $\quad \mathcal{R}_T \leftarrow \{r_i : v_i \in V, f(v_i) \in T\}$
6: $\quad \mathcal{R}'_T \leftarrow \{r_i : v_i \in V, f(v_i) \notin T\}$
7: $\quad x_T \leftarrow \sum_{r_i \in \mathcal{R}_T, r_j \in \mathcal{R}'_T} S_{i,j} \mathbb{I}[(r_i, p_j) \in \mathcal{Z}_S^*]$
8: **end for**
9: $T^* = \arg\max_T x_T$
10: $\mathcal{R}_1 \leftarrow \mathcal{R}_{T^*}$; $\mathcal{R}_2 \leftarrow \mathcal{R}'_{T^*}$
11: $\mathcal{Z} \leftarrow$ max-similarity assignment with loads k respecting $(\mathcal{R}_1, \mathcal{R}_2)$
12: **return** assignment $\mathcal{Z}$ and partition $(\mathcal{R}_1, \mathcal{R}_2)$

defined as follows.

Definition 5.2. *For any $\alpha \in \mathbb{Z}_+$, an **equitable** α**-coloring** of a directed graph $G = (V, E)$ is a function $f : V \to [\alpha]$ such that $f(v_i) \neq f(v_j) \quad \forall (v_i, v_j) \in E$ and $|\{v : f(v) = x\}| - |\{v : f(v) = y\}| \leq 1 \quad \forall x, y \in [\alpha]$.*

The following well-known result shows that an equitable coloring of limited size can be found in polynomial time.

Theorem 5.3. *[62, 83] A graph $G = (V, E)$ with maximum degree at most Δ has an equitable $\Delta + 1$-coloring that can be found in $O(\Delta |V|^2)$ time.*

Algorithm 5.3 uses this result as a subroutine to find an equitable $(2k+2)$-coloring of $G_{\mathcal{Z}^*}$. It then partitions the colors in the way that maximizes the total similarity of pairs in $\mathcal{Z}^*$ split by the partition. The following result proves that this algorithm is worst-case optimal.

Theorem 5.4. *For any k and any S, if m is divisible by $2k+2$, Algorithm 5.3 finds a strategyproof-via-partitioning assignment with similarity at least $\frac{k+1}{2k+1}\mathsf{Opt}_S$ in polynomial time.*

Proof. Each vertex in $G_{\mathcal{Z}^*}$ has in-degree and out-degree k, so the maximum (total) degree is at most $2k$. Therefore, Line 3 can be implemented using Theorem 5.3 as a subroutine. Further, since m is divisible by $2k+2$, all colors have exactly $\frac{m}{2k+2}$ vertices and so $|\mathcal{R}_1| = |\mathcal{R}_2|$.

Next, we bound the value of the returned assignment $\mathcal{Z}$. Suppose we modify Line 4 to choose T uniformly at random from the set. Then, the expectation of x_T in Line 7 is $\mathbb{E}[x_T] = \sum_{(r_i,p_j) \in \mathcal{Z}_S^*} S_{i,j} \left(\frac{k+1}{2(k+1)-1}\right) = \frac{k+1}{2k+1}\mathsf{Opt}_S$. Therefore, $x_{T^*} \geq \frac{k+1}{2k+1}\mathsf{Opt}_S$. Since it is feasible to assign all pairs whose similarity is counted in x_{T^*}, the assignment $\mathcal{Z}$ has similarity at least x_{T^*}.

Assuming k is constant, the time complexity of the partitioning step is dominated by the $O(m^2)$ time taken to find the equitable coloring. Finding the two maximum-similarity matchings can be done with high probability in $\widetilde{O}(m^3)$ time [153]. □

The assumption that m is divisible by $2k+2$ is needed to guarantee that the partition is balanced. However, for arbitrary m, the subsets of the partition differ in size by only $k+1$ reviewers at most. If there are a small number of "reserve" reviewers who did not submit any

Algorithm 5.4 Multi-Partition Algorithm

Input: reviewers $\mathcal{R}$, papers $\mathcal{P}$, similarities S, load k

1: $\mathcal{Z}_S^* \leftarrow$ max-similarity assignment with loads k
2: $G_{\mathcal{Z}^*} \leftarrow$ directed graph representing $\mathcal{Z}_S^*$
3: $f \leftarrow$ equitable $(2k+1)$-coloring of $G_{\mathcal{Z}^*}$
4: **return** assignment $\mathcal{Z}_S^*$ and partition with $2k+1$ subsets $(\{r_j : v_j \in V, f(v_j) = i\}_{i \in [2k+1]})$

work and are not used in $\mathcal{Z}^*$, these reviewers can provide any evaluations needed for a feasible assignment. Since k is a small constant (often ≤ 3), having access to enough reserve reviewers is likely not an issue in practice. For example, in a scientific peer review setting, many extra non-author reviewers are available; in a classroom setting, an instructor could grade the extra papers.

5.2.5 Hardness

Although our algorithms are optimal on the worst-case input, one might hope for algorithms that can guarantee optimal performance on all inputs. However, the following result shows that when $k \geq 2$, this is NP-hard.

Theorem 5.5. *For any $k \geq 2$, it is NP-hard to find the optimal strategyproof-via-partitioning assignment, even when similarities are binary (that is, when $S \in \{0,1\}^{m \times m}$).*

Proof Sketch. The proof is by reduction from the "Simple Max Cut on Cubic Graphs" problem [162]. We construct an instance of the strategyproof-via-partitioning assignment problem where each reviewer corresponds to a vertex. For some orientation of the input graph, we set $S_{i,j} = 1$ for each directed edge (v_i, v_j), and set similarities to zero elsewhere. These edges could all be assigned by $\mathcal{Z}^*$ when $k \geq 2$, but the optimal strategyproof-via-partitioning assignment is limited to the max-cut value in the original graph. □

The complete proof is provided in Section 5.6.

5.2.6 Partitions With More Than Two Subsets

We now relax the definition of "strategyproof-via-partitioning" given in Definition 5.1. Rather than requiring that reviewers be partitioned into two subsets, we allow them to be partitioned into any constant (i.e., not depending on m) number of subsets. This slight relaxation of our problem formulation allows us to obtain a strategyproof-via-partitioning assignment that achieves total similarity Opt_S for any S.

Theorem 5.6. *For any $k \geq 1$ and any S, Algorithm 5.4 finds a partition of reviewers into $2k+1$ subsets, where each subset contains either $\lfloor \frac{m}{2k+1} \rfloor$ or $\lceil \frac{m}{2k+1} \rceil$ reviewers, and a strategyproof-via-partitioning assignment respecting this partition in polynomial time. This assignment has total similarity Opt_S.*

Proof. Each vertex in $G_{\mathcal{Z}^*}$ has in-degree and out-degree k, so the maximum (total) degree is at most $2k$. Therefore, by Theorem 5.3 we can find an equitable $(2k+1)$-coloring of $G_{\mathcal{Z}^*}$ in $O(m^2)$ time. By Definition 5.2, the entirety of $\mathcal{Z}_S^*$ respects the partition induced by the coloring and so

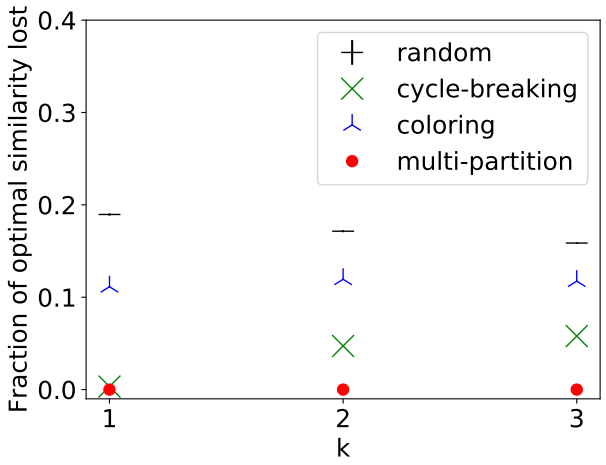


Figure 5.1: Assignment similarity lost on data from ICLR 2018.

is strategyproof-via-partitioning with respect to this partition. Also by Definition 5.2, all color classes differ in size by at most 1. □

Algorithm 5.4 constructs a directed graph representing $\mathcal{Z}^*$ as described in Section 5.2.4. It then finds an equitable $(2k+1)$-coloring using Theorem 5.3 and uses this coloring as the partition.

Although we can recover the entire optimal similarity with this method, increasing the number of subsets comes at the cost of reliability in determining the post-evaluation outcomes, since all relative outcomes must be chosen independently in each subset. In Section 5.3, we experimentally examine this cost.

5.3 Experimental Results

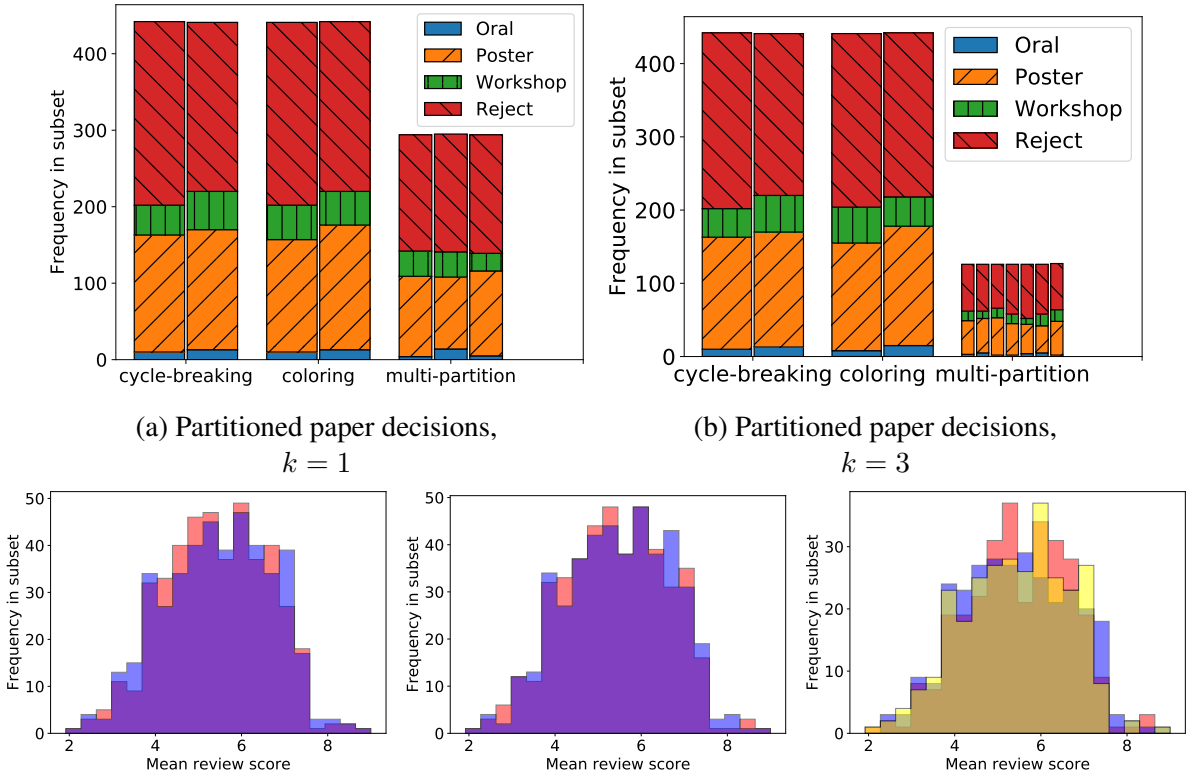
In this section, we experimentally examine the performance of algorithms for strategyproof-via-partitioning assignment.

5.3.1 Setup

We evaluate our algorithms on data from the peer-review process at the 2018 International Conference on Learning Representations (ICLR). We use similarities recreated in [160]. To evaluate the partition quality, we also use the actual review scores and the accept/reject decisions at the ICLR 2018 conference [63].

Since our algorithms require that each reviewer authors exactly one paper, we find a maximum one-to-one matching on the real authorship graph and use this as the authorship for our experiments. This resulted in matching 883 out of the 911 papers. We then discarded any reviewers and papers not included in the authorship graph. Any additional reviewers required for feasibility (due to the divisibility of m) have zero similarity with all papers.

We evaluate four partitioning algorithms: random partitioning (Algorithm 5.1), the cycle-breaking algorithm (Algorithm 5.2), the coloring algorithm (Algorithm 5.3), and the multi-partition algorithm (Algorithm 5.4). Since each paper received 3 reviews at ICLR 2018, we



(a) Partitioned paper decisions, $k = 1$

(b) Partitioned paper decisions, $k = 3$

(c) Partitioned paper scores for the cycle-breaking algorithm, $k = 1$

(d) Partitioned paper scores for the coloring algorithm, $k = 1$

(e) Partitioned paper scores for the multi-partition algorithm, $k = 1$

Figure 5.2: Partition quality on data from ICLR 2018.

test values of $k \in \{1, 2, 3\}$.

5.3.2 Assignment Similarity

We first examine the similarity of the strategyproof-via-partitioning assignments produced by each algorithm. In Figure 5.1, we report the price of strategyproofness: the difference in total similarity between the proposed algorithm's assignment and the optimal non-strategyproof assignment, as a fraction of the optimal assignment's total similarity. Results for the random partitioning algorithm are averaged over 100 trials; error bars representing standard error of the mean are too small to be visible. As expected from our theoretical results, the multi-partition algorithm achieves the full similarity of the optimal non-strategyproof assignment. On all values of k, the cycle-breaking algorithm performs very well: it loses less than 1% of the optimal similarity when $k = 1$, and furthermore, it outperforms the coloring algorithm even for higher values of k (where it does not have theoretical guarantees). The coloring algorithm loses around 12% of the optimal similarity for all values of k. The baseline of random partitioning still loses less than 20% of the optimal similarity, but is outperformed by the other algorithms. Overall, on real data our algorithms perform quite well in terms of the quality of the assignment as compared to the optimal non-strategyproof assignment.

Algorithm	k	p	D
Cycle-breaking	-	0.9007	0.0373
Coloring	1	0.8902	0.0379
	2	0.6445	0.0487
	3	0.5389	0.0530
Multi-partition	1	0.4282	0.0702
	2	0.6805	0.0742
	3	0.3457	0.1142

Table 5.1: Results of the Kolmogorov-Smirnov test of whether the review scores in the two partitioned subsets are drawn from the same distribution. p indicates the p-value and D indicates the effect size. For all algorithms and values of k, we see that the test is unable to distinguish the distributions of review scores between the partitions.

5.3.3 Partition Quality

We next examine whether the partitions produced by these algorithms place similar-quality papers into each subset, since under the partition-based method, the final accept/reject decisions for papers are performed independently in each subset. In Figures 5.2a and 5.2b, we display the number of papers receiving each decision (oral presentation, poster presentation, invitation to workshop track, or rejection) in each subset of the partitions. For each algorithm, each bar displays the decisions for the papers in one subset of the partition. Across all algorithms and values of k, the partitions constructed have very similar numbers of papers receiving each decision in each subset. Since a very small number of papers (23 out of 883) are accepted for oral presentation overall, the relative difference in the number of oral papers between subsets is sometimes large; however, the absolute difference in the number of oral papers remains small.

Further, in Figures 5.2c, 5.2d, and 5.2e, we show the mean review scores given to each paper for the case of $k = 1$. In Figures 5.2c and 5.2d, the red and blue histograms correspond to the scores given to the papers in the two subsets of the algorithm's partition, with the purple section indicating their overlap; in Figure 5.2e, the third subset is additionally indicated in yellow. For all algorithms, the distributions of scores appear very similar across subsets of the partition. Formally, we test the difference between the score distributions of different subsets via the two-sample Kolmogorov-Smirnov test, a non-parametric test of the null hypothesis that the two samples came from the same distribution. Each sample is the set of scores given to the papers in one subset of the partition. We report the results of the test in Table 5.1, which contains the p-values of the test along with the effect size D, defined as the maximum difference between the empirical cdfs of the two samples. For the multi-partition algorithm, we test each pair of subsets and we report results for the pair with highest D. In all cases, the p-values are high, meaning that the test cannot reject the hypothesis that the subsets were drawn from the same distribution.

These experiments provide evidence that the partitions created by our algorithms do not have any substantial difference in the quality of papers in each subset.

Algorithm 5.5 Heuristic Algorithm for Arbitrary Authorship

Input: reviewers $\mathcal{R}$, papers $\mathcal{P}$, similarities S, authorship graph $\mathcal{A}$, paper load k_p, maximum reviewer load k_r

1: $\overline{\mathcal{Z}}^* \leftarrow$ max-similarity assignment with loads (k_r, k_p)
2: $\{V_1, \ldots, V_N\} \leftarrow$ vertices of the connected components of $\mathcal{A}$
3: $\mathcal{R}' \leftarrow \{r'_i : i \in [N]\}$; $\mathcal{P}' \leftarrow \{p'_i : i \in [N]\}$
4: **for** $i, j \in [N]$ **do**
5: $\quad S'_{i,j} \leftarrow \sum_{r_a \in V_i, p_b \in V_j} S_{a,b} \mathbb{I}[(r_a, p_b) \in \overline{\mathcal{Z}}^*] + \sum_{r_a \in V_j, p_b \in V_i} S_{a,b} \mathbb{I}[(r_a, p_b) \in \overline{\mathcal{Z}}^*]$
6: **end for**
7: $\mathcal{Z}', (\mathcal{R}'_1, \mathcal{R}'_2) \leftarrow$ output of Algorithm 5.2 on input $(\mathcal{R}', \mathcal{P}', S', k' = 1)$
8: $\mathcal{T}_1 \leftarrow \bigcup_{i: r'_i \in \mathcal{R}'_1} V_i$; $\mathcal{T}_2 \leftarrow \bigcup_{i: r'_i \in \mathcal{R}'_2} V_i$
9: $\mathcal{Z} \leftarrow$ max-similarity assignment with loads (k_r, k_p) respecting $(\mathcal{T}_1, \mathcal{T}_2)$
10: **return** assignment $\mathcal{Z}$ and partition $(\mathcal{T}_1, \mathcal{T}_2)$

5.4 Heuristic Algorithm for Arbitrary Authorship

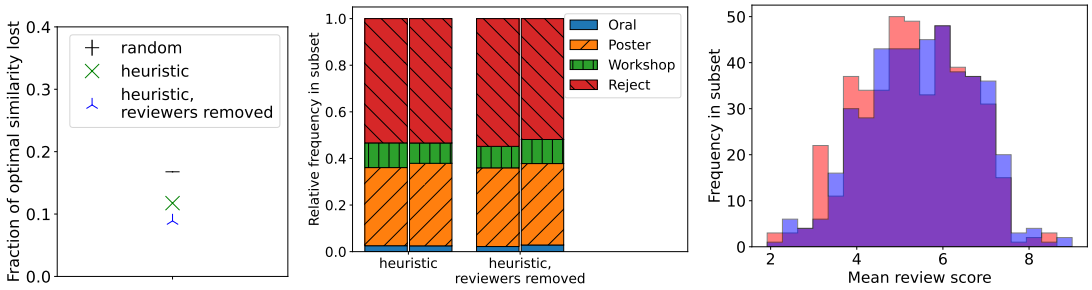
In this section, we propose an algorithm for strategyproof-via-partitioning assignment that can accommodate arbitrary authorship of papers, as opposed to the one-to-one authorship that we assume in our problem formulation (Section 5.1). This algorithm is closely based on the cycle-breaking algorithm (Algorithm 5.2) from Section 5.2.3. We do not have any theoretical guarantees for this algorithm, but we provide evaluations on the ICLR 2018 dataset introduced in Section 5.3.

5.4.1 Algorithm

Arbitrary authorship can be represented as a graph $\mathcal{A}$ where each reviewer and each paper are represented as vertices, and an edge between an reviewer and paper indicates that the reviewer authored that paper. Since authorship is not one-to-one, the number of reviewers and papers may differ and the reviewer and paper loads need not be the same. Define k_p as the paper load and k_r as the maximum reviewer load. A strategyproof-via-partitioning assignment algorithm in this setting will produce a partition of both reviewers and papers, along with an assignment that respects this partition by assigning each paper only reviewers from the other subset.

Algorithm 5.5 works by taking a problem instance with arbitrary authorship, using it to construct a (fake) problem instance with one-to-one authorship, and running Algorithm 5.2 on this fake instance to find a partition. Each reviewer in the fake instance corresponds to a connected component of the authorship graph $\mathcal{A}$. Similarities between fake reviewers are set equal to the total similarity of pairs in the optimal non-strategyproof assignment that are split between the respective components. After construction, we pass this fake instance into Algorithm 5.2.

We slightly modify Algorithm 5.2 to encourage more balanced partitions in this setting before calling it in Line 7. In Lines 14-18 of Algorithm 5.2, we take the larger of A and B and add it to the smaller of $\mathcal{R}_1$ and $\mathcal{R}_2$ as measured by the total number of papers in the connected components represented within each set. In addition, we iterate through vertices (when finding cycles) in the order of largest connected component to smallest, where size is again determined by the number of papers in each component.



(a) Assignment similarity lost

(b) Partitioned paper decisions

(c) Partitioned paper scores, heuristic algorithm with reviewers removed

Figure 5.3: Experimental results using Algorithm 5.5 on the authorship from ICLR 2018.

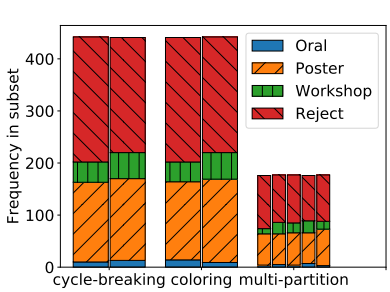
5.4.2 Experimental Results

We test Algorithm 5.5 on the ICLR 2018 dataset, using the full authorship graph from the conference. Following the suggestion in [160], we also try running Algorithm 5.5 after removing reviewers with a large number of authored papers; this breaks up large connected components in the authorship graph, thus allowing more flexibility in choosing a partition. Specifically, we remove the 53 reviewers with more than 3 papers authored (2.2% of reviewers) from the reviewer pool. As a baseline for comparison, we also test 100 trials of random partitioning, which chooses half of the connected components at random for each subset. We set loads of $k_p = 3$ and $k_r = 6$, since these are standard conference loads [160].

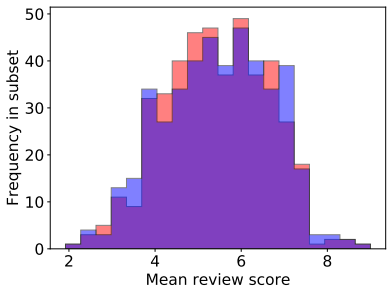
First, we see in Figure 5.3a that Algorithm 5.5 outperforms random partitioning in terms of similarity. Our algorithm loses 11.7% of the non-strategyproof optimal similarity, whereas the random partitioning loses 16.8% of optimal on average. When we remove high-authorship reviewers before running Algorithm 5.5, it only loses 8.9% of the optimal similarity (which is still allowed to use all reviewers).

Finally, we examine the partition quality in a similar manner as in Section 5.3. In Figure 5.3b, we plot the proportion of papers within each subset of the partitions produced by Algorithm 5.5 that received each decision. We see that the subsets have similar proportions of papers receiving each decision, regardless of whether we remove high-authorship reviewers. However, removing these reviewers results in a significantly more balanced partition: the number of papers differs between subsets by 109 when high-authorship reviewers are not removed and by only 1 when they are. In Figure 5.3c, we see that the two subsets also have similar distributions of paper scores when high-authorship reviewers are removed.

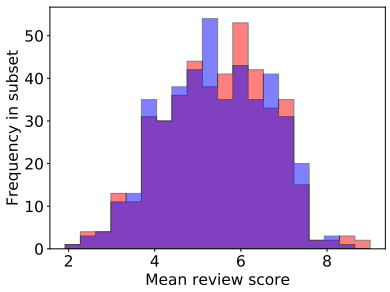
Our results are highly comparable to those of [160], who provide a partitioning algorithm that simply returns an arbitrary feasible partition of the connected components of the authorship graph. The authors report that this algorithm loses only 11.4% of the optimal similarity on the ICLR 2018 data with the same loads, a similar performance to our algorithm's despite the fact that our algorithm more carefully chooses the partition. This phenomenon may be related to the results in Chapter 4, where we found that randomly splitting reviewers into two "phases" of reviewing does not significantly degrade assignment quality on real conference datasets.



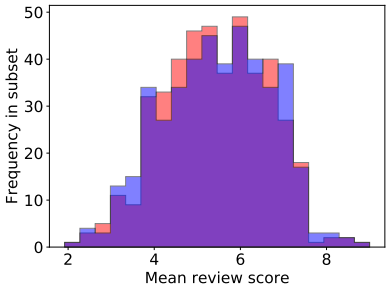
(a) Partitioned paper decisions, $k = 2$



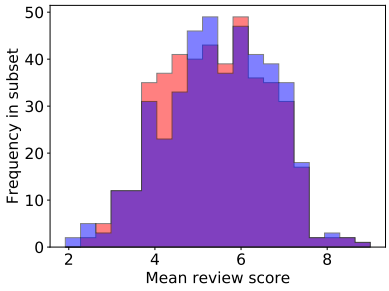
(b) Partitioned paper scores for the cycle-breaking algorithm, $k = 2$



(c) Partitioned paper scores for the coloring algorithm, $k = 2$



(d) Partitioned paper scores for the cycle-breaking algorithm, $k = 3$



(e) Partitioned paper scores for the coloring algorithm, $k = 3$

Figure 5.4: Additional experimental results on data from ICLR 2018.

5.5 Supplemental Material: Additional Experimental Results

In this section, we display additional experimental results regarding the partition quality of our algorithms on the data from ICLR 2018 (introduced in Section 5.3).

In Figure 5.4a, we display the number of papers receiving each decision in each subset of the partitions for $k = 2$, where each bar displays the decisions for the papers in one subset of the partition. The partitions constructed by all algorithms have very similar numbers of papers receiving each decision in each subset.

In Figures 5.4b-5.4e, we show the mean review scores given to each paper for the cases of $k = 2$ and $k = 3$. As before, the red and blue histograms correspond to the scores given to the papers in each subset of the algorithm's partition, with the purple section indicating their overlap. For all algorithms, the distribution of scores appear very similar across subsets of the partition. Results for the multi-partition algorithm are not shown, as there are too many subsets for the histogram to be readable.

5.6 Omitted Proofs

In this section, we present the proofs omitted from the previous sections.

Proof of Theorem 5.5

We first prove the following supplementary lemma.

Lemma 5.1. *Any graph $G = (V, E)$ with maximum degree at most 4 can be oriented in polynomial time such that the in-degree and out-degree of all vertices are at most 2.*

Proof. Consider the following procedure, which takes any arbitrary orientation of G and modifies it that it obeys the desired properties. For a vertex $v \in V$, denote the out-degree of v by $\delta_o(v)$ and the in-degree of v by $\delta_i(v)$.

On each iteration, choose v_o such that $\delta_o(v_o) \geq 3$. Consider the set $\mathcal{S}$ of all vertices reachable on a directed path from v_o. There cannot be any edges from $\mathcal{S}$ to vertices outside of $\mathcal{S}$, so $\sum_{v \in \mathcal{S}} \delta_o(v) \leq \sum_{v \in \mathcal{S}} \delta_i(v)$. Since $\sum_{v \in \mathcal{S}} \delta_i(v) + \delta_o(v) \leq 4|\mathcal{S}|$, $\sum_{v \in \mathcal{S}} \delta_o(v) \leq 2|\mathcal{S}|$. Therefore, there exists $v' \in \mathcal{S}$ such that $\delta_o(v') \leq 1$. Reversing the direction of all edges on the path from v_o to v' reduces $\delta_o(v_o)$ by 1 and increases $\delta_i(v_o)$ by 1, increases $\delta_o(v')$ by 1 and decreases $\delta_i(v')$ by 1, and does not change the in- or out-degree of any other vertices. Since $\delta_o(v') \leq 1$ and $\delta_i(v_o) \leq 1$ originally, this change does not cause any additional constraints to be violated. Therefore, this step can be repeated until all vertices satisfy $\delta_o(v) \leq 2$. Reversing the direction of all edges and repeating the entire procedure ensures that all vertices satisfy $\delta_i(v) \leq 2$ as well.

Each iteration takes $O(|V|)$ time to find a path to an appropriate v'. The number of iterations is $O(|V|)$, since each vertex is v_o at most twice. □

We now prove the main result, showing that it is unlikely that an instance-optimal algorithm for strategyproof-via-partitioning assignment exists if $k \geq 2$. We reduce from the "Simple Max Cut on Cubic Graphs" problem, which is NP-complete [162]. An instance of this problem consists of an unweighted, undirected graph $G = (V, E)$ where all vertices have degree 3 and an integer K. The question is: is there a partition of V that cuts at least K edges?

Fix any $k \geq 2$. We reduce this problem to a decision variant of our problem, defined as follows. Given reviewers $\mathcal{R}$, papers $\mathcal{P}$, similarities S, and a number x, does there exist a strategyproof-via-partitioning assignment with loads of k such that the total similarity is at least x? If it is NP-hard to determine if there is a strategyproof-via-partitioning assignment with similarity at least x, finding the strategyproof-via-partitioning assignment with maximum similarity must also be NP-hard.

Consider any instance of the max cut problem $G = (V, E)$, K. Construct a graph G' by adding $|V|$ disconnected vertices to G. By Lemma 5.1, we can find an orientation of G' in polynomial time to get a directed graph $\hat{G} = (\hat{V}, \hat{E})$ such that all out-degrees and in-degrees are at most 2. For each vertex $v_i \in V$, construct one reviewer r_i and their paper p_i. For each directed edge $(v_i, v_j) \in \hat{E}$, set similarity $S_{i,j} = 1$; set all other similarities to 0. Set $x = K$.

Suppose that V has a partition (V_1, V_2) that cuts at least K edges. Add the disconnected vertices to the subsets so that $|V_1| = |V_2| = |V|$. Partition the corresponding reviewers in the same way. Assign the reviewer r_i to paper p_j for each directed edge $(v_i, v_j) \in \hat{E}$ cut by the partition. Each reviewer has an edge to at most 2 papers and each paper has an edge to at most 2 reviewers, so these can all be assigned since $k \geq 2$. Assign the remaining reviewers and papers arbitrarily, which can be done since the partitions are balanced. This assignment has similarity at least x and is strategyproof-via-partitioning.

Suppose that V does not have a partition that cuts at least K edges. This means that there does not exist a partition of reviewers such that at least x reviewer-paper pairs with non-zero similarity can be assigned respecting the partition.

5.7 Discussion

We jointly considered two key aspects of the peer-assessment process—strategyproofing and assignment quality—and derived fundamental limits as well as designed computationally-efficient algorithms that achieve these limits. Our theoretical and empirical contributions lead to several directions of future work.

A first key direction of future work is to extend these theoretical results to arbitrary authorship graphs, as in conference peer review. We present a heuristic algorithm with an empirical evaluation in Section 5.4, but the problem of establishing fundamental limits and optimal algorithms is open. Second, most of our work considered worst-case guarantees, while showing that it is NP-hard to attain instance-wise optimality. However, our experimental results showed that our algorithms perform much better than worst-case on real-world instances. This suggests a theoretically interesting and practically useful direction of future work: designing algorithms with approximately-optimal instance-wise guarantees. Third, in contrast to past work, our partitions are non-random. Building on our experimental results revealing that these non-random partitions still result in subsets with roughly equal paper strengths, future work could dig deeper into this phenomenon both theoretically and empirically. Fourth, recent work [107] provides a strategyproof algorithm with theoretical guarantees that does not rely on partitioning. Even though partitioning is by far the dominant way of strategyproofing, it is of interest to extend our results to such strategyproofing methods that may not employ partitioning.

Chapter 6

Tradeoffs in Mitigating Manipulation of Paper Assignments

In this chapter, we consider the problem of malicious reviewers attempting to manipulate the paper assignment in order to get assigned to a target paper. This problem encompasses two of the identified forms of undesirable behavior: reviewer-author collusion and torpedo reviewing. We have previously introduced one approach to addressing the problem of reviewer-author collusion in Chapter 2: mitigating the ability of colluding reviewers to manipulate the assignment via randomization. In this chapter, rather than advocating for a specific approach, we compare a variety of conceptually different mitigation-based approaches to this issue.

The primary avenue by which malicious reviewers can manipulate the paper assignment is through paper bidding, a major part of the similarity computation. During paper bidding, each reviewer has the option of indicating how interested they are in reviewing each of the submitted papers by choosing a "bid" from a list of options (e.g., "Not willing", "In a pinch", "Willing", "Eager"). Reviewers make these decisions based on the paper title, subject areas, and abstract. Paper bidding is near-universally used in practice, and tends to have a major impact on the resulting reviewer assignment. At the 2021 AAAI Conference on Artificial Intelligence (AAAI) [96]: "*Reviewers were assigned papers for which they bid positively (willing or eager) 77.4% of the time. A back-of-the-envelope calculation leads us to estimate that 79.3% of these matches may not have happened had the reviewer not bid positively.*" Beyond bidding, malicious reviewers can also potentially modify their subject areas or their record of past work in order to achieve a desired paper assignment. However, we focus primarily on bid manipulation in this chapter as the easiest and most obvious avenue through which the paper assignment can be manipulated.

Possible manipulation of the paper assignment is taken seriously by major conferences (e.g., AAAI 2021 [96] and AAAI 2022 [134]), which have used a variety of approaches to mitigate the impact of this sort of malicious behavior in recent years. Several techniques are described in recent research papers [72, 96, 134, 159]. In this chapter, we take a high-level look at several of these approaches and consider: to what extent do they satisfy properties that we would want paper assignment algorithms to satisfy? We enumerate a list of desiderata for assignment algorithms and present a preliminary evaluation of the strengths and weaknesses of various proposed approaches on these desiderata.

6.1 Desiderata

The simplest approach to handling the problem of bid manipulation is simply to not use paper bidding at all, relying solely on text similarities and subject areas for the assignment. However, bids are near-universally used in practice and some venues even assign reviewers based only on bids. This is because there are several significant benefits to considering bids when assigning papers.

- Bids can capture aspects of a reviewer's preferences or expertise not captured by text similarities, either because the text modeling failed to accurately represent the relationship between the submission and the reviewer's past work or because relevant factors were not represented in the reviewer's past work.
- Bidding allows reviewers to correct erroneous text similarities by expressing interest in papers that are truly a good match but with which the reviewer has a low text similarity.
- Reviewers may be more likely to provide high-quality reviews for papers that they explicitly expressed interest in reviewing during bidding. This is supported by [26], which found that reviewers reported higher confidence in their reviews for papers that they bid on.

Thus, the assignment algorithms we consider here attempt to carefully use bids in order to achieve the above benefits while remaining robust against manipulation from malicious reviewers.

Based on these objectives, we present several desirable and potentially conflicting properties that an ideal assignment algorithm should satisfy.

(A) **Assignment quality:** The algorithm should produce assignments with a high level of expertise, as represented by text similarities, subject areas, and bids.

(B) **Preference expressiveness:** The algorithm should allow reviewers to express their true preferences in a flexible manner. In particular, this means that it should produce good assignments for reviewers with idiosyncratic preferences not captured by text similarities and for reviewers with erroneous text similarities.

(C) **Incentives to bid:** The algorithm should incentivize reviewers to provide accurate bids by assigning reviewers to papers that match their own bids to some extent.

(D) **Low attack success rate:** The algorithm should not allow a malicious reviewer to significantly increase their probability of assignment with a specific target paper through manipulating their bids. We call this assignment probability the "probability of successful manipulation" and call this manipulation of bids an "attack."

(E) **High attack cost:** A malicious reviewer should require extra information (e.g., other reviewers' bids/text similarities) or resources (e.g., additional colluding reviewers) in order to effectively manipulate the paper assignment.

(F) **Adjustability:** Conference program chairs should be able to easily adjust the algorithm in order to achieve a desired tradeoff between the other desiderata.

(G) **Computational scalability:** The algorithm should be feasible to run at the large scale of modern conferences (with thousands of reviewers and papers), in terms of computational resources such as runtime and memory.

Algorithm	Strengths	Weaknesses
BID LIMIT	(A), (B), (C), (G)	(D), (E)
RANDOM DISPLAY	(B), (C), (F), (G)	(A), (E)
CYCLE PREVENTION	(B), (C)	(D), (F), (G)
GEOGRAPHIC DIVERSITY	(A), (B), (C), (E)	(D), (F)
BID MODELING	(A), (D), (E)	(B), (C),(F)
REVIEWER CLUSTERING	(D), (F)	(B), (C),(E), (G)
PROBABILITY-LIMITED RANDOMIZED ASSIGNMENT	(A), (B), (C), (F), (G)	(E)

Table 6.1: Key strengths and weaknesses of algorithms.

These objectives are often contradictory and cannot all be satisfied simultaneously. We instead hope for assignment algorithms that can effectively achieve a balance between them.

6.2 Algorithms

Several different approaches have been proposed for paper assignment in the presence of malicious behavior, both in practice and in the literature. Although these approaches take a wide variety of forms, we view each of them as an end-to-end algorithm for the paper assignment process, encompassing the solicitation of bids and other features from reviewers and ending by outputting the final paper assignment. In this section, we present a brief description of some of these algorithms, along with what we see as their strengths and weaknesses on the various desiderata from Section 6.1. These strengths and weaknesses are summarized in Table 6.1.

6.2.1 Algorithm: BID LIMIT

Description: This simple approach requires each reviewer to enter at least some number of positive bids, and may also limit the number of negative bids that can be placed. If a reviewer does not meet these bidding criteria, the assignment algorithm may down-weight their bids or ignore them entirely when computing similarities. Intuitively, if a reviewer must bid positively on several papers (and these bids are weighted heavily when computing similarities), a malicious reviewer will have high similarity with some papers other than their target paper and may be assigned to those papers instead of their target. This idea has been used at numerous conferences, including AAAI 2021 and 2022.

Evaluation: On the strong side, this approach is minimally disruptive to the standard assignment process, since honest reviewers need only make additional positive bids or remove negative bids in order to meet the requirements. Thus, the approach maintains the benefits of using bids in the standard way: it finds a high-quality assignment (A), and works well for reviewers with inaccurate text similarities as they can bid positively on any papers they think are truly the best fit (B). This approach has benefits even in the absence of malicious behavior as it encourages honest reviewers to provide information (C). It also makes it more likely that each paper gets several positive bids, as [136] observes that the standard bidding process leaves many papers with very few positive bids. The algorithm requires negligible additional computation (G).

As for weaknesses, this approach is not robust against malicious behavior if malicious reviewers are behaving strategically (D), since they can choose to bid positively only on papers with which they have very low text similarity and thus are unlikely to be assigned to. Furthermore, this attack is simple to execute (E). While the parameter denoting the number of required bids is easily adjustable, the connection between this parameter and the algorithm's performance on other desiderata (e.g., the probability of successful manipulation) is unclear (F).

6.2.2 Algorithm: RANDOM DISPLAY

Description: Under this algorithm, each reviewer is shown a randomly-chosen subset of papers during the bidding process and can only bid on these papers. A similar procedure was used for bidding at AAAI 2020, where only a limited number of papers were shown to each reviewer. Since a malicious reviewer only has a limited probability of being able to bid on their target paper, this can lower the likelihood that they succeed at getting assigned. If desired, a conference can provide a hard limit on the probability of successful manipulation by disallowing the assignment of any reviewer to a paper not shown to them for bidding; we refer to this as the hard-constraint variant of RANDOM DISPLAY. In other words, if half of the papers are displayed to each reviewer under the hard-constraint variant, the probability of successful manipulation would be limited at 0.5 since the target paper is not be displayed to the malicious reviewer half of the time.

Evaluation: One strength is that the subset of papers shown to each reviewer should be representative of the conference as a whole, so an honest reviewer should not have difficulty finding good matches to bid on (B). An honest reviewer also has a strong incentive to bid since bids are used in the same way as under the standard assignment algorithm (C). Under the hard-constraint variant, the program chairs can easily achieve a desired maximum probability of successful manipulation by appropriately choosing the proportion of displayed papers (F). The algorithm requires negligible additional computation (G).

On the weak side, the optimal strategy for a malicious reviewer is simple (E): bid positively on the target paper if it is displayed and bid negatively on all others. Further, one can show that the hard-constraint variant of RANDOM DISPLAY is dominated by PROBABILITY-LIMITED RANDOMIZED ASSIGNMENT (another algorithm described later in Section 6.2.7), in terms of expected similarity (A) when they control the probability of successful manipulation at the same level. See Section 6.3 for the formal result. Note that the RANDOM DISPLAY and the PROBABILITY-LIMITED RANDOMIZED ASSIGNMENT algorithms are directly comparable because they both use the same similarity objective and provide a guarantee on the probability of successful manipulation.

Overall, the algorithm's ability to effectively limit the probability of successful manipulation (D) is unclear. Regardless of whether the hard-constraint variant is used, sufficiently limiting the probability of successful manipulation may require imposing impractical restrictions on the bidding options for honest reviewers. Furthermore, if the hard-constraint variant is not used, then a malicious reviewer may still be able to succeed even if their target paper is not displayed for bidding. By bidding negatively on all displayed papers, they may be able to lower their similarity with enough papers so that their target paper is one of the highest-similarity papers remaining (even though it was not displayed). This issue can be solved by using the hard-constraint variant, but this comes at the cost of severely restricting the assignments for honest reviewers.

6.2.3 Algorithm: CYCLE PREVENTION

Description: In some cases, malicious reviewers who have authored a paper may collude with other reviewers who have also authored a paper at the same conference. These reviewers will attempt to get assigned to each others' papers through bidding as part of a deal to benefit each other. This algorithm [19, 61] attempts to prevent this collusion by restricting the assignment so that it cannot contain any 2-cycles of reviewers: that is, if Alice is assigned to review Bob's paper, then Bob cannot be assigned to review Alice's paper. 3-cycles and larger may also be restricted if computational resources allow. This approach has been taken by AAAI 2021 [96].

Evaluation: We first consider strengths. Note that unlike most of the other algorithms we discuss, this algorithm assumes that the malicious reviewers are part of a colluding group. As mentioned in Chapter 1, there is reason to believe that collusion rings are a common form of manipulation. If so, this algorithm can provide some robustness without impacting the expressiveness of bids (B) or the incentives to bid (C).

As for weaknesses, this algorithm does not do anything to stop a malicious reviewer who is not colluding with others (D). For example, this may be a reviewer aiming to torpedo-review a rival's paper. Furthermore, this algorithm can be circumvented by groups of reviewers who decide to collude across multiple different conferences or otherwise compensate each other outside the scope of a single conference's peer review process. Program chairs cannot effectively adjust the algorithm to their needs, as even increasing the size of the removed cycles is computationally difficult (F). This computational difficulty poses a challenge for scalability (G), as finding a maximum-similarity assignment subject to cycle constraints requires solving an integer program.

The impact of this algorithm on the quality of the assignment is unclear (A). With enough expert reviewers for each topic, it's possible that most honest reviewers involved in a high-similarity cycle can be replaced with a similarly-qualified reviewer; at AAAI 2021, preventing 2-cycles lowered the total assignment similarity by only 0.01% [96]. However, the conference in question may not have a deep enough reviewer pool and this claim may not hold even if it does. Additionally, the difficulty of attacking this algorithm is dependent on the type of attacker (E). For colluding pairs of reviewers, the algorithm is not trivial to circumvent, since either an additional collaborator must be recruited or the submission venue of one of the papers must be changed; however, large colluding groups can easily set up cycles of higher length to avoid detection.

6.2.4 Algorithm: GEOGRAPHIC DIVERSITY

Description: Like the CYCLE PREVENTION algorithm, this approach focuses on defending against malicious reviewers who collude in groups. It specifically defends against groups of colluding reviewers that are based in a single geographic region by adding some form of geographic diversity constraint on the reviewer assignment. For example, AAAI 2021 used a constraint that no two reviewers assigned to the same paper belonged to the same region [96], and AAAI 2022 used a constraint that at least one assigned reviewer must be from a different region as the paper's authors. This approach is motivated by the idea that colluding groups are more likely to be from a single region, since reviewers from different areas are less likely to know each other or be able to communicate easily.

Evaluation: Large conferences include reviewers from a wide range of geographic regions, and experts in any particular topic exist in many regions. Thus, a strength is that this algorithm should not impose significant limitations on the assignments for honest reviewers. The overall assignment quality (A), expressiveness of bids (B), and incentive to bid (C) should all remain quite high, even if some expert reviewers are blocked from their optimal assignment. For example, the geographic diversity constraint imposed by AAAI 2021 lowered the assignment similarity by only 0.85% [96]. Malicious reviewers who would be stopped by this algorithm can attempt to avoid detection by recruiting colluders from a different region or by changing their location and affiliation in the conference system. However, recruiting colluders from other regions may be difficult and falsified locations can be detected by careful program chairs, making effectively circumventing this algorithm difficult (E).

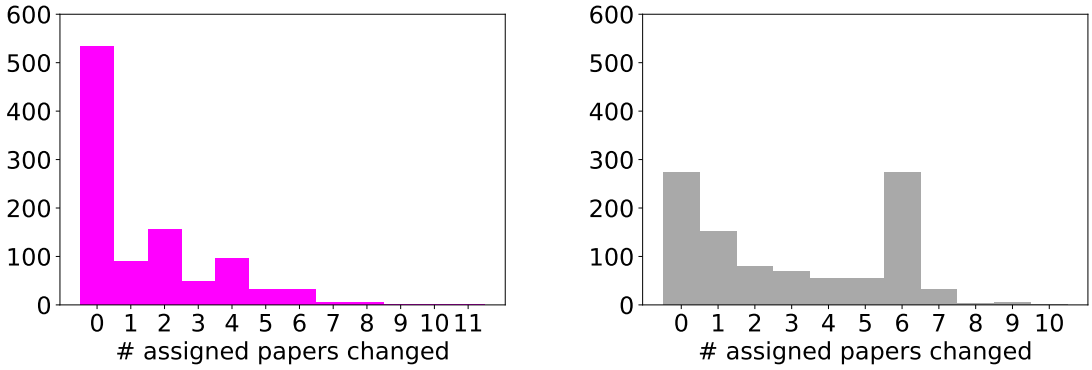
One weakness is that, like CYCLE PREVENTION, this algorithm does not defend against a malicious reviewer who is not colluding with others or against colluding reviewers who compensate each other outside of the conference's peer review process (D). It further does not defend well against colluding groups containing reviewers from several different regions, which could have formed because the reviewers previously met in some professional setting or because reviewers have moved institutions to a different region. The program chairs can choose the specific form of geographic constraint that is desired, but cannot easily see how effectively this will prevent collusion (F) since the geographical distribution of colluding groups is unknown. The computational cost of the algorithm depends on the exact form of geographic constraint posed (G).

6.2.5 Algorithm: BID MODELING

Description: This algorithm, proposed in [159], uses the submitted bids from all reviewers to train a linear regression model. This model aims to predict the bid value for each reviewer-paper pair as a function of various features of that reviewer-paper pair, including the text similarity and the subject area intersection. The paper assignment is then chosen to maximize the total *predicted* bid value of the assigned reviewers. The authors propose further techniques to defend against groups of colluding reviewers.

Evaluation: The primary strength of the BID MODELING algorithm is its robustness against malicious behavior (D): assuming that malicious reviewers cannot manipulate the reviewer-paper features, [159, Figure 1-2] demonstrates that a single malicious reviewer is unable to improve their probability of assignment to a target paper using a naive attack and has limited success with a more advanced heuristic attack. Furthermore, computing an effective attack against this model requires knowledge of the features and bids of other reviewers (E), which is unlikely to be available to malicious reviewers. The authors also find that the text similarity and bid values of the resulting assignment remain comparable to standard assignment methods (A): BID MODELING achieves a 16% increase in the average text-similarity score of the assignment over a standard assignment algorithm with the similarity function from the 2014 Conference on Neural Information Processing Systems (NeurIPS), and a 38% increase in the average bid value of the assignment over a standard assignment algorithm using only text similarities [159, Table 1].

As for weaknesses, the algorithm pays a price for this robustness in terms of its flexibility to reviewer preferences (B), as a reviewer with incorrect text similarities may find their predicted



(a) BID MODELING.

(b) Standard assignment algorithm, NeurIPS 2016 similarity function [136].

Figure 6.1: Symmetric differences between the sets of papers assigned to 1000 reviewers with honest bids and with no bids. Each reviewer is assigned at most 6 papers and each paper is assigned to 3 reviewers, within a dataset of around 2500 papers and reviewers [159].

bid values to be incorrect. The algorithm also does not allow for easy tuning by the program chairs (F), since the hyperparameters are not clearly connected to any desiderata. Additionally, if reviewer and paper features such as the subject areas and text similarities can be strategically manipulated, this approach may not be effective. Computing appropriate reviewer-paper features and fitting the model will add some additional time to the assignment algorithm at scale (G), but the algorithm does run in polynomial time.

Additionally, we conducted experiments which indicate that honest reviewers may not be sufficiently incentivized to provide bids to the algorithm (C). We sample 1000 reviewers from the dataset provided in [159] and for each compute the assignments that would result if they provide their honest bids and if they provide no bids. In Figure 6.1a, we plot the size of the symmetric difference between the set of papers assigned to this reviewer in these two cases under BID MODELING. We see that a majority of reviewers have identical assignments under BID MODELING, regardless of whether or not they provide bids; the mean number of papers changed is 1.394 and the median is 0. For comparison, we also plot in Figure 6.1b the same metric under the standard paper assignment algorithm using the NeurIPS 2016 similarity function [136]; the mean change is 2.973 and the median is 2.

6.2.6 Algorithm: REVIEWER CLUSTERING

Description: Similar to BID MODELING, this algorithm takes as input various features for each reviewer, such as their subject areas and their text similarity scores with each paper. Based on these features, it clusters reviewers into groups of some fixed size m. Papers are then assigned to each group based on the averaged bids of that group and randomly distributed among reviewers within the group. This algorithm is our attempt to capture some of the ideas behind BID MODELING in a simple manner while also providing a guarantee on the maximum probability of successful manipulation: at most $1/m$. The idea of clustering reviewers by their features

and arbitrarily distributing papers within each cluster is already used in contexts where reviewer assignment is done entirely by subject area [109].

Evaluation: On the strong side, the algorithm appears to limit much of the control that a malicious reviewer has over their assignment in the same manner as BID MODELING (D), and it also provides a parameter that can easily be tuned to adjust the tradeoff between assignment quality and probability of successful manipulation (F).

However, weaknesses of the algorithm are that it would not work well for reviewers with inaccurate text similarities (B) and that a malicious reviewer does not require knowledge of other reviewers' features in order to determine how to bid (E). Further, some honest reviewers may choose to not submit bids in the hopes that the bids of their cluster are suitable enough (C). It could also be computationally expensive to find good fixed-size clusters, since heuristic approaches may perform poorly (G). The quality of the resulting assignment depends strongly on how well the reviewer pool can be clustered into groups of similar expertise and interests, which may vary by conference (A).

6.2.7 Algorithm: PROBABILITY-LIMITED RANDOMIZED ASSIGNMENT

Description: This algorithm, proposed in [72] and detailed in Chapter 2, adds a randomized aspect to the standard assignment algorithm. Like the standard assignment algorithm, it takes bids and computes similarities as normal. Then, given a parameter $q \in [0, 1]$, it finds a randomized assignment with maximum expected similarity, subject to the constraint that the maximum probability of any reviewer-paper assignment is at most q.

Evaluation: We first consider strengths. By definition, PROBABILITY-LIMITED RANDOMIZED ASSIGNMENT finds the assignment with highest similarity among all assignments that provide a guarantee on the maximum probability of successful manipulation (A). On data from the 2018 International Conference on Learning Representations (ICLR), PROBABILITY-LIMITED RANDOMIZED ASSIGNMENT achieves 90.8% of the standard assignment algorithm's similarity with $q = 0.5$ [72, Figure 1]. Program chairs can compute this percentage for various values of q before choosing one to use in deployment, allowing them to easily control the tradeoff between the assignment similarity and the maximum probability of successful manipulation (F). Additionally, the algorithm's guarantees on the maximum probability of successful manipulation hold without any assumptions on the malicious reviewers' capabilities, so it is still effective even if aspects like the subject areas and text similarities can be manipulated. Since reviewers' bids are used without modification, the expressiveness of bids is fully preserved (B) and honest reviewers are still incentivized to bid (C). The randomized assignment can be found with the same computational resources as the standard assignment algorithm, and sampling the assignment adds little additional overhead (G).

However, one weakness of the algorithm is that it's easy for a malicious reviewer to determine their best strategy (E): bid the maximum value on their target paper and the minimum value on all others. In this manner, malicious reviewers may easily be able to achieve this theoretical maximum probability in practice, as demonstrated in simulations by [72]. Additionally, although PROBABILITY-LIMITED RANDOMIZED ASSIGNMENT is optimal in terms of similarity (subject to the constraint on the probability of successful manipulation), it remains agnostic to the computation of similarities. If some similarity components (e.g., text similarity) are believed to be more

trustworthy than the bids, this algorithm may not be able to control the probability of successful manipulation as efficiently as other algorithms that leverage this distinction (D). Although one can place greater weight on trustworthy components when computing similarities, this approach may not be the optimal way to accommodate such assumptions.

In Section 6.2.2, we mention that PROBABILITY-LIMITED RANDOMIZED ASSIGNMENT dominates the hard-constraint variant of RANDOM DISPLAY in terms of expected similarity. However, one downside of PROBABILITY-LIMITED RANDOMIZED ASSIGNMENT is that reviewers may waste time bidding on papers that they will not be assigned due to the subsequent randomization. In contrast, by doing the randomization before bidding, RANDOM DISPLAY ensures that reviewers only spend time bidding on papers for which they are eligible to be assigned.

6.3 Supplemental Material: Comparison of RANDOM DISPLAY and PROBABILITY-LIMITED RANDOMIZED ASSIGNMENT

In this section, we provide a formal comparison of the RANDOM DISPLAY and PROBABILITY-LIMITED RANDOMIZED ASSIGNMENT algorithms.

We consider the hard-constraint variant of RANDOM DISPLAY, described in Section 6.2.2, which does not allow a reviewer to be assigned to papers that were not displayed to them during bidding. Define the "display fraction" of RANDOM DISPLAY as the proportion of papers in the subset displayed to each reviewer. In this section, we compare the hard-constraint variant of RANDOM DISPLAY with display fraction q to PROBABILITY-LIMITED RANDOMIZED ASSIGNMENT (from Section 6.2.7) with probability limit q, in terms of expected similarity. These algorithms are directly comparable, since both limit the maximum probability of successful manipulation at q.

We first introduce some notation. Call n the number of reviewers and m the number of papers. Define $S \in [0,1]^{m \times n}$ as the matrix of similarities used by both algorithms, where $S_{r,p}$ is the similarity of paper p with reviewer r. S can be computed from the bids along with other features using any method, since both algorithms are agnostic to the method of similarity computation. We assume that the bids of each reviewer are the same regardless of which algorithm is used or which papers are displayed to that reviewer.

The following result shows that PROBABILITY-LIMITED RANDOMIZED ASSIGNMENT outperforms RANDOM DISPLAY in terms of expected similarity.

Theorem 6.1. *For any $q \in [0,1]$, the expected similarity of the assignment produced by* PROBABILITY-LIMITED RANDOMIZED ASSIGNMENT *with probability limit q is no less than the expected similarity of the assignment produced by the hard-constraint variant of* RANDOM DISPLAY *with display fraction q.*

Proof. Define the matrix $Q \in \{0,1\}^{m \times n}$ as the random variable representing the papers displayed to each reviewer by RANDOM DISPLAY; $Q_{r,p} = 1$ if paper p is displayed to reviewer r. Since qm of the m papers are chosen uniformly at random for each reviewer, $\mathbb{E}[Q_{r,p}] = q$. Call $Q^{(1)}, \ldots, Q^{(N)}$ the possible realizations of Q, from which Q is chosen uniformly at random. For each $i \in [N]$, define $Z^{(i)} \in \{0,1\}^{m \times n}$ as the matrix representing the assignment produced by RANDOM DISPLAY if $Q^{(i)}$ was displayed; $Z^{(i)}_{r,p} = 1$ if paper p is assigned to reviewer r.

The expected similarity of the assignment produced by RANDOM DISPLAY is

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{r=1}^{n}\sum_{p=1}^{m} Z_{r,p}^{(i)} S_{r,p}.$$

The matrix $F = \frac{1}{N}\sum_{i=1}^{N} Z^{(i)}$ satisfies $F_{r,p} \leq q$ for all entries (r, p), since

$$\frac{1}{N}\sum_{i=1}^{N} Z_{r,p}^{(i)} \leq \frac{1}{N}\sum_{i=1}^{N} Q_{r,p}^{(i)} = \mathbb{E}[Q_{r,p}] = q.$$

Consider the randomized assignment represented by F, where $F_{r,p}$ represents the marginal probability of assigning paper p to reviewer r. This randomized assignment has the same expected similarity as the assignment from RANDOM DISPLAY. Further, this is a feasible randomized assignment for PROBABILITY-LIMITED RANDOMIZED ASSIGNMENT with probability limit q, meaning that PROBABILITY-LIMITED RANDOMIZED ASSIGNMENT will return an assignment with at least this expected similarity. □

6.4 Discussion

Addressing bid manipulation in a manner that maintains the valuable properties of paper bidding is a pressing issue, given the scale and importance of modern conferences. The approaches we consider tackle the issue in a variety of ways, with different strengths and weaknesses. The least intrusive approaches (BID LIMIT, RANDOM DISPLAY, CYCLE PREVENTION, and GEOGRAPHIC DIVERSITY) keep the paper assignment process largely the same as under the standard assignment algorithm, which make them easier to deploy in practice. These algorithms preserve the essential benefits of bids but may not do enough to prevent manipulation effectively, as they have not been rigorously examined.

The other algorithms can be divided into two categories based on how they use the non-bid similarity features (e.g., text similarities). BID MODELING, along with the related REVIEWER CLUSTERING algorithm, gains significant power to stop manipulation under the assumption that these features are harder for an adversary to change. If the adversary can manipulate these features (e.g., via falsifying their TPMS profile or strategically providing subject areas [134, Section 4.2]), these algorithms may lose some effectiveness. In contrast, PROBABILITY-LIMITED RANDOMIZED ASSIGNMENT entirely abstracts away the similarity computation, ignoring any differences in the cost of manipulating different features. This algorithm thus may be most appropriate for a worst-case setting where program chairs are not willing to make assumptions about the capabilities of malicious reviewers.

CYCLE PREVENTION and GEOGRAPHIC DIVERSITY specifically focus on defending against colluding reviewers, but other approaches also can be extended to handle collusion. The formulation of the BID MODELING algorithm as proposed by [159] includes a component that effectively prevents colluding groups of a known size from manipulating the learned model. In Chapter 2, we provide an extension to our PROBABILITY-LIMITED RANDOMIZED ASSIGNMENT algorithm that additionally enforces that each paper be assigned diverse reviewers, essentially combining the PROBABILITY-LIMITED RANDOMIZED ASSIGNMENT and GEOGRAPHIC DIVERSITY approaches.

The algorithms we consider in this work sit at different positions on the tradeoffs between our proposed desiderata, but many other positions on these tradeoffs remain unfilled. We hope that our list of desiderata can help direct the development of additional algorithms to address bid manipulation. For example, we proposed the REVIEWER CLUSTERING algorithm as a simplified variant of the BID MODELING algorithm that improves on desideratum (F). Further study on the bid manipulation problem can improve on the balance between these various desired properties.

In addition, some past conferences have used multiple of these approaches at the same time. AAAI 2021 used both CYCLE PREVENTION and GEOGRAPHIC DIVERSITY, and AAAI 2022 used forms of BID LIMIT, GEOGRAPHIC DIVERSITY, and PROBABILITY-LIMITED RANDOMIZED ASSIGNMENT. A useful direction of future work is to develop new algorithms that combine multiple previous approaches in order to simultaneously achieve their benefits.

Finally, our analysis indirectly compares algorithms based on whether they satisfy our desiderata. One might hope to additionally conduct some form of direct comparison between algorithms, e.g., by comparing the assignment quality of each algorithm at a given probability of successful manipulation. However, there are numerous challenges in making such a comparison. Different algorithms make different assumptions about adversary capabilities and may optimize different objectives, such that both "probability of successful manipulation" and "assignment quality" may be incomparable between algorithms. Furthermore, non-malicious reviewers may behave differently under different algorithms (e.g., by providing more bids under BID LIMIT than under another algorithm). Determining from past data how these reviewers might have behaved in a different environment is difficult, as seen in the literature on valuation estimation in auctions [76]. We leave addressing these challenges for future work.

Chapter 7

On the Detection of Reviewer-Author Collusion From Manipulated Bidding

In this chapter, we return to the problem of reviewer-author collusion, this time from the perspective of *detection*. Approaches to addressing academic fraud can generally be divided into mitigation-based and detection-based methods [157]. In Chapter 6, we compared a variety of proposed modifications to paper assignment algorithms that aim at mitigating the impact of collusion rings [19, 61, 159], including the randomized assignment method we introduced in Chapter 2. However, such mitigation-based approaches necessarily come with a cost in terms of assignment quality, as they ignore some aspects of reviewer preferences expressed through bids. In contrast, detection methods have received little attention in prior work. Ideally, an accurate detection method could identify any colluders and remove them from the reviewer pool before the paper assignment. Such an approach could potentially lead to higher-quality assignments among the honest reviewers, as well as allow the conference organizers to manually investigate and take action against the colluders. In the words of Littman [99], *"Better paper-assignment technology would help close one loophole that is being exploited. But, without better investigative tools, we may never be able to hold the colluders to account."* However, careless deployment of detection algorithms may result in false positives: honest reviewers and authors being falsely identified as colluders. This is a serious danger in scientific peer review, since falsely-accused reviewers and authors may have their reputations tarnished or their papers unfairly rejected. Establishing effective methods to detect collusion rings is thus a critical path for research on this problem.

A long line of research has applied anomaly-detection techniques to successfully detect various forms of fraud in other settings. Many of these settings involve a network of people in which the fraud appears as a set of anomalous interactions: for example, auction fraud on online platforms, fraudulent financial transactions, fake reviews on sites such as Amazon, and fake accounts on social media [3]. This raises an important question of whether these techniques can be similarly applied to effectively detect fraudulent interactions in the context of paper bidding for peer review. However, there is a vast range of approaches to detection in the research literature that could potentially be applied to this problem. To facilitate a thorough analysis, this chapter focuses on methods that attempt to detect collusion using only the paper bidding and authorship data. Concretely, we consider the question: **is it possible to effectively detect collusion rings from only the bids and authorships?** Answering this question alone requires an exten-

sive empirical analysis. However, resolving this question provides important direction for future research on detecting collusion rings. Our focus on bidding is additionally motivated by the fact that real-world investigations of collusion often begin with analysis of the bidding data.

One major challenge in the peer review setting is that there is no ground-truth data about how colluding reviewers behave in practice. In Chapter 8, as one approach to solve this, we observe paper bidding in a mock conference setting and collect data on malicious reviewer strategies. In this chapter, instead of assuming a specific form of collusive bidding behavior, we address this challenge by considering a wide range of parameterized behavior for the collusion rings. Specifically, since the purpose of a collusion ring is to achieve colluder-to-colluder paper assignments, we vary the density of bidding between colluding reviewers and colluder-authored papers. Denser bidding corresponds to a stronger attack, but also may allow the colluding group to be more easily detected. Thus, our analysis aims to establish the regions of the space of colluder behavior at which collusion can be effectively detected.

Our contribution in this chapter is a set of empirical results on two different realistic bidding datasets concerning the feasibility of detecting collusion rings. Our analysis consists of three parts. First, we characterize the typical density of bidding found within groups of honest (non-colluding) reviewers, as high-density groups of honest reviewers are potential false positives for detection algorithms. Second, we evaluate the performance of existing algorithms at detecting injected collusion rings based on the anomalous density of bidding within the ring. We consider a combination of fundamental approaches to finding dense subgraphs and density-based fraud detection methods, including two algorithms that have been shown to effectively detect fraud in other settings. Third, we evaluate the success of the injected collusion rings at achieving their desired paper assignments, contextualizing the potential impact of fraud.

Overall, our results suggest that collusion rings cannot be effectively detected from only the bidding and authorship data. Our findings include the following:

- All detection algorithms fail to detect some injected collusion rings that are larger than any honest-reviewer groups with a similar density of bidding. For example, when the collusion ring consists of 10 reviewers who bid on all of each others' authored papers, the output of the best-performing detection algorithm across both datasets has only a 31% overlap (Jaccard similarity) with the true colluders on average.
- Colluders are able to achieve assignment to a substantial fraction of the papers authored by other colluders while avoiding detection by all algorithms (30% and 24% in each of the two datasets).
- A sizeable fraction of colluders can get at least one of their papers reviewed by another colluder while avoiding detection (54% and 35% in each of the two datasets).

These results suggest that future research on detecting collusion rings must focus on more complex detection methods that leverage various other metadata, such as reviewer-paper text-similarity scores or reviewer publication history.

The code and data we use in our analysis is available online at `https://github.com/sjecmen/peer-review-collusion-detection`.

Related Work: Fraud Detection. Outside of the setting of scientific peer review, similar problems of fraud have been studied in the anomaly detection literature. In online platforms such as

Amazon or Yelp, several methods have been proposed to detect products or sellers who purchase fraudulent reviews from users [2, 45, 90]. Other works propose density-based methods for detecting groups of fraudsters in these and similar online network settings (e.g., fraudulent transactions on eBay or fake followers on Twitter) [66, 116, 123]; we evaluate the performance of [66] in our analysis. However, our setting (scientific peer review) is distinct from these online platform settings in a few ways. Most significantly, our work focuses on collusion in the paper bidding phase, where the objective of colluders is to achieve a desired outcome in the subsequent paper assignment; in contrast, the fraudulent reviews are themselves the objective of fraudsters on (e.g.) Amazon. As a result, the incentives for peer-review colluders are not the same as those of frausters in the other settings: for example, since each reviewer is assigned to review a limited number of papers, "camouflaging" by bidding positively on non-colluder papers may result in those papers being assigned instead of the targeted papers. In addition, fraudulent interactions on online platforms are often provided by large numbers of fake accounts specifically used for fraud; since making fake accounts on common peer-review platforms is non-trivial, interactions between colluders in peer review may be more often done under their real identities, leading to different patterns of behavior. Numerous works have proposed other methods for graph-based anomaly detection, including those that address online spam, cyber-attacks, and other forms of fraud [3]. Generic approaches for dense-subgraph detection [58, 67, 150] can also be applied to detect anomalies in graphs; we evaluate these methods in our analysis.

7.1 Preliminaries

In this section, we detail the setting of our analysis, the datasets we analyze, and the research questions we answer.

7.1.1 Setting

We consider a conference peer review setting with a set of submitted papers $\mathcal{P}$ and a set of reviewers $\mathcal{R}$. Conferences generally recruit the authors of the submitted papers to serve as reviewers, along with external, non-author reviewers. The authorship set $\mathcal{A} \subset \mathcal{R} \times \mathcal{P}$ contains all pairs (r, p) such that reviewer r authored paper p. Additionally, the conflict-of-interest set $\mathcal{C} \subset \mathcal{R} \times \mathcal{P}$ contains all pairs (r, p) such that reviewer r has a conflict-of-interest with paper p and should not be assigned to review it ($\mathcal{A} \subseteq \mathcal{C}$). Once submissions are received, the conference asks each reviewer to indicate their level of interest on each submitted paper via paper bidding. While conferences often allow each reviewer to choose from a number of discrete levels (e.g., "Eager", "Willing", "Not willing"), we consider a simplified setting with binary bids (positive or neutral); note that allowing additional levels of bids can only give colluders more flexibility to manipulate the paper assignment. The bid set $\mathcal{B} \subset \mathcal{R} \times \mathcal{P}$ contains all pairs (r, p) such that reviewer r bid positively on paper p ($\mathcal{B} \cap \mathcal{C} = \emptyset$). The conference also computes text-similarity scores between each reviewer and paper using a function $T : \mathcal{R} \times \mathcal{P} \to [0, 1]$, where $T(r, p)$ indicates the level of similarity between the text of paper p and the text of the past work of reviewer r.

The conference then computes the paper assignment in the following manner. First, the conference computes similarity scores between each reviewer and paper using a function $S : \mathcal{R} \times \mathcal{P} \to [0, 1]$. In our experiments, we use $S(r, p) = \frac{1}{2} T(r, p) 2^{\mathbb{I}[(r,p) \in \mathcal{B}]}$ based on the function used

in the 2016 Conference on Neural Information Processing Systems (NeurIPS), a large machine-learning conference [136]. The conference then computes an assignment of papers to reviewers such that the total similarity of the assigned pairs is maximized, subject to constraints that (i) each paper is assigned to exactly 3 reviewers, (ii) each reviewer is assigned at most 6 papers, and (iii) no reviewer-paper pairs in $\mathcal{C}$ are assigned. While we use the stated values in our experiments, the exact reviewer and paper loads vary between conferences. The maximum-similarity assignment can be computed efficiently as a min-cost flow or as a linear program. This framework for paper assignment has been used in numerous conferences and venues [134].

7.1.2 Datasets

We next provide details regarding the two datasets that we analyze in this chapter.

The first dataset, which we refer to as "AAMAS", contains a subset of the real bidding from the 2021 International Conference on Autonomous Agents and Multiagent Systems, an AI conference. This dataset is publicly available from PrefLib [106] and contains de-identified bids from reviewers that did not opt-out from data collection. In this conference, bids were selected from {"Yes", "Maybe", "Conflict", "No response"}; we consider "Yes" and "Maybe" responses to be positive bids and include those reviewer-paper pairs in $\mathcal{B}$. We set $\mathcal{C}$ as the set of all reviewer-paper pairs with "Conflict" bids. Since the dataset does not include authorship information, we reconstruct the authorships $\mathcal{A}$ by subsampling 3 conflicts-of-interest uniformly at random for each paper. The resulting dataset has 526 papers and 596 reviewers, 398 of whom authored at least one paper. The dataset also does not contain text-similarity scores $T(r, p)$. We generate synthetic text-similarity scores using the procedure described in Section 7.4.1, based on the text-similarities from the 2018 International Conference on Learning Representations reconstructed by [160].

Our second dataset, which we refer to as "S2ORC", is the semi-synthetic dataset constructed and made publicly available by [159]. This dataset contains synthetic bids between a large subset of published computer science papers and authors from the Semantic Scholar Open Research Corpus [6], designed to match statistics from the NeurIPS 2016 conference [136]. Bids were chosen from values {0, 1, 2, 3}, where non-zero values indicate a positive bid (included in $\mathcal{B}$). For the authorship set $\mathcal{A}$, we use the real authorships between the reviewers and papers in the dataset, discarding the 90 bids placed on self-authored papers. We assume that the conflicts-of-interest are only the authorships ($\mathcal{C} = \mathcal{A}$). The resulting data has 2446 papers and 2483 reviewers, 984 of whom authored at least one paper. This dataset also contains real text-similarity scores between each reviewer and paper $T(r, p)$, computed using the popular TPMS algorithm [29]. We compare the level of agreement between these text-similarity scores and the bids to those found by [143] and find that they have a realistic level of error; see Section 7.4.1 for details.

In our analysis, we assume that both of these datasets contain only bids from "honest" (non-colluding) reviewers. The AAMAS dataset contains information from reviewers that did not opt-out from the data collection, and we expect that any colluding reviewers in the conference would have done so. The S2ORC bids are synthetic and thus do not model malicious reviewer behavior.

7.1.3 Problem Statement

Our goal is to detect collusion rings. We suppose that there exists a group of colluding reviewers $\mathcal{M} \subset \mathcal{R}$ who try to manipulate the paper assignment by altering their bids, with the aim of being assigned to review the papers authored by other members of the colluding group. The objective of a collusion-detection algorithm is to output $\mathcal{M}$ given the set of bids $\mathcal{B}$, along with the authorships $\mathcal{A}$ and the other conflicts $\mathcal{C}$. Note that the text-similarities $T(r, p)$ cannot be used for detection in our analysis–we consider the problem of detection using only the bidding itself and the authorships.

As our original datasets do not contain colluding reviewers, we inject collusion into the datasets by choosing a group of reviewers to be the colluders $\mathcal{M}$ and modifying the bids of these reviewers (i.e., adding or removing elements of $\mathcal{M} \times \mathcal{P}$ to/from $\mathcal{B}$). The strongest form of collusion would be to add bids between all reviewers in $\mathcal{M}$ and all papers authored by other reviewers in $\mathcal{M}$, while additionally removing all other bids by reviewers in $\mathcal{M}$. However, malicious reviewers may not want to perform such an obvious manipulation out of fear of being caught. Due to the lack of concrete evidence on colluding groups and the exact bidding strategy that they might employ, we consider a wide range of possible colluding groups parameterized by both a size parameter (i.e., the number of colluding reviewers) and a "density" parameter (roughly corresponding to the attack strength).

As there are some modeling choices involved in concretely defining the bidding density parameter, we consider two different notions of density, corresponding to two different graph representations of the bidding relationships between the reviewers of the conference. The first representation (Section 7.2) is a unipartite graph in which vertices represent reviewers. The second representation (Section 7.3) is a bipartite graph in which vertices represent both reviewers and papers. We motivate and define each of these formulations in their respective sections. For each graph formulation, we investigate the following three high-level research questions:

- **Q1:** For what values of size and density do groups of honest reviewers already exist in the dataset (without injected collusion)? (Sections 7.2.1 and 7.3.1)
- **Q2:** For what values of size and density can groups of injected colluders be accurately detected by existing algorithms? (Sections 7.2.2 and 7.3.2)
- **Q3:** At each size and density, how successful are groups of injected colluders at achieving their desired paper assignments? (Sections 7.2.3 and 7.3.3)

All of our experiments in these sections were conducted on a server with 515 GB RAM and 112 CPUs.

7.2 Unipartite Bidding Graph

In this section, we represent the reviewer bidding data in the form of a unipartite, directed graph where each vertex corresponds to a reviewer. The graph contains a directed edge (r_1, r_2) if reviewer r_1 bid on at least one paper authored by reviewer r_2. Informally, the density of a group of reviewers in this graph is the fraction of possible edges present between the reviewers. We formalize these definitions shortly. In Section 7.3, we consider an alternative, bipartite graph representation of the bidding.

Our motivation for considering this graph representation is that it most naturally represents

the reciprocal behavior expected of a collusion ring. Several existing works [19, 61, 96] on reviewer-author collusion consider a similar reviewer-to-reviewer graph, aiming to prevent cycles (i.e., "rings") between reviewers in the paper assignment. For our purposes, the notion of density implied by this graph has several useful properties. First, malicious reviewers may not attempt to manipulate the assignment of every paper they author. For example, each colluder might have one bad paper that they want their fellow colluders to review (e.g., because it is lower quality) and many good papers not involved in the manipulation. Even if each colluder authors many good papers, collusion would still appear as a dense subgraph. Additionally, collusion rings are based on a quid-pro-quo arrangement between the colluders, where each colluder needs help getting some papers accepted to the conference. Each colluder therefore should receive some benefit from the other colluders, reflected as density.

Formally, we denote the graph by $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$, where $\mathcal{V}_1 = \mathcal{R}$ and $\mathcal{E}_1 = \cup_{p \in \mathcal{P}} \{(r_1, r_2) \in \mathcal{R}^2 : (r_1, p) \in \mathcal{B} \wedge (r_2, p) \in \mathcal{A}\}$. Given this graph representation, we characterize a potential group of colluding reviewers $\mathcal{S} \subseteq \mathcal{R}$ in terms of two parameters. The first parameter $k_{\mathcal{S}} = |\mathcal{S}|$ is the size of the group. The second parameter $\gamma_{\mathcal{S}}$ is the "edge density" of the group, defined as follows. For an arbitrary graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and any subset of vertices $\mathcal{V}' \subseteq \mathcal{V}$, we use $\mathcal{E}[\mathcal{V}']$ to denote the set of edges in the subgraph induced by $\mathcal{V}'$. We then define the edge density of $\mathcal{S}$ to be $\gamma_{\mathcal{S}} = \frac{|\mathcal{E}[\mathcal{S}]|}{2\binom{|\mathcal{S}|}{2}}$, the fraction of possible edges present. We henceforth omit the subscript $\mathcal{S}$ from $k_{\mathcal{S}}$ and $\gamma_{\mathcal{S}}$ since the subset in question will be clear from context.

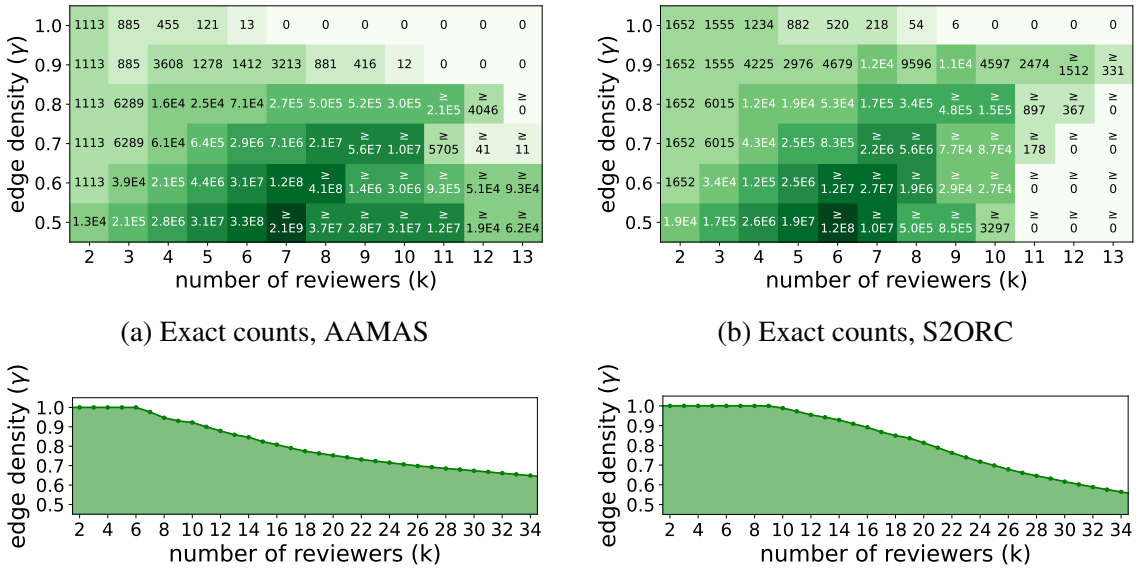
We now analyze the problem of detecting collusion from $\mathcal{G}_1$. In the following three subsections, we provide empirical analysis to answer each of the three research questions identified in Section 7.1.3.

7.2.1 Honest-Reviewer Groups (Q1)

In this subsection, we characterize the density of bidding among groups of honest (i.e., non-colluding) reviewers in our datasets. For each value of the parameters (k, γ), we count the number of groups of size k with edge density at least γ. Since we aim to detect collusion among authors, we consider only reviewers that have authored at least one paper. This shows the frontier of identifiable detection: if honest-reviewer groups exist at some (k, γ), then a colluding group with that same size and density cannot be identified as suspicious with high confidence based on the bidding within the colluding group. Stated differently, this analysis gives the number of potential false positives for a detection algorithm at each (k, γ). Recall that false positives are a serious concern in collusion detection due to the danger to honest reviewers' reputations.

In Figures 7.1a-7.1b, we show the results of this analysis. These counts were obtained via a standard backtracking search. Cells marked with "$\geq$" were unable to be completed in 24 hours, as obtaining exact counts takes worst-case time exponential in k. Instead, the values in these cells represent lower-bounds. We note that these figures may be independently useful to conferences, since many conferences use heuristic checks for collusion based on reviewer bidding patterns as part of the paper assignment (e.g., not assigning certain reviewer-paper pairs or ignoring certain reviewers' bids). The results in Figures 7.1a-7.1b may help these conferences understand the false positive rates of such heuristics.

Since exact counts are infeasible for large values of k, we additionally use a heuristic method to find dense subgraphs of larger sizes. We start with the entire graph and iteratively remove the



(a) Exact counts, AAMAS

(b) Exact counts, S2ORC

(c) Groups found by greedy peeling, AAMAS

(d) Groups found by greedy peeling, S2ORC

Figure 7.1: Exact counts of honest-reviewer groups with varying size and edge density (Figures 7.1a-7.1b), and the size and edge density of honest-reviewer groups found by a heuristic method (Figures 7.1c-7.1d). In Figures 7.1a-7.1b, values in cells marked with "$\geq$" represent lower bounds since exact counts were infeasible to compute. In Figures 7.1c-7.1d, each point corresponds to an existing honest-reviewer group (found by a greedy peeling method), and the shaded area indicates the region in which there exists at least one honest-reviewer group. Note that the vertical axis does not start at 0 for easier comparison with other figures.

vertex of smallest degree to produce a sequence of subsets of decreasing size (commonly called "greedy peeling"). In Figures 7.1c-7.1d, we plot the edge density γ of these subsets against the size k. Thus, for all points (k, γ) in the shaded region, at least one group of honest reviewers exists of size k and with edge density at least γ.

Overall, these results show that there is a large region of the space of (k, γ) where a colluding group could not be detected based on its size and edge density due to the existence of many similar groups of honest reviewers.

7.2.2 Detection Algorithm Evaluation (Q2)

For values of (k, γ) larger than those that exist in the honest bidding, we may hope that colluding groups can be identified as suspicious by an appropriate algorithm. However, this may not always be possible: since the size of the colluding group is unknown, a detection algorithm must identify that the colluding group is more suspicious than the honest-reviewer groups of different sizes. In this section, we evaluate whether several existing techniques for dense-subgraph discovery suffice to detect colluding groups with larger values of (k, γ), including one algorithm demonstrated to effectively detect fraud on Twitter.

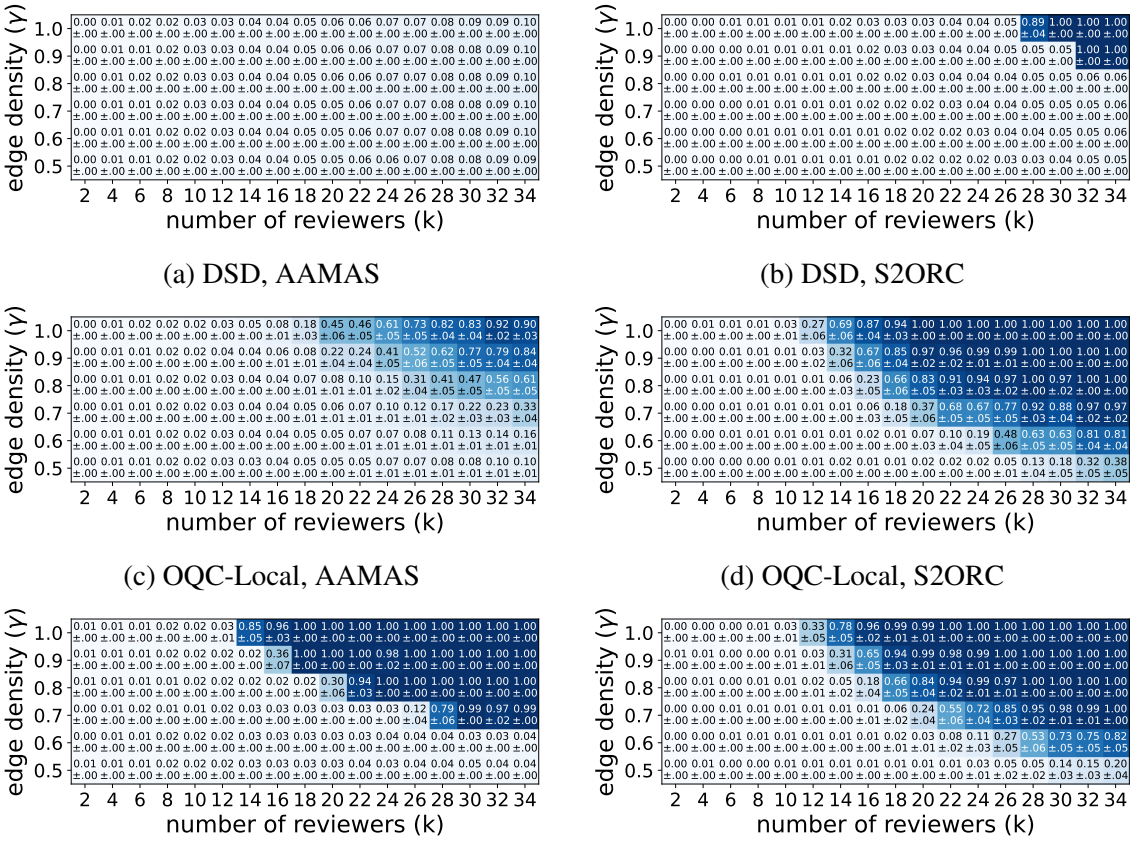
We consider two desiderata in selecting algorithms to evaluate. First, the algorithms should not require explicitly specifying the size of the output subset. This is a requirement in our setting,

since conference program chairs have no evidence regarding the size of the colluding groups that they can use to direct the detection algorithms. Second, our algorithms should be based on different notions of subgraph density that implicitly balance the size and edge density of the groups in a different way. Since $\mathcal{G}_1$ is unattributed, this choice is the primary design decision made by a density-based detection algorithm. By considering a variety of such choices, we hope to find one that detects the true colluders across the largest possible range of injected size and density. As a result, we consider the following algorithms:

- Traditional densest-subgraph discovery [28, 58]: Output the subset of vertices $\mathcal{S}$ that maximizes $f(\mathcal{S}) = |\mathcal{E}_1[\mathcal{S}]|/|\mathcal{S}|$. This corresponds to the subgraph with highest average degree. In our case, since we count all edges in both directions, this is a generalization of the standard definition for undirected graphs. To solve this problem, we implement the LP-based exact algorithm from [28]. We refer to this as "DSD".
- Optimal quasi-clique discovery [150]: For a given parameter $\alpha \in [0, 1]$, output the subset of vertices $\mathcal{S}$ that maximizes the "edge surplus" $f(\mathcal{S}) = |\mathcal{E}_1[\mathcal{S}]| - \alpha 2\binom{|\mathcal{S}|}{2}$. The second term in $f(\mathcal{S})$ corresponds to the expected number of edges in the subgraph if each edge occurs independently with probability α. Since we count all edges in both directions, we add the factor of 2 in the second term as compared to the standard definition for undirected graphs. To solve this problem, we implement the two approximation algorithms proposed by [150], one based on greedy peeling and one based on local search. We refer to these algorithms as "OQC-Greedy" and "OQC-Local" respectively. As recommended in [150], we set $\alpha = 1/3$.
- TellTail [67]: This method first defines the "adjusted mass" of a subset as the difference between the number of edges in the subset and the expected number of edges in the subset if edges are rewired randomly (preserving vertex degrees). Subsets are then scored based on the probability of the adjusted mass under a fitted Generalized Pareto distribution. Most parameters of this distribution are fixed as constants based on empirical observations across several datasets. Since this method operates on undirected graphs and is non-trivial to adapt to a directed setting, we first map our bidding graph $\mathcal{G}_1$ to an undirected graph $\mathcal{G} = (\mathcal{V}_1, \mathcal{E})$ before inputting it to this algorithm. The input graph $\mathcal{G}$ has the same vertex set as $\mathcal{G}_1$ and has an edge (r_1, r_2) iff both edges $(r_1, r_2) \in \mathcal{E}_1$ and $(r_2, r_1) \in \mathcal{E}_1$; that is, $\mathcal{E} = \{(r_1, r_2) \in \mathcal{R}^2 : (r_1, r_2) \in \mathcal{E}_1 \wedge (r_2, r_1) \in \mathcal{E}_1\}$). Denote the degree of a vertex v in $\mathcal{G}$ by $deg(v)$ and the CDF of the Generalized Pareto distribution as F_{GP}. Concretely, the algorithm finds a subset of vertices $\mathcal{S} \subseteq \mathcal{V}$ that approximately maximizes the objective function $f(\mathcal{S}) = F_{GP}\left(|\mathcal{E}[\mathcal{S}]| - \frac{\sum_{v \in \mathcal{S}} deg(v)}{4|\mathcal{E}|}\right)$. Note that this algorithm implicitly takes into account the connectivity between the chosen subset and the rest of the graph.

DSD, OQC-Greedy, and OQC-Local operate on pre-specified notions of density, without considering the sparsity of subgraphs in the input graph. In contrast, TellTail defines density in a data-driven fashion by identifying the distribution of masses that subgraphs of certain sizes follow, arguing that other notions of density tend to be biased towards larger subgraphs. TellTail was also shown to detect fraudulent followers on Twitter. These algorithms cover a representative set of the landscape of dense-subgraph mining solutions.

We evaluate the detection algorithms against the following collusion model. For each setting



(a) DSD, AAMAS

(b) DSD, S2ORC

(c) OQC-Local, AAMAS

(d) OQC-Local, S2ORC

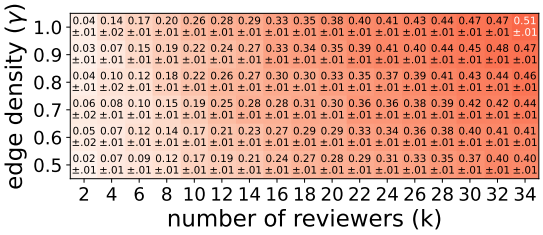
(e) TellTail, AAMAS

(f) TellTail, S2ORC

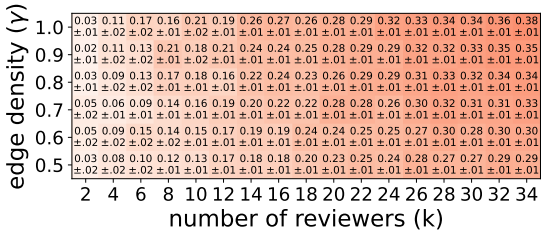
Figure 7.2: Performance of detection algorithms on $\mathcal{G}_1$. Values indicate the mean Jaccard similarity between the true set of colluders and the algorithm output, along with standard errors. Higher values correspond to better detection performance.

of (k, γ), we choose a subset of k reviewers uniformly at random from among those reviewers that authored at least one paper. We then add edges uniformly at random between reviewers until the subgraph has edge density at least γ. This modified graph is then passed as input to each detection algorithm. We repeat this procedure for 50 trials for each setting. We report the mean Jaccard similarity between the set of injected colluders and the set of reviewers output by the detection algorithms.

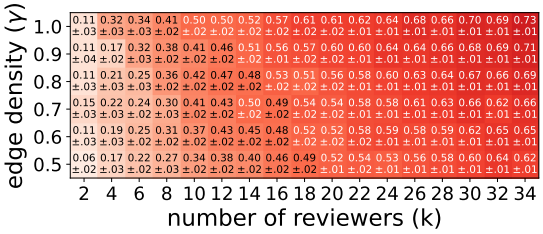
For the detection algorithms that use local search to optimize their objective (OQC-Local and TellTail), we return the best result over 11 initializations. The first run is initialized according to the triangle-counting heuristic suggested in [150], and the remaining 10 runs are initialized uniformly at random. We find that the detection performance of OQC-Local is significantly better when only the heuristic initialization is used, since the random initializations often resulted in output with a higher objective value but lower overlap with the true colluders; thus, we show the results with heuristic initialization only in this section and defer the results with all initializations to Section 7.4.2. However, the poor performance of OQC-Local when all initializations are used indicates that the objective value of OQC-Local is misaligned with the detection objective.



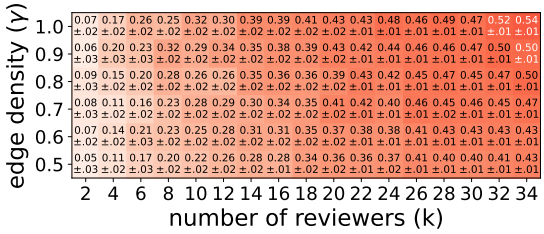
(a) Fraction of target papers with a colluder assigned, AAMAS



(b) Fraction of target papers with a colluder assigned, S2ORC



(c) Fraction of colluders with a colluder assigned, AAMAS



(d) Fraction of colluders with a colluder assigned, S2ORC

Figure 7.3: Success of colluders in terms of k and γ. Values indicate the mean for each metric along with standard errors.

The results for DSD, OQC-Local, and TellTail are shown in Figure 7.2. Results for OQC-Greedy are shown in Section 7.4.2 as it generally performs worse than OQC-Local. In both datasets, DSD performs very poorly; this is because it consistently returns a overly large subset of reviewers. On the AAMAS dataset, OQC-Local does somewhat better, achieving Jaccard similarity above 0.5 for values of $k \geq 26$ and $\gamma \geq 0.9$. TellTail performs by far the best, detecting colluders consistently for moderate-to-large values of (k, γ). However, there exists a wide range of settings in which all algorithms fail to detect injected colluders: for example, reviewer groups with $8 \leq k \leq 12$ and $\gamma = 1.0$ do not exist in the honest bidding and yet are still not detected by any algorithm. On the S2ORC dataset, OQC-Local and TellTail appear to perform equally well. There is a similar region in which honest-reviewer groups do not exist and yet detection fails to identify the correct group with high probability (e.g., $10 \leq k \leq 12$ with $\gamma = 1.0$), albeit smaller than in the AAMAS dataset.

7.2.3 Manipulation Success Evaluation (Q3)

In this subsection, we evaluate the impact of collusion in terms of the colluders' success at achieving their desired paper assignments, contextualizing the results of the previous two subsections. We conduct experiments in which we inject colluding groups into $\mathcal{G}_1$ for each setting of the parameters (k, γ) in the exact same manner as in Section 7.2.2. We then fix this modified version of $\mathcal{G}_1$ (i.e., the input to the detection algorithms in Section 7.2.2) as the "target graph" and modify $\mathcal{B}$ in order to realize this graph.

However, there are many possible strategies that these colluders could employ to modify their bids that correspond to the addition of these edges, since each edge (r_1, r_2) denotes the

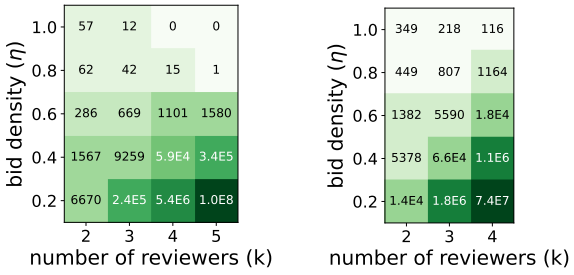
presence of at least one bid from reviewer r_1 on a paper authored by reviewer r_2. Additionally, the relationship between bids and edges is not one-to-one due to co-authorship between reviewers, which means that exactly realizing the target graph via bidding may not be possible. Due to these challenges, we instead modify the bids of colluders to achieve a modified version of $\mathcal{G}_1$ that is "no more suspicious" than the target graph. Specifically, we consider each edge (r_1, r_2) in the target graph such that r_1 is in the injected colluding group. If r_2 is also a colluder, we add bids from r_1 on each paper authored by r_2; otherwise, we choose one existing bid from r_1 on a paper authored by r_2 uniformly at random and remove all other such bids. In this way, we ensure that the edges within the injected colluder group are a subset of the edges in the target graph and the edges outside the injected colluder group are a superset of the edges in the target graph. Subject to these constraints, this procedure allows colluders to bid according to a worst-case strategy.

Given the modified bids, we compute a paper assignment as detailed in Section 7.1.1. Since the exact objective of colluders is not obvious, we evaluate the success of the malicious reviewers in two ways. First, we count the fraction of colluder-authored "target" papers with at least one colluder assigned; this indicates the extent to which colluders succeeded at influencing the acceptance decision for all of their papers. This may be too harsh of a metric, since colluders may not aim to influence the decision for all of their papers simultaneously. Second, we count the fraction of colluders who authored at least one paper that has at least one other colluder assigned; this indicates the fraction of colluders who recieved some benefit from participating in the collusion ring. We conduct 50 trials for each setting of the parameters and report the mean for each of the two metrics.

The results are shown in Figure 7.3. For the results on the AAMAS dataset, of particular interest are the results with $k \leq 20$ and $\gamma \leq 0.8$, since these were not detected by any of the algorithms in Section 7.2.2. We see that at the extremes of this region, approximately one-third of colluder-authored papers have at least one colluder assigned, and approximately one-half of colluders have at least one colluder assigned to one of their papers. At $(k = 10, \gamma = 0.8)$, where a larger number of honest-reviewer groups exist, both success metrics are also reasonably high. The results on the S2ORC dataset are similar: for example, at $(k = 14, \gamma = 0.8)$, 22% of target papers and 35% of colluders are successfully targeted while all detection algorithms achieve low success rates. Similarly, 21% of papers and 34% of colluders are successfully targeted when colluders are camouflaged among numerous honest-reviewer groups at $(k = 12, \gamma = 0.9)$. Thus, colluders can influence the acceptance decisions for a moderate fraction of their submitted papers without being detected by the evaluated algorithms. The counts of honest-reviewer groups in Section 7.2.1 suggest that a portion of the success of colluders may be unavoidable in this setting, regardless of the strength of the detection algorithms.

7.3 Bipartite Bidding Graph

In this section, we represent the reviewer bidding data as a bipartite graph, in which one set of vertices corresponds to reviewers and the other set of vertices corresponds to papers. This graph contains three types of undirected edges between a reviewer r and a paper p: bid edges, indicating that reviewer r bid positively on paper p; authorship edges, indicating that reviewer r authored paper p; and conflict-of-interest edges, indicating that reviewer r has a conflict-of-interest with paper p but did not author it. Our motivation for considering this graph is that it



(a) AAMAS (b) S2ORC

Figure 7.4: Exact counts of honest-reviewer groups with varying size and bid density.

directly represents all data that the detection algorithms have access to.

Formally, we denote this graph as $\mathcal{G}_2 = (\mathcal{V}_2, (\mathcal{E}_2^{(B)}, \mathcal{E}_2^{(A)}, \mathcal{E}_2^{(C)}))$, where $\mathcal{V}_2 = \mathcal{R} \cup \mathcal{P}$ is the vertex set and $\mathcal{E}_2^{(B)} = \mathcal{B}$, $\mathcal{E}_2^{(A)} = \mathcal{A}$, and $\mathcal{E}_2^{(C)} = \mathcal{C} \setminus \mathcal{A}$ are the sets of bid, authorship, and conflict edges respectively. As in Section 7.2, we characterize a (potentially colluding) group of reviewers $\mathcal{S} \subseteq \mathcal{R}$ in terms of a size parameter $k_{\mathcal{S}} = |\mathcal{S}|$ and a density parameter $\eta_{\mathcal{S}}$. In $\mathcal{G}_2$, we consider a new notion of density, which we call "bid density". For any subset of reviewers $\mathcal{R}' \subseteq \mathcal{R}$, define $\mathcal{P}[\mathcal{R}'] = \cup_{r \in \mathcal{R}'}\{p \in \mathcal{P} : (r, p) \in \mathcal{A}\}$ to be the subset of papers authored by at least one reviewer in $\mathcal{R}'$. The bid density of $\mathcal{S}$ is then defined as the total number of bids made by reviewers in $\mathcal{S}$ on papers in $\mathcal{P}[\mathcal{S}]$, divided by the maximum possible number of bids by reviewers in $\mathcal{S}$ on papers in $\mathcal{P}[\mathcal{S}]$:

$$\eta_{\mathcal{S}} = \frac{|\mathcal{E}_2^{(B)}[\mathcal{S} \cup \mathcal{P}[\mathcal{S}]]|}{|\mathcal{S}||\mathcal{P}[\mathcal{S}]| - |\mathcal{E}_2^{(A)}[\mathcal{S} \cup \mathcal{P}[\mathcal{S}]]| - |\mathcal{E}_2^{(C)}[\mathcal{S} \cup \mathcal{P}[\mathcal{S}]]|}.$$

We again will omit the subscript $\mathcal{S}$ from $k_{\mathcal{S}}$ and $\eta_{\mathcal{S}}$ since it will be clear from context.

We now analyze the problem of detecting collusion from $\mathcal{G}_2$. As in Section 7.2, the following three subsections present empirical analysis aimed at answering the three research questions identified in Section 7.1.3.

7.3.1 Honest-Reviewer Groups (Q1)

First, we count the number of honest-reviewer groups in $\mathcal{G}_2$ for each value of size k and bid density η. We consider only reviewers that have authored at least one paper. Since the bid density for a group is naturally lower than the edge density in general, we consider values of bid density ranging between 0.2 and 1.0.

The counts are shown in Figure 7.4. Due to the more complicated definition of bid density, we cannot efficiently prune branches of the search tree and must enumerate all $\binom{|\mathcal{R}|}{k}$ reviewer subsets. As a result, the range of feasible k are significantly more limited than in Section 7.2.1: counts for $k \geq 6$ for AAMAS and $k \geq 5$ for S2ORC could not be completed in 24 hours. In S2ORC, we see that numerous honest-reviewer groups do exist at every feasible setting, up to $\eta = 1.0$ and $k = 4$. Honest-reviewer groups are less prevalent in AAMAS, but still exist at $k = 5$ and $\eta \leq 0.8$. Like the figures in Section 7.2.1, these figures may be independently useful to conferences as indicators of the number of false positives for any heuristic defenses against collusion they use.

As in Section 7.2.1, we also run a greedy peeling method to heuristically find high-density reviewer groups of larger sizes. However, this method performs very poorly and so we relegate the results to Section 7.4.2.

7.3.2 Detection Algorithm Evaluation (Q2)

Next, we evaluate the performance of a variety of dense-subgraph discovery algorithms at detecting injected collusion in $\mathcal{G}_2$, including an influential fraud-detection algorithm for the online review setting. Specifically, we consider the following algorithms:

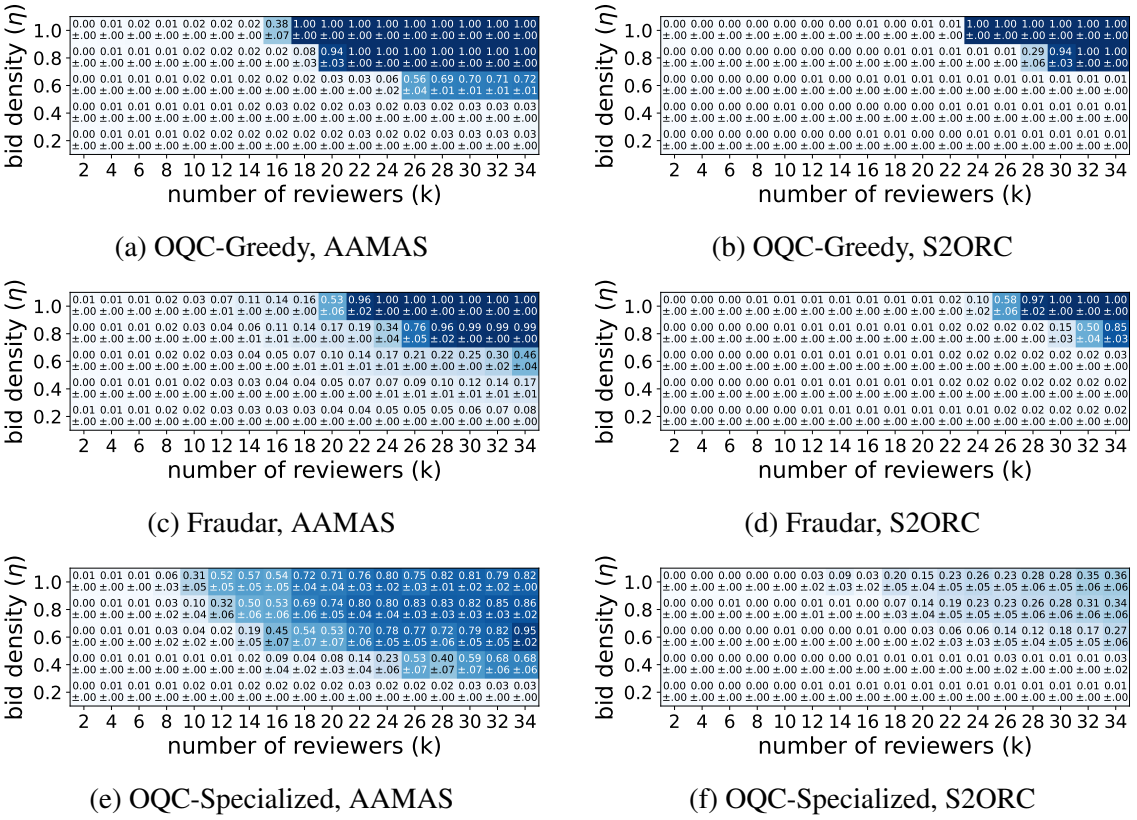
- DSD, OQC-Greedy, OQC-Local, and TellTail: These algorithms are introduced in Section 7.2.2. As input to each algorithm, we provide an unattributed bipartite graph $\mathcal{G} = (\mathcal{V}_2, \mathcal{E})$ with the same vertex set as $\mathcal{G}_2$. We consider two variants for the edge set of the input graph: the edge set consists of only the bid edges $\mathcal{E} = \mathcal{E}_2^{(B)}$, and the edge set consists of the union of the bid and authorship edges $\mathcal{E} = \mathcal{E}_2^{(B)} \cup \mathcal{E}_2^{(A)}$. In the OQC algorithms, we use the original objective function for undirected graphs $f(\mathcal{S}) = |\mathcal{E}[\mathcal{S}]| - (1/3)\binom{|\mathcal{S}|}{2}$.
- Fraudar [66]: This algorithm is designed to detect fake product reviews (e.g., on Amazon/Yelp) or fake followers (e.g., on Twitter) from a bipartite graph of users and products $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E}$ contains edges from users to products. Denote by $deg(v)$ the degree of a vertex v in $\mathcal{G}$. The returned subset of vertices $\mathcal{S} \subset \mathcal{V}$ is chosen to approximately maximize the objective function $f(\mathcal{S}) = \frac{1}{|\mathcal{S}|}\sum_{(u,v)\in\mathcal{E}[\mathcal{S}]} \left(\log(5 + deg(v))\right)^{-1}$. This objective is "camouflage-resistent", meaning that fraudulent users cannot affect their own objective value by adding extra edges to honest products. In our setting, we consider the reviewers to be users and papers to be products. We again input an unattributed bipartite graph with the same vertex set as $\mathcal{G}_2$ and two variants for the edge set: $\mathcal{E} = \mathcal{E}_2^{(B)}$ and $\mathcal{E} = \mathcal{E}_2^{(B)} \cup \mathcal{E}_2^{(A)}$.
- OQC-Specialized: We additionally adapt the objective function of the optimal quasi-clique discovery algorithm to our setting in a natural way. For a subset of reviewers $\mathcal{S} \subseteq \mathcal{R}$, we define the objective function

$$f(\mathcal{S}) = |\mathcal{E}_2^{(B)}[\mathcal{S} \cup \mathcal{P}[\mathcal{S}]]| - \alpha\Big(|\mathcal{S}||\mathcal{P}[\mathcal{S}]| - |\mathcal{E}_2^{(A)}[\mathcal{S} \cup \mathcal{P}[\mathcal{S}]]| - |\mathcal{E}_2^{(C)}[\mathcal{S} \cup \mathcal{P}[\mathcal{S}]]|\Big).$$

  Analogous to the concept of "edge surplus" that motivates the optimal quasi-clique objective function, this corresponds to the number of excess bids above the expectation if each bid is made independently with probability α. We use local search to optimize this objective. As in the other OQC algorithms, we set $\alpha = 1/3$.

Since we aim to detect a group of colluding reviewers, we discard any paper vertices in the subset output by each algorithm, considering only the output reviewer vertices as the detected reviewers.

In addition to the algorithms from Section 7.2.2, Fraudar is an influential fraud-detection algorithm that leverages a density-based signal for detection. It was designed to detect fraudulent reviews on platforms like Amazon, a similar problem to our own. OQC-Specialized was not proposed in prior work, but naturally generalizes the objective function of OQC-Local to operate directly on $\mathcal{G}_2$.



(a) OQC-Greedy, AAMAS

(b) OQC-Greedy, S2ORC

(c) Fraudar, AAMAS

(d) Fraudar, S2ORC

(e) OQC-Specialized, AAMAS

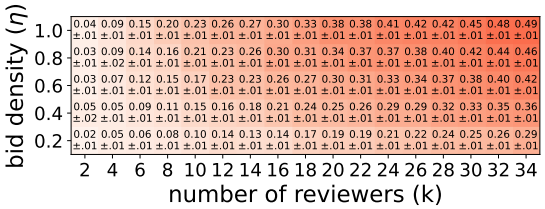
(f) OQC-Specialized, S2ORC

Figure 7.5: Performance of detection algorithms on $\mathcal{G}_2$. Values indicate the mean Jaccard similarity between the true set of colluders and the algorithm output, along with standard errors. Higher values correspond to better detection performance.

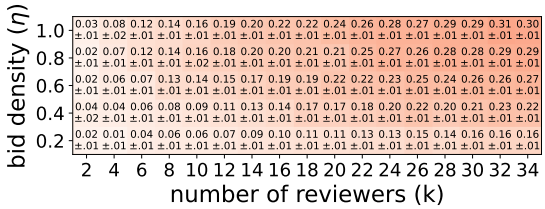
We now describe the collusion model against which we evaluate the detection algorithms. For each value of (k, η), we inject a group of colluding reviewers into $\mathcal{G}_2$ as follows. We first choose a set of k reviewers, denoted by $\mathcal{M}$, uniformly at random from among all reviewers who have authored at least one paper. We then choose a reviewer-paper pair uniformly at random from $\mathcal{M} \times \mathcal{P}[\mathcal{M}]$ and add this pair to $\mathcal{E}_2^{(B)}$ if there is no existing edge of any type between this pair. We repeat this until the bid density is at least η. This modified graph is then used to construct the input to each detection algorithm. We repeat this procedure for 50 trials for each setting of parameters. We report the mean Jaccard similarity between the set of injected colluders and the output set of reviewers from the detection algorithms.

As in Section 7.2.2, for each local search algorithm (OQC-Local, TellTail, OQC-Specialized), we return the best result over 11 initializations. The first initial subset is chosen according to a generalization of the heuristic suggested in [150]. For each vertex, we count the number of bid-author-bid-author cycles that this vertex participates in and divide by the total bid- and authorship-degree of the vertex; we then choose the initial subset to be the maximum vertex according to this metric along with all of its neighbors. The remaining 10 runs are initialized uniformly at random.

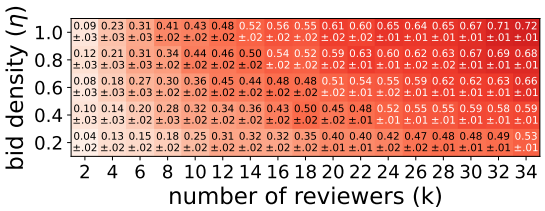
We show results for the OQC-Greedy, Fraudar, and OQC-Specialized algorithms in Fig-



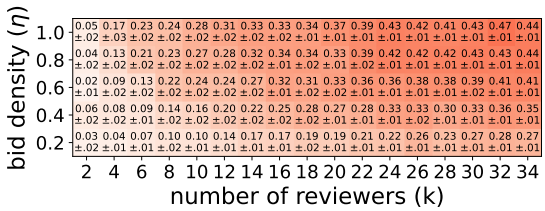
(a) Fraction of target papers with a colluder assigned, AAMAS



(b) Fraction of target papers with a colluder assigned, S2ORC



(c) Fraction of colluders with a colluder assigned, AAMAS



(d) Fraction of colluders with a colluder assigned, S2ORC

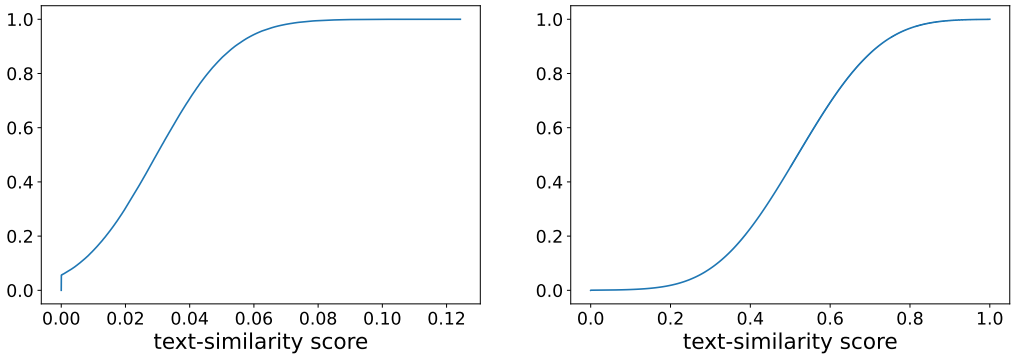
Figure 7.6: Success of colluders in terms of k and η. Values indicate the mean for each metric along with standard errors.

ure 7.5. Results for the remaining algorithms, which generally performed worse, are in Section 7.4.2. For all algorithms (other than OQC-Specialized), we find that the performance is very similar when the input edge set is $\mathcal{E}_2^{(B)}$ and when it is $\mathcal{E}_2^{(B)} \cup \mathcal{E}_2^{(A)}$; thus, all results shown are for the case with edge set $\mathcal{E}_2^{(B)}$. On AAMAS, we see that both OQC-Greedy and Fraudar are perform very well for high values of (k, η) (e.g., $k \geq 20, \eta \geq 0.8$). However, both of these algorithms fail to detect colluders entirely at more moderate values of k (e.g., $k = 14, \eta = 1.0$). OQC-Specialized achieves some success for a much wider range of parameters, but fails to achieve Jaccard similarities above 0.9 even for extreme values of (k, η); this may indicate that the true colluding group is not a local optimum for this algorithm's objective function. On S2ORC, performance of all algorithms is significantly worse. OQC-Greedy and Fraudar still identify the colluding groups for extreme values of (k, η), but OQC-Specialized fails to consistently identify the colluders for any parameter values. Overall, there exists a significant region of moderate parameter values at which no algorithm achieves good detection performance.

7.3.3 Manipulation Success Evaluation (Q3)

Finally, we evaluate the success of the colluders at manipulating the assignment as a function of the parameters (k, η). For each setting of these parameters, we inject a group of colluders as described in the preceding section and compute a paper assignment using the procedure detailed in Section 7.1.1. As in Section 7.2.3, we evaluate the success of the colluders in two ways: the fraction of colluder-authored papers with at least one colluder assigned, and the fraction of colluders with at least one colluder assigned to one of their authored papers. We conduct 50 trials for each setting of the parameters and report the mean for each of the two metrics.

The results are shown in Figure 7.6. On AAMAS, we see that at $(k = 16, \eta = 0.8)$ where no



(a) AAMAS (b) S2ORC

Figure 7.7: CDF of the text-similarity scores for reviewer-paper pairs without a conflict-of-interest. The mean text similarity score among these pairs is 0.030 for AAMAS and 0.52 for S2ORC.

detection algorithm was able to detect colluders with high probability, the colluders can successfully achieve assignments to 30% of their target papers and 54% of the colluders. Similar results are seen elsewhere on the frontier of the undetected region. On S2ORC, although the detection algorithms performed poorly, the success values are also lower than in the corresponding case on AAMAS. Still, in the cell $(k = 26, \eta = 0.8)$ where the detection algorithms performed poorly, colluders can achieve assignment to 26% of target papers and 42% of colluders, a sizeable influence on the paper assignment. We note that these success rates are quite similar to those found in Section 7.2.3 for the unipartite graph setting, which may provide some indication that the ability of undetected colluders to manipulate the assignment is robust to the exact graph representation. Overall, colluders are still able exert influence over the acceptance decisions for a non-trivial fraction of their own papers while avoiding detection.

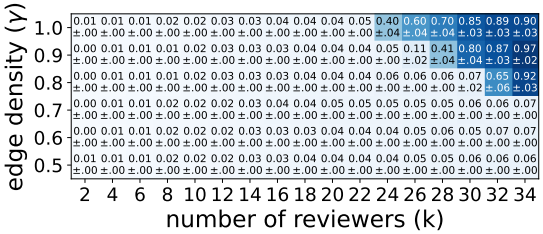
7.4 Supplemental Material

In this section, we present additional details and experimental results omitted from the previous sections.

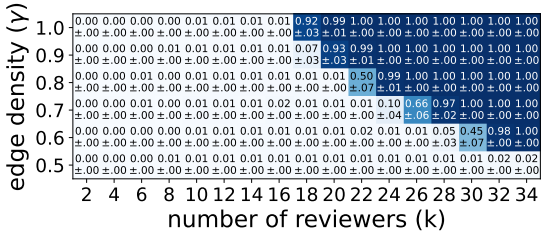
7.4.1 Text Similarity Details

In this subsection, we provide more details about the text-similarity scores in our datasets, introduced in Section 7.1.2. We synthetically generated the AAMAS text-similarity scores to supplement the original bidding dataset. The S2ORC text-similarity scores were included in the original dataset by [159], and were computed by running the widely-used TPMS algorithm on the abstracts of the reviewer's past papers and the paper in question. Figure 7.7 shows the CDFs of the text-similarity scores among the reviewer-paper pairs that did not have conflicts-of-interest.

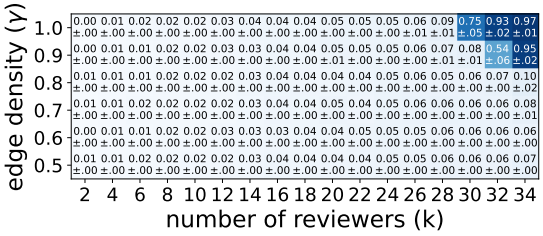
Since our work analyzes the effect of bids on the paper assignment, it is important to validate that our text similarity scores have a realistic level of agreement with the bids. We do this by comparing to the results of [143], which evaluates the accuracy of commonly-used text-similarity algorithms at predicting reviewer expertise using a ground-truth dataset. Specifically,



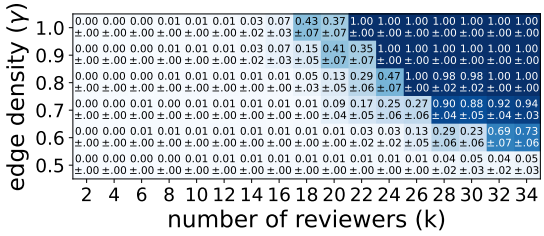
(a) OQC-Greedy, AAMAS



(b) OQC-Greedy, S2ORC

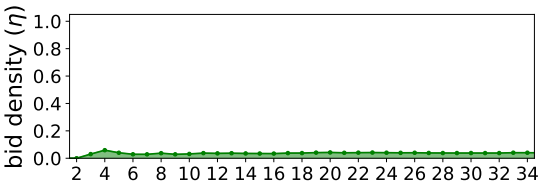


(c) OQC-Local (including random initializations), AAMAS

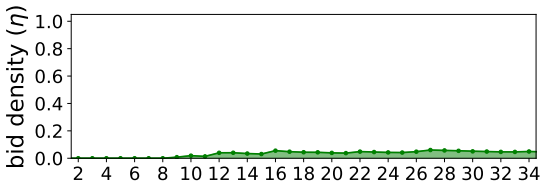


(d) OQC-Local (including random initializations), S2ORC

Figure 7.8: Performance of additional detection algorithms on $\mathcal{G}_1$. Values indicate the mean Jaccard similarity between the true set of colluders and the algorithm output, along with standard errors. Higher values correspond to better detection performance.



(a) AAMAS



(b) S2ORC

Figure 7.9: Size and bid density of honest-reviewer groups found by a heuristic method on $\mathcal{G}_2$. Each point corresponds to an existing honest-reviewer group (found by a greedy peeling method), and the shaded area indicates the region in which there exists at least one honest-reviewer group.

[143] evaluates the accuracy of the text-similarity scores by considering reviewer-paper-paper triples $(r, p_1, p_2) \in \mathcal{R} \times \mathcal{P} \times \mathcal{P}$ where reviewer r reported greater expertise on paper p_1 than on paper p_2. They find that the TPMS algorithm produces a weakly greater text-similarity score for (r, p_1) than for (r, p_2) on 80% of "easy" triples (where the reviewer reported high-expertise vs low-expertise) and on 62% of "hard" triples (where the reviewer reported high-expertise vs higher-expertise). We use these results to generate and/or validate the text-similarity scores in our datasets, considering the bids to be a proxy for reviewer expertise.

For the AAMAS dataset, text-similarity scores were sampled i.i.d. from one of three Gaussian distributions, depending on the bid value for that reviewer-paper pair (from {"Yes", "Maybe", "No response"}). For the "No response" pairs, the Gaussian distribution was fit to the dataset of text-similarities reconstructed from the 2018 International Conference on Learning Representa-

tions by [160]. The variance of the Gaussian distributions for the "Maybe" and "Yes" pairs were the same, but the means were chosen based on the statistics from [143]. Specifically, we set the means such that in expectation: (a) 80% of the triples $(r, p_1, p_2) \in \mathcal{R} \times \mathcal{P} \times \mathcal{P}$ where r bid "Yes" or "Maybe" on p_1 and r bid "No response" on p_2 had $T(r, p_1) \geq T(r, p_2)$; and (b) 62% of the triples $(r, p_1, p_2) \in \mathcal{R} \times \mathcal{P} \times \mathcal{P}$ where r bid "Yes" on p_1 and r bid "Maybe" on p_2 had $T(r, p_1) \geq T(r, p_2)$.

For the S2ORC dataset, we compare the text-similarity scores in the dataset to the statistics from [143]. Recall that the S2ORC dataset contains bid values for each reviewer-paper pair in $\{0, 1, 2, 3\}$. We find that 83% of the triples $(r, p_1, p_2) \in \mathcal{R} \times \mathcal{P} \times \mathcal{P}$ where r bid from {1, 2, 3} on p_1 and r bid 0 on p_2 had $T(r, p_1) \geq T(r, p_2)$. Additionally, we find that 65% of the triples $(r, p_1, p_2) \in \mathcal{R} \times \mathcal{P} \times \mathcal{P}$ where r bid 3 on p_1 and r bid {1, 2} on p_2 had $T(r, p_1) \geq T(r, p_2)$. We see that the accuracy of the text-similarities in the S2ORC dataset is similar to that of the ground-truth dataset.

7.4.2 Additional Experimental Results

In Figure 7.8, we provide evaluations of detection algorithms on $\mathcal{G}_1$ omitted from Section 7.2.2. This includes the OQC-Greedy algorithm, as well as the OQC-Local algorithm with all initializations (detailed in Section 7.2.2). These algorithms generally perform worse than those with results shown in Section 7.2.2.

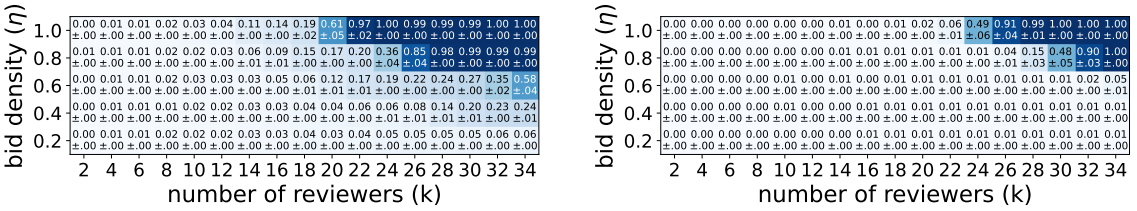
In Figure 7.9, we show the size and bid density of honest-reviewer groups detected by a greedy peeling method on $\mathcal{G}_2$ omitted from Section 7.3.1. In this method, we begin with the entire set of reviewers. At each iteration, we greedily remove the reviewer whose removal results in the highest bid density within the set of remaining reviewers. We then plot the bid density for each group size across the iterations. Clearly, this method performs very poorly at finding groups of high density.

In Figure 7.10, we present the results of the detection algorithms on $\mathcal{G}_2$ omitted from Section 7.3.2: DSD, OQC-Local, and TellTail. These algorithms generally perform worse than those with results shown in Section 7.3.2.

7.5 Discussion

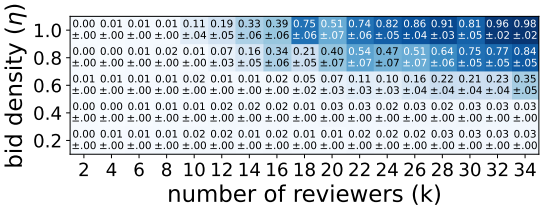
In this chapter, we provide an empirical exploration of the problem of detecting reviewer-author collusion rings from manipulated bidding without the use of other metadata. We frame the problem as a dense-subgraph discovery problem in two different graph representations. Overall, we find that malicious reviewers can manipulate the paper assignment to moderate success while remaining within typical or difficult-to-detect levels of density in the bidding graph. This provides evidence to support the conclusion that malicious reviewer behavior cannot be effectively detected using just the bidding data. While our analysis cannot conclusively prove that bidding data is insufficient to detect collusion, our methodology thoroughly explores this problem from several different analyses of realistic conference data. As such, if detection of malicious reviewers is possible, it likely requires leveraging other features in addition to the bids.

One limitation of our work is that our analysis is performed on semi-synthetic datasets, since real datasets containing bidding and authorship data are not publicly available. Thus, it is possible that the bidding in our datasets is not fully representative of real conference bidding: for example,

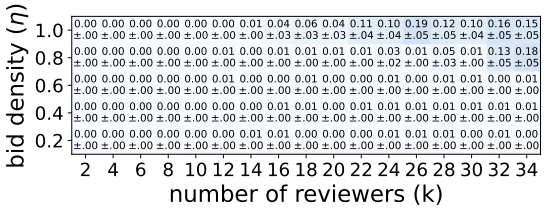


(a) DSD, AAMAS

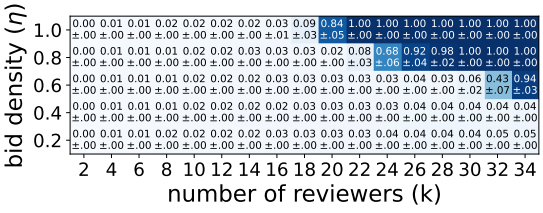
(b) DSD, S2ORC



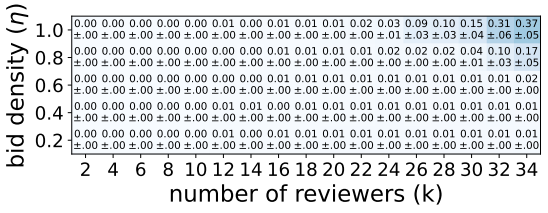
(c) OQC-Local, AAMAS



(d) OQC-Local, S2ORC



(e) TellTail, AAMAS



(f) TellTail, S2ORC

Figure 7.10: Performance of additional detection algorithms on $\mathcal{G}_2$. Values indicate the mean Jaccard similarity between the true set of colluders and the algorithm output, along with standard errors. Higher values correspond to better detection performance.

in the AAMAS dataset, honest reviewers with abnormal bidding patterns may have opted-out of data collection. To further verify our results, program chairs could replicate our methodology on their own conference's real bidding data. Another limitation of our work is that we consider an intentionally narrow research question: the feasibility of detection with a highly limited set of data. As one of the first works on detecting collusion, our analysis of this question provides a basis for future work to build on.

In Chapter 8, we directly gather a dataset on malicious reviewer bidding in a mock conference setting. We opt not to directly use this dataset in this chapter because, in the bids-only setting we consider, we can more confidently make a negative claim about the feasibility of detection by sweeping out a wide range of malicious bidding strategies. In contrast, such a claim would be weaker if based only on evaluations on the malicious bidding found in the collected dataset, which has some limitations (i.e., the small size and mock conference setting). However, we note that the "dense subgraph"-based malicious bidding strategies that we analyze in this chapter are supported by the most common strategies found in the collected dataset.

An important contribution of our work is to provide direction for future work. Most clearly, future work can consider the problem of detection in richer-featured settings than just the binary bidding setting we consider here and demonstrate if detection is feasible in such settings. Some

examples of additional features that may be helpful are the strengths of bids, text-similarity scores, and past co-authorships. In these settings, future work can evaluate other types of anomaly-detection methods beyond the dense-subgraph discovery algorithms we consider. More ambitious detection algorithms could potentially attempt to leverage signals from the submitted reviews of colluders, such as reports from the other assigned reviewers that a review has some specified suspicious qualities. Such review-based features are critical if we want to additionally detect collusion rings formed after papers are assigned; this chapter does not attempt to address this more challenging problem (see Chapter 9). While developing effective detection algorithms is the ideal goal of this line of research, establishing negative results on the feasibility of detection in a broader setting would be very helpful for further directing the scope of future research on malicious bidding–if detection is hopeless even with more advanced algorithms, research ought to focus on mitigation-based techniques to address collusion rings. As most research so far has been focused on mitigation, our work also provides some direction for this line of research by establishing the region of colluding behavior that can be easily detected. Mitigation efforts need not be concerned with collusion in this region and can instead focus on mitigation within the undetectable region, particularly in the area where honest-reviewer groups do not exist.

Chapter 8

A Dataset on Manipulated Bidding for Reviewer-Author Collusion

In this chapter, we continue our investigation into the problem of reviewer-author collusion. As mentioned in Chapter 7, a significant challenge in solving this problem is that there is no dataset on which proposed algorithms can be evaluated and compared. Data from real conferences often cannot be released due to privacy concerns. More importantly, it is nearly impossible to claim for sure which reviewers were acting maliciously. Any public information about the aforementioned incidents of collusion that have been uncovered is kept vague. Researchers thus must rely on synthetic implementations of malicious behavior in order to test their algorithms, without any hard data on which to base the implementation. Such data is necessary in order to develop effective solutions to this important issue, despite the impossibility of collecting the ideal real-world dataset. In Chapter 7, we side-step the issue of a lack of data by intentionally working with a simplified bidding setting where malicious reviewer behavior can be easily parameterized, allowing us to analyze a wide range of potential malicious bidding strategies. Here, we tackle this problem head-on by gathering data directly on malicious reviewer behavior.

Several datasets containing conference bidding information are publicly available for research use. The PrefLib library [106] contains a few datasets with bidding data from real AI conferences. Wu et al. [159] provide a synthetic conference dataset including synthetic bidding data, constructed by analyzing the text and citations of a large set of recent AI papers. Xu et al. [160] also reconstruct similarities for the papers and reviewers at the 2018 International Conference on Learning Representations (ICLR), but this dataset contains only text-similarity scores and not bidding data. However, these datasets crucially lack labels of which reviewers are acting maliciously, or are constructed under the assumption that all reviewers act honestly. This necessitates researchers to implement any malicious behavior synthetically (as was done in Chapters 2 and 7 and in [159]). A few sources [99, 154] discuss specific incidents of bid manipulation in real conferences, but provide only high-level details and not a structured dataset. In contrast, our dataset contains data from human participants labeled with whether they were acting honestly or maliciously.

Similar problems of malicious behavior have been studied outside of the peer review setting. Fraudulent reviews are a major concern on platforms like Yelp and Amazon, spurring research on methods for fraud detection in these settings [2, 45, 90]. These settings notably differ from ours

in that reviewers are not assigned items to review: we focus on malicious behavior in the paper bidding phase, which has no analogue in product review settings. Within the crowdsourcing literature, detecting and mitigating malicious behavior by workers is the subject of some research, which often proposes using careful task design in addition to "gold standard" questions [42, 53]. While these techniques make low-effort responses more costly for malicious workers, the behavior of malicious reviewers in peer review is aimed specifically at manipulating the paper assignment rather than minimizing effort.

Our contributions:

1. We construct and publicly release a dataset containing bidding data from human participants in a mock conference setting (Section 8.1), taking the first step towards filling a critical gap in the research landscape on this important problem. To our knowledge, this is the first dataset of this kind available to other researchers. Participants were instructed to bid first as an honest reviewer and then as a malicious reviewer, so the data contains both honest bids and malicious bids with ground-truth labels on whether the behavior was malicious.
2. We supplement this dataset with descriptions of participants' behavior and a categorization of the strategies employed to manipulate the assignment (Section 8.2).
3. We empirically evaluate the success of participant strategies in terms of their ability to manipulate the assignment (Section 8.3.1).
4. We propose several simple detection algorithms, which can serve as baselines for other researchers aiming to develop algorithms to detect bid manipulation. We then evaluate the success of these algorithms at detecting different strategies (Section 8.3.2).
5. We synthetically scale up the dataset and provide additional large-scale evaluations (Section 8.4).

The dataset and our analysis code is publicly available at `https://github.com/sjecmen/malicious_bidding_dataset`.

8.1 Dataset

We first describe the data collection process, and then the contents of the collected data. Full documentation is given in Section 8.5.1.

8.1.1 Data Collection Process

This dataset was collected as part of an voluntary, ungraded activity conducted in a graduate-level course on artificial intelligence at Carnegie Mellon University, during the game theory component of the course. Participants were students enrolled in the course, primarily PhD students in computer science. Although our data is not from a real conference, this participant population is not very different from the population of reviewers in computer science conferences: for example, 33% of reviewers at the 2016 Conference on Neural Information Processing Systems (NeurIPS) were PhD students [136]. Potential participants were explicitly informed that "the activity is optional and won't affect your grade in any way" before consenting to participate. The full instructions given to participants are available in Section 8.5.2.

In the activity, participants act as reviewers during the paper bidding phase of a mock AI conference. Before the activity began, some setup was required. First, we constructed a list of 25

Subject area topic	# Papers	# Reviewers
Humans and AI	3	9
Social choice theory	3	11
Game theory	7	16
Probabilistic modeling	3	11
Search	3	7
Optimization	3	4
Machine learning	6	12

Table 8.1: Distribution of high-level subject area topics among the 28 papers and the 31 reviewers that completed the activity. Note that each reviewer can have up to 3 subject area topics.

AI topics to use as "subject areas" similar to those in real conferences; these subject areas were grouped into seven high-level "subject area topics". Potential participants (i.e., students in the class) were then polled to ask for their areas of interest among these subject areas. 56 out of 61 total students responded to this poll. Based on these responses, we constructed a list of 28 fake paper titles; these titles were chosen so that the distribution of paper subject areas would match the distribution of participant interest. In Table 8.1, we display the distribution of subject area topics for papers and for the subset of reviewers that completed the activity.

The next step in setup was to create "reviewer profiles" for each potential participant. We chose three subject areas for each participant as their areas of expertise as a reviewer, chosen to match their true areas of interest as much as possible. We chose one paper for each reviewer from within one of their subject areas as the paper authored by that reviewer; papers were chosen so that each paper was the authored paper of two reviewers.

Finally, we placed reviewers into colluding groups. We made six groups of size two, two groups of size three, and eight groups of size four, leaving six participants to act solo without a colluding group. Groups were chosen so that the subject areas of the reviewers in each group overlapped as much as possible and so that reviewers with the same authored paper were not in the same group. For the six reviewers without colluders, we assigned them each a target paper from one of their subject areas; this could represent colluding with an author reviewing at a different venue or targeting a paper with the intent to "torpedo review" [4, 16, 117]. Each participant was provided with all information in their reviewer profile before beginning the activity, including contact information for the other participants in their colluding group.

The activity took place in two phases, so that each participant could act as an honest reviewer in the first phase and as a malicious reviewer in the second phase. In each phase of the activity, each participant was presented with the list of 28 paper titles and asked to submit a bid on each one. Bids were chosen from the options "Not willing to review", "Indifferent", and "Eager to review." The bidding interface is shown in Figure 8.1. After bids were placed, they could provide text responses to questions regarding their strategy. Participants were told that bids would be used to determine a paper assignment, where each paper would be assigned three reviewers and each reviewer would be assigned three papers.

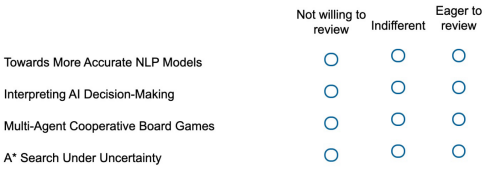


Figure 8.1: Illustration of the participant bidding interface.

- In phase one, participants were instructed to bid on papers "according to your own personal interests, as if you were actually going to review the assigned papers."
- In phase two, participants were instructed to work with their groups to manipulate the paper assignment. Specifically, participants were instructed to bid "so that you are assigned to review each other's papers." Participants were also instructed to coordinate their strategy with their groups and were free to use any method of communication to do so. Reviewers without a group were instructed to bid in order to get assigned to their given target paper. Reviewers were also told that conference program chairs were investigating the bidding for suspicious behavior: "If they notice any reviewers bidding suspiciously, they can manually modify the assignment to their liking. For example, the PCs may look through the bids to notice any reviewers that bid positively only on a single paper and choose to ignore those bids."

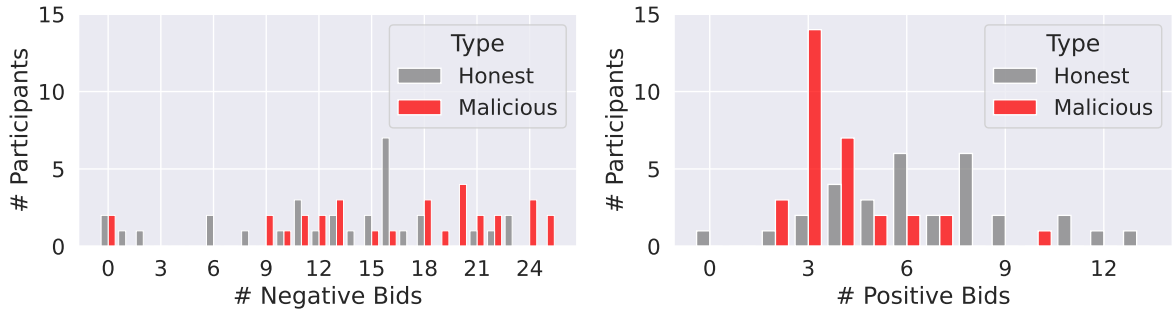
In what follows, we use the terms "honest reviewer" and "malicious reviewer" to refer to participants acting in the respective role.

8.1.2 Dataset Contents

Each of the 56 participants was given a "reviewer profile" as described above, consisting of three subject areas and an authored paper. Each profile also specified the participant's group, as well as a target paper if they had no colluders. We henceforth will also use the term "target papers" to refer to the papers authored by a reviewer's fellow group members. Of the 56 participants, we received 35 responses to phase one of the activity (the honest bidding) and 31 responses to phase two (the malicious bidding). In each phase, each response includes a set of 28 bids (one per paper) and a few text responses to short-answer questions asked after the bidding. Bids were either "Not willing to review", "Indifferent", or "Eager to review"; a missing bid on a paper was interpreted as an "Indifferent" bid. We henceforth refer to these as "negative", "neutral", and "positive" bids respectively. In both phases, participants were asked: "Did you follow any kind of strategy when bidding and if so, what was it?" In phase two, participants were additionally asked "Did you communicate with your other group members and if so, what did you discuss?" and "Do you have any thoughts on how to prevent this kind of malicious behavior in conferences?" The dataset was de-identified by course instructors.

8.2 Description of Bidding Behavior

In this section, we provide descriptions of the collected bidding data. We first quantitatively consider the bidding data itself, and then qualitatively analyze the strategy descriptions provided by participants.



(a) Negative bids

(b) Positive bids

Figure 8.2: Distributions of positive and negative bids.

8.2.1 Quantitative Description

In Figure 8.2, we compare the distributions of positive and negative bids between the honest reviewers and the malicious reviewers. Each bar corresponds to a specific number of positive or negative bids, with the height of the bar indicating the number of honest or malicious reviewers submitting that many bids. In Figure 8.2a, we see that malicious reviewers generally provided more negative bids than honest reviewers, although both honest and malicious reviewers provided high numbers of negative bids. We see in Figure 8.2b that honest reviewers generally gave more positive bids than malicious reviewers. Nearly half of the malicious reviewers (14 responses) bid positively on exactly three papers (the number of papers they will be assigned), whereas honest reviewers were much more likely to bid on additional papers.

8.2.2 Qualitative Description

During both phases of the activity, participants were asked to describe any strategy they used in a text response. We analyze these responses in conjunction with the actual provided bids in order to determine what strategy participants implemented. These strategies can be used by researchers to realistically "scale up" the dataset, as we demonstrate in Section 8.4.

When providing honest bids, almost everyone (32 responses) specified their strategy as some form of "bidding based on my interests." A few of the comments were more detailed about an additional strategy they followed to get assigned high-expertise papers (e.g., bid positively on exactly three papers), but none of these were common.

For the malicious bids, we categorize them into five broad "strategies." Below we describe the prototypical aspects of each strategy, although participants' actual implementations varied.

1. *Basic:* On target papers, bid mostly positively and sometimes neutral. On non-target papers within the reviewer's subject area, bid mostly neutral with a few positive bids. (11 people)
2. *Negative-in-area:* As in *Basic*, but bid mostly negatively on non-target papers within the reviewer's subject area (still with a few positive bids). (9 people)
3. *Overlap:* As in *Basic*, but coordinate the bids on non-target papers with other group members. Specifically, all group members bid positively on the same set of non-target, in-area papers (with the aim of "overloading" them so that not all group members can be assigned to them). (3 people)

4. *Cycle:* As in *Basic*, but bid in a particular manner on target papers. Specifically, each group member bids positively on one other member's target paper to create a cycle of positive bids. Group members bid neutral or negative on other target papers. (4 people)
5. *Popularity:* As in *Basic*, but choose bids on non-target, in-area papers based on perceived paper "popularity". Specifically, bid positively on a small number of non-target papers that are likely to be unpopular among other reviewers (with the aim of being assigned to these in addition to the target papers). Bid neutral on non-target papers that are likely to be popular among other reviewers (with the aim of not being assigned to them). (2 people)

Two other participants did not describe or implement an understandable strategy. Additionally, 19 of these responses specified that they coordinated with their group in forming their strategy.

Our choice of strategy categorization is not unique, as participant strategies could further be broken down on the basis of additional information. Some strategies specified how they chose the number of positive bids, usually in consideration of the fact that each reviewer would be assigned three papers. Some strategies specified how they chose which non-target papers to bid positively on (e.g., only those outside their subject area). Some strategies bid positively on all target papers while others split between positive and neutral. We choose to focus on the above categorization, leaving analysis on the basis of these additional factors for future work.

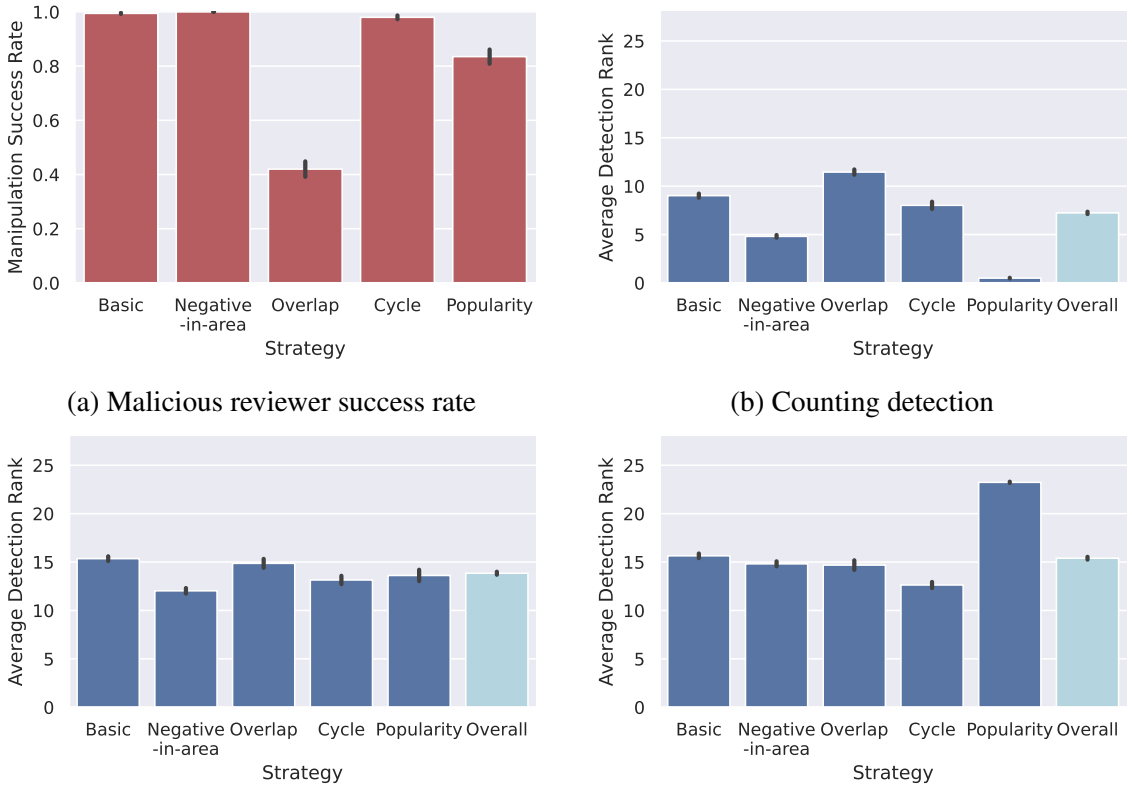
8.3 Evaluation of Bidding Behavior

In this section, we analyze the performance of malicious reviewers empirically in terms of successfully manipulating the assignment and avoiding detection. We also consider the performance of several baseline detection algorithms. Specifically, we run two empirical evaluations: one which examines how successful each reviewer is at manipulating the assignment, and one which examines how well each reviewer avoids detection by simple detection algorithms. We run multiple trials of each evaluation, where each trial considers one group of malicious reviewers. In each trial, we construct a full set of reviewers by taking the malicious reviewers in the group under consideration and adding honest reviewers at random until the number of reviewers equals the number of papers (28). We then use this set of reviewers along with the fixed set of 28 papers for the experiment. In total, we run 100 trials of each evaluation for each group, aggregating results over the 100 trials.

8.3.1 Manipulation Success Evaluation

Malicious reviewers were instructed to bid in order to get assigned to their target papers, but were not given a specific numerical objective to optimize. Therefore, some participants may have divided the target papers among their group while others simultaneously targeted all target papers. In our analysis, we define a "successful manipulation" as when a malicious reviewer is assigned to at least one of their target papers. This definition is reasonable for different forms of strategic coordination within groups and is robust to the non-participation of a reviewer's group members.

We compute similarities $S_{r,p} \in \mathbb{R}$ between each reviewer r and paper p as follows. We first compute a subject area score $K_{r,p}$ by comparing the reviewer subject areas and paper subject areas. If the paper's subject area matches one of the reviewer's subject areas, we set $K_{r,p} = 1$. Otherwise, if the paper's high-level subject area topic matches one of the reviewer's, we set



(a) Malicious reviewer success rate

(b) Counting detection

(c) Ring detection

(d) Low-rank detection

Figure 8.3: Average success rate of manipulation strategies (Figure 8.3a) and average rank of malicious reviewers under different detection algorithms (Figures 8.3b-8.3d). In Figures 8.3b-8.3d, a lower rank value indicates that the algorithm has marked the malicious reviewer as more suspicious.

$K_{r,p} = 0.5$. Else, we set $K_{r,p} = 0$. Bid values $B_{r,p} \in \{-1, 0, 1\}$ are set corresponding to negative, neutral, and positive bids respectively. Final similarities are then computed as $S_{r,p} = (1 + K_{r,p})2^{B_{r,p}}$, modeled off of the similarity formula used for NeurIPS 2016 [136].

After computing similarities, we then find a maximum-similarity assignment of reviewers to papers [29, 30, 52, 59, 85], subject to constraints that each paper is assigned three reviewers, each reviewer is assigned to three papers, and no reviewer is assigned to a paper they authored. This method of assigning reviewers to papers is standard in modern conferences [134, Section 3]. In the resulting assignment, we determine whether each malicious reviewer was successfully assigned to at least one of their target papers.

In Figure 8.3a, we display the results from this empirical evaluation. Each bar represents the average success rate of all reviewers with a given strategy, with error bars representing standard error of the mean. The two strategies used by the greatest number of reviewers (*Basic* and *Negative-in-area*) had a perfect success rate, indicating that the additional sophistication of the *Negative-in-area* strategy was not necessary to secure the target assignments. The *Cycle* strategy was also highly successful, likely due to its similarity to *Basic*. The *Overlap* strategy had a very

low success rate despite also being very similar to the *Basic* strategy; we suspect this was due to poor implementation of the strategy, since only a few participants attempted this strategy. The *Popularity* strategy had mixed success, perhaps due to the difficulty of correctly guessing which papers are popular.

8.3.2 Detection Evaluation

We consider three simple detection algorithms as baselines for analysis. Each algorithm outputs a ranking of all reviewers, where reviewers higher in the ranking are deemed more likely to be malicious. Before running any algorithm, bids made by each reviewer on their own paper were discarded.

- *Counting Detection*: Compute a score for each reviewer equal to the number of negative bids minus the number of positive bids. Output the ranking of all reviewers by score (where higher scores are higher-ranked).
- *(Pairwise) Ring Detection*: We first construct a ranking of all pairs of reviewers. Pairs in which both reviewers bid positively on each others' papers are ranked the highest, followed by pairs in which exactly one reviewer bid positively on the other's paper, followed by pairs in which neither reviewer bid positively on the other's paper. Within each of these categories, we rank the pairs of reviewers as follows. For each reviewer in the pair, count the number of negative bids made and subtract the number of positive bids, not counting the bid made on the paired reviewer's paper. Rank the pairs within each category by the total score for the two reviewers in the pair. Finally, output a ranking of reviewers by their highest position in the ranking of pairs (breaking ties arbitrarily).
- *Low-Rank Detection*: This algorithm leverages the assumption that the bidding matrix B (with entries $B_{r,p}$ defined in Section 8.3.1) is approximately low-rank[1]. Compute a low-rank approximation L to B. Compute a score for each reviewer r as $\sum_{p \in \text{Papers}} |B_{r,p} - L_{r,p}|$ (where the sum is taken over all papers). Output the ranking of all reviewers by this score. Intuitively, this method classifies bids as anomalous when they cannot be represented well in a low-rank matrix. In our experiments, we use a rank-3 approximation; this number was chosen heuristically based on inspection of the singular values of several sampled honest bidding matrices.

In each trial of this evaluation, we select a set of reviewers as described at the beginning of Section 8.3 and collect the bids of each selected honest or malicious reviewer. We then run a detection algorithm on these bids and report the output rank of each malicious reviewer.

In Figures 8.3b-8.3d, we display the average ranks of malicious reviewers output by the detection algorithms, with error bars representing standard error of the mean. Rank values (on the y-axis) indicate the number of other reviewers marked as more suspicious than the malicious reviewer, ranging between 0 (most suspicious) and 27 (least suspicious); a lower rank value indicates that the algorithm was more successful at detecting the manipulation. In each plot, the five leftmost bars display the average ranks of malicious reviewers using each strategy under a given detection algorithm. Between the two most popular strategies (*Basic* and *Negative-in-area*), all three detection algorithms were better at detecting the *Negative-in-area*. *Negative-in-*

[1]Fiez et al. [49] heuristically show that the ICLR 2018 similarity matrix [160] (2435 reviewers, 935 papers) is approximately rank-10. In Chapter 4, we leverage the low-rank structure of similarities to design algorithms for two-stage paper review processes.

area was particularly well-detected by the *Counting Detection* algorithm, since it specifically targets the kind of behavior done by the *Negative-in-area* strategy. *Counting Detection* also does very well against the *Popularity* strategy, although this is simply because both participants implementing this strategy happened to bid negatively on nearly all papers outside of their subject area. The rightmost bar in each plot shows the overall performance of the detection algorithms by averaging the output ranks across all malicious reviewers. *Counting Detection* has decent performance overall, averaging ranking malicious reviewers around the 25th percentile. The other algorithms do poorly, averaging ranking malicious reviewers around the 50th percentile (essentially no better than random). Some malicious reviewers consciously avoided seeming "too connected" to their group by bidding neutral on some target papers, hurting the performance of *Ring Detection*; this algorithm may also have suffered from the small size of the dataset, since pairs of honest reviewers are likely to bid positively on each others' papers by chance.

8.4 Analysis of Synthetically Scaled-up Data

In this section, we run experiments on synthetically scaled-up versions of the data. We first describe the procedure we use to scale up the data, followed by the experimental results.

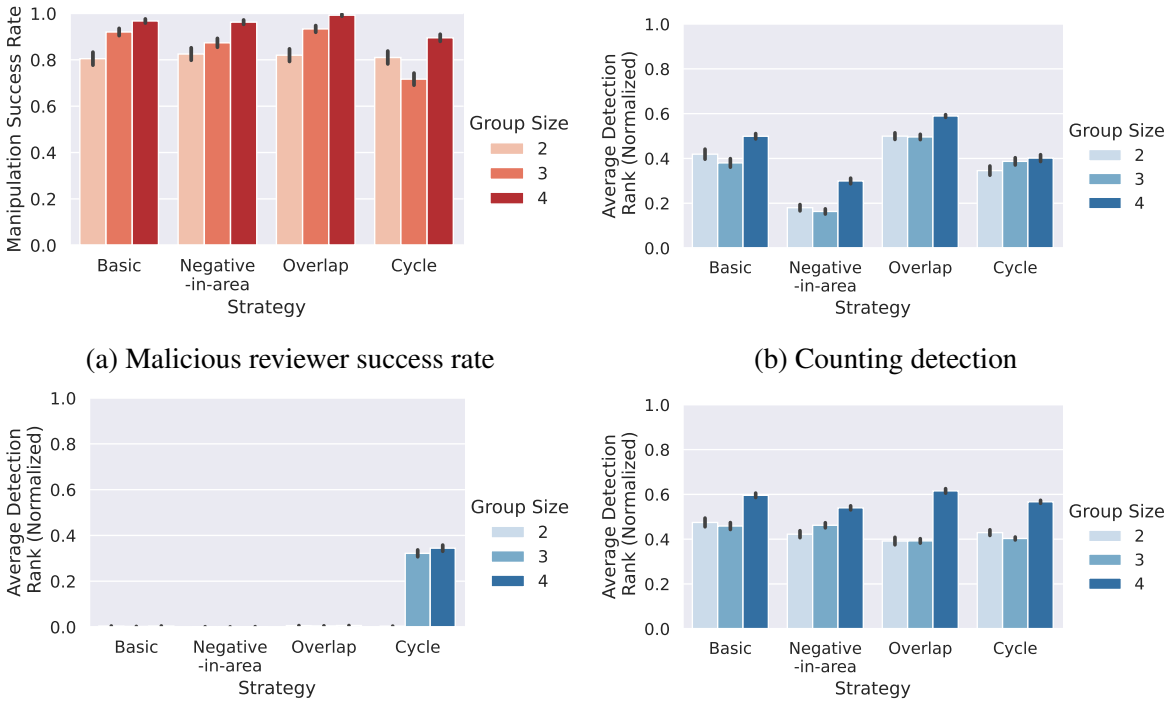
8.4.1 Synthetic Dataset Construction

We construct a large-scale synthetic dataset based on characteristics of our collected data. Using the strategy categorization introduced in Section 8.2.2, we wrote our own implementation of each strategy (other than *Popularity*), modeling any bidding behavior not specified by the strategy after a random reviewer in the original dataset. We remark that this procedure is only one example of how our dataset can be used to inform the creation of larger-scale synthetic datasets and is far from the only way to do so.

We first choose the number of reviewers and papers in the scaled-up data, the malicious group size (2, 3, or 4), and the strategy that the malicious reviewers will employ. For simplicity, authorship will be one-to-one between papers and reviewers, so each reviewer corresponds to one authored paper.

We then construct the reviewer and paper subject areas. To determine the subject areas of the malicious reviewers and the papers authored by them, we randomly choose a malicious group of the chosen size from the original data. We then copy the subject areas of these reviewers and their authored papers. Similarly, to determine subject areas for each honest reviewer and their authored paper, we randomly choose any reviewer from the original data and copy the subject areas of the reviewer and their authored paper.

We next construct the bids for each honest reviewer. For each honest reviewer, we randomly choose an honest reviewer in the original data to use as a "model". We count the number of positive bids made by this original reviewer on papers within their high-level subject area topics, and add this many positive bids for the new reviewer on random papers within their subject area topics. We do the same for positive bids on papers outside their subject area topics, counting the number of bids made by the original reviewer and adding this many at random for the new reviewer. We then repeat this process for negative bids, but scale up the number of negative bids added by the ratio between the new and old numbers of papers. For example, if the original reviewer made two negative bids on papers within their subject area topic and we are scaling up



(a) Malicious reviewer success rate

(b) Counting detection

(c) Ring detection

(d) Low-rank detection

Figure 8.4: Results from synthetic scaled-up experiments with 5000 reviewers and papers. Different colors indicate different malicious group sizes. In Figures 8.4b-8.4d, a lower normalized rank indicates the malicious reviewer was detected as more suspicious.

from 28 papers to 280 papers, we would add 20 negative bids on papers within the new reviewer's subject area topics. We choose to scale up the number of negative bids and not the number of positive bids because the reviewer loads are still three in the new experiments despite the larger number of papers. Many reviewers considered the reviewer loads in choosing how many positive bids to place (e.g., by bidding positively on exactly three papers), and this procedure preserves that behavior. In contrast, many reviewers bid negatively on a large number of papers, suggesting that they would place even more negative bids if the number of papers was increased.

Finally, we construct bids for the malicious reviewers. For each malicious reviewer, we first construct bids on all non-target papers using the method described in the previous paragraph for honest reviewers. However, rather than randomly choosing a model reviewer from among the honest reviewers, we randomly choose a model reviewer from among the malicious reviewers with the same strategy. We then modify the bids in a different way depending on the strategy chosen. For the *Basic* strategy, we simply add a positive bid on each target paper. For the *Overlap* strategy, we also add a positive bid on each target paper; we then further adjust the bids of all malicious reviewers so that the positive bids are on the same set of papers. For the *Cycle* strategy, we have each reviewer bid positively on only one target paper and neutral on the others, constructing a cycle. We do not implement the *Popularity* strategy due to its rarity and the difficulty of modeling how a reviewer might predict which papers are popular.

8.4.2 Synthetic Results

We run the experiments described in Section 8.3 on the scaled-up bids and subject areas, aggregating results over 100 trials of synthetic dataset construction for each setting.

In Figure 8.4, we display results from both experiments for each malicious reviewer strategy when the data is scaled up to 5000 papers and reviewers, varying the malicious group size. In Figure 8.4a, we see that all four implemented strategies are very successful at all group sizes. The *Cycle* strategy is slightly less successful than the others; this is perhaps because by bidding positively on only one target paper, the strategy is less robust to the many honest reviewers also bidding on the target papers. All strategies are most successful for the largest malicious group size, likely because these groups have more target papers that can be assigned.

In Figures 8.4b-8.4d, we display the detection ranks output by the three detection algorithms. The rank values (on the y-axis) are normalized by the number of reviewers so that they range from 0 (most suspicious) to 1 (least suspicious). In Figure 8.4b, we see that the *Counting Detection* algorithm does moderately well at detecting the *Negative-in-area* and *Cycle* strategies, although it does relatively worse on all strategies as compared to the original data. This may be because both malicious and honest reviewers make relatively more negative bids than positive bids in the scaled-up data, and so the algorithm must look for a weaker signal in a larger set of reviewers than in the original data. In contrast, Figure 8.4c shows that the *Ring Detection* algorithm does extremely well at detecting malicious behavior. This is because our implementations of the *Basic*, *Negative-in-area*, and *Overlap* strategies bid positively on all target papers, and so the detection of these clusters cuts through the noise of the many honest reviewers. *Cycle* avoids detection by this algorithm when the malicious group size is greater than 2, since these malicious reviewers avoid forming pairwise rings of positive bids. Figure 8.4d shows that the *Low-Rank Detection* algorithm performs poorly, as in the original data.

We also run additional experiments varying the total number of reviewers and papers, which we present in Section 8.5.3.

8.5 Supplemental Material

In this section, we present additional details and experimental results omitted from the previous sections.

8.5.1 Dataset Documentation

The dataset and our analysis code can be found at `https://github.com/sjecmen/malicious_bidding_dataset`. This dataset is licensed under a CC BY 4.0 license.[2]. This work was conducted under the approval of the Carnegie Mellon University IRB. This dataset is intended for use by other researchers, specifically on the topic of addressing malicious behavior in peer review. See Section 8.1.1 for a detailed description of the data collection process.

The dataset consists of 2 text files and 4 CSV files. The two text files respectively list the subject areas and paper titles used in the activity. Below, we describe the format of the CSV files.

- The file ‘setup.csv‘ contains the reviewer profile information, in the following columns:
  - **name**: Anonymized string ID for each potential participant.

[2] `https://creativecommons.org/licenses/by/4.0/`

- **sas**: Three space-separated integers, indicating the indices of the subject areas for this reviewer.
- **authored_sa**: Subject area index of the paper authored by this reviewer.
- **authored_id**: Paper title index of the paper authored by this reviewer.
- **target_sa**: Subject area index of the target paper for this reviewer (if no colluders).
- **target_id**: Paper title index of the target paper for this reviewer (if no colluders).
- **group**: Integer ID for the reviewer's group of colluders.

- The file 'honest_bidding.csv' contains the responses to the first phase of the activity on honest bidding. The **Name** column contains the participant ID for each response. The remaining columns indicate responses to the questions stated in the second header row.
- The file 'malicious_bidding.csv' contains the responses to the second phase of the activity on malicious bidding, formatted in the same way.
- The file 'strategy_annotations.csv' contains our categorization of participant responses by strategy. The **Name** column contains the participant ID. The **Strategy** column contains an integer indicating the strategy, as an index into the strategy list [Basic, Negative-in-area, Overlap, Cycle, Popularity]; an entry of -1 indicates no strategy could be discerned. The **Discussed** column contains an entry in {Y, N} indicating whether the participant discussed their strategy with their colluders, with an empty entry indicating an unclear response.

8.5.2 Participant Instructions

The participants were first verbally told about the problem of bid manipulation, along with a brief description of the activity. They were also told that participation is optional and ungraded. Some motivations to participate were stated: to get a hands-on experience in game-theoretic thinking, to help the community understand what kinds of bidding manipulation may be possible, and to experience what may be a fun exercise.

The participants were subsequently provided written instructions, reproduced below. The text in brackets differed between participants.

Before beginning the activity: *As mentioned in class, we are running a fun game to give you a hands-on experience with game theory. This fun game is about strategic behavior in paper bidding for an academic conference. The activity is optional and won't affect your grade in any way.*

You (along with your classmates) will play the role of a reviewer for a fictional conference called FAIC (the Fake AI Conference). To determine which papers you should review, FAIC is asking you to "bid" on various papers. See the slides for more details. The activity has two parts:

Part 1: You play the role of an honest reviewer and submit bids at FAIC according to your interests. Complete this part using [this personalized link].

Part 2: You play the role of a manipulative reviewer who wants to manipulate the assignment algorithm. You are working with a group of friends to do these manipulations: [group emails]. For this part, please read the instructions, discuss your strategy with your group, and then return to complete the activity. View this part using [this personalized link].

Before the first phase (honest reviewing): *In this activity, you (along with your classmates) will be playing the role of a reviewer for a fictional conference called FAIC (the Fake AI Conference). FAIC is currently attempting to determine which papers each reviewer should be assigned to review based on their expertise and interests.*

As a reviewer, your expertise is in the areas of [subject areas]. This means that you will be more likely to be assigned to papers matching this description. You are also an author on the paper [paper title] which you have submitted to FAIC.

In order to further determine which papers you should be assigned to review, FAIC is asking you for your level of interest in each paper, commonly known as "bidding". FAIC will then take the bids into account when assigning papers to reviewers.

Suppose that you are an honest reviewer at FAIC. This means that you should bid on papers according to your own personal interests, as if you were actually going to review the assigned papers. Keep in mind that each paper will be assigned 3 reviewers, and each reviewer will be assigned to at most 3 papers.

Before the second phase (malicious reviewing): *Now, you will take the role of a malicious reviewer at FAIC. Such malicious reviewers work in groups with their friends with the goal of getting assigned to each other's papers. Here is an example strategy that a pair of malicious reviewers working together might use: [image depicting two reviewers bidding positively on each other's paper and negatively on all others].*

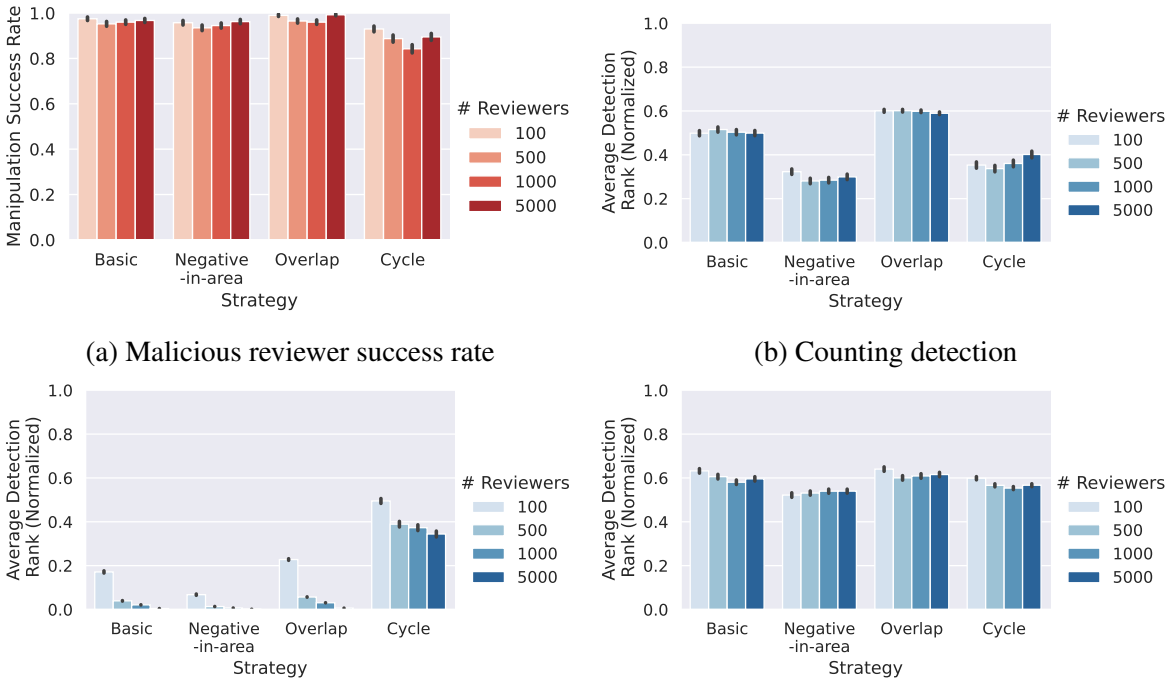
The program chairs (PCs) organizing FAIC are aware that such manipulations can occur. If they notice any reviewers bidding suspiciously, they can manually modify the assignment to their liking. For example, the PCs may look through the bids to notice any reviewers that bid positively only on a single paper and choose to ignore those bids [image with example of such a malicious reviewer being detected]. As a malicious reviewer, you should be aware that your bidding manipulation may be detected.

To improve your paper's chances of acceptance, you are working with your friends who have authored the papers [paper names]. Recall that you are an author on the paper [paper title] which you have submitted to FAIC. All of you are experts in [subject area].

Your goal is to strategically coordinate with your group as a team to bid so that you are assigned to review each other's papers. You should communicate with them to discuss your bidding strategy (you can leave and return to this page at any time). Keep in mind that each paper will be assigned 3 reviewers, and each reviewer will be assigned to at most 3 papers. The more reviewers from within your group assigned to each paper, the higher that paper's chances of acceptance are (which is good for your group).

8.5.3 Additional Synthetic Results

In Figure 8.5, we display the results of additional scaled-up experiments (described in Section 8.4). In these experiments, we fix the malicious group size at 4 and vary the total number of reviewers and papers (held equal) between 100 and 5000. In Figures 8.5b-8.5d, the rank values (on the y-axis) are normalized by the number of reviewers so that they range from 0 (most suspicious) to 1 (least suspicious). These results show generally that within this range, the performance of the malicious reviewer strategies and detection algorithms is not affected by the size of the data. One exception is that in Figure 8.5c, we see that the *Ring Detection* algorithm does better as the number of reviewers and papers increases. This fits with the intuition that the



(a) Malicious reviewer success rate

(b) Counting detection

(c) Ring detection

(d) Low-rank detection

Figure 8.5: Results from synthetic scaled-up experiments with a malicious group size of 4. Different colors indicate different numbers of reviewers and papers. In Figures 8.5b-8.5d, a lower normalized rank indicates the malicious reviewer was detected as more suspicious.

detected rings stand out more among larger numbers of honest reviewers, which are unlikely to form rings.

8.6 Discussion

In this chapter, we construct and release a dataset on malicious paper bidding, along with our analysis of the behavior employed by participants. We also evaluate the effectiveness of various participant strategies and detection algorithms. Our dataset has been de-identified, and furthermore the risk to participants in the event of any re-identification is low since the dataset includes no sensitive information.

One major limitation of our work is that our dataset is from a mock conference setting and may not be perfectly representative of real-world behavior. Thus, in future work, our dataset should be used as just one method of evaluation alongside others. Any proposed detection algorithm should at least be effective against the strategies identified here, but good performance on our dataset alone is not sufficient to show an algorithm's effectiveness in practice. Another possible limitation of our work is that malicious reviewers could use the data we provide and any future research on it to improve their strategies. Those researching defenses against malicious behavior should consider that an adversary can adapt in response to a new defense and develop methods that are robust to adversarial changes in behavior.

Our dataset may be useful within various directions of future work that aim to address mali-

cious behavior in peer review. First, our work considers three algorithms for detecting malicious bidding, which we intentionally choose as very simple baselines. The vast literature on anomaly detection proposes many more complex techniques that could be adapted for our setting, as we attempt to do in Chapter 7. In addition to new techniques for detecting malicious bids, new algorithms for mitigating the impact of malicious bids on the paper assignment (e.g., Chapter 2, [159]) can be developed and evaluated using our dataset. Additionally, as more techniques to address malicious behavior are proposed and deployed (see Chapter 6), a valuable goal for future work is to provide guidance to conference program chairs about which techniques they should deploy at their venue. For example, as one approach, the data and strategies we present could be analyzed in a game-theoretic framework to identify the optimal defensive strategy for program chairs to deploy against an adversarial group of malicious reviewers. Finally, there is a clear opening for future work to address the limitations of our dataset by collecting data on malicious reviewer bidding at a larger scale, or potentially by working with a real conference to conduct a similar experiment. The collection of such data, while an enormously difficult undertaking, would expand the possibilities for detection-based approaches to addressing reviewer-author collusion.

Chapter 9

Conclusion

In this thesis, I have presented several solutions to problems of undesirable behavior that plague conference peer review processes. Randomized paper assignments are a simple and practical method for mitigating the impact of reviewer-author collusion (as well as the problems of torpedo reviewing and reviewer de-anonymization). As a result, our algorithms have already been deployed in major computer-science conferences. The use of these randomized assignments unlocks the potential of our off-policy evaluation methods, which can be used to evaluate the quality of alternative paper assignment policies and thus improve the review quality of future conferences. The damage caused by missing or low-effort reviewers can be mitigated via the use of a two-phase review process, and we present empirical and theoretical evidence to support the use of a simple "random split" algorithm for finding high-quality paper assignments in this setting. To address strategic reviewing, we introduce algorithms that find strategyproof assignments with quality guarantees. Our work also compares various approaches to mitigating manipulation of paper assignments, provides empirical analysis on the feasibility of detecting reviewer-author collusion from bidding, and releases a dataset on malicious bidding in reviewer-author collusion.

The work in this thesis has already had impact on the peer review landscape within computer science. Specifically, our algorithm for finding randomized paper assignments introduced in Chapter 2 has been used by the large 2022 and 2023 AAAI Conferences on Artificial Intelligence and the 2023 ACM Conference on Knowledge Discovery and Data Mining (among other venues), and is available for future conferences to use on the `OpenReview.net` peer-review platform. The code for the algorithms proposed in all chapters is publicly available, and we hope that conferences will continue to adopt these algorithms in the future. Our work has also been recognized by some awards at conferences and workshops: the work [74] (Chapter 4) received a "Best Paper" Honorable Mention from the 2022 AAAI Conference on Human Computation and Crowdsourcing , and the work [73] (Chapter 6) received an "Outstanding Paper" Award at the Machine Learning Evaluation Standards Workshop at ICLR 2022.

While our work addresses some problems in conference peer review, the issues we consider are far from being entirely solved. One major objective of this line of research is to have impact in the actual venues conducting peer review. Towards this end, many of the individual chapters in this dissertation have room to be extended in future work: for example, to incorporate additional constraints on the paper assignment that may be desired by program chairs, to remove overly simplistic assumptions, or to provide more accurate evaluations on real conference data or as part

of trial deployments. In addition, our work on detection-based approaches to reviewer-author collusion leaves many questions open. Our analysis on the feasibility of detecting reviewer-author collusion in Chapter 7 focuses on a simplified setting with only binary bidding data, but future work in this direction may show that our pessimistic results do not carry over to more complex settings.

Furthermore, undesirable behavior in peer review extends beyond the forms that we identify in this thesis. This is particularly true when discussing issues of malicious behavior, as malicious agents will try their best to adapt their attacks on the peer review process in order to circumvent any defenses that are implemented. As one example, the work in this thesis attempts to prevent reviewer-author collusion by adding defenses against manipulation of the paper assignment, such as randomized paper assignments and bidding-based detection methods. However, in one recent conference, malicious reviewers instead attempted to set up collusion rings after the paper assignment phase. These reviewers posted the IDs of their assigned papers on external websites in an attempt to find authors they could collude with. This kind of post-assignment collusion is not addressed by our proposed methods and thus requires entirely new solutions. More generally, the "cat-and-mouse" game between publication venues and bad actors in peer review will necessitate continued effort from researchers to fortify these peer review systems. This thesis represents just one step towards improving the institution of peer review, something that all members of the scientific community have a stake in.

Bibliography

[1] Anastasia Ailamaki, Periklis Chrysogelos, Amol Deshpande, and Tim Kraska. The SIGMOD 2019 research track reviewing system. *ACM SIGMOD Record*, 48(2):47–54, 2019. 4

[2] Leman Akoglu, Rishi Chandy, and Christos Faloutsos. Opinion fraud detection in online reviews by network effects. *Proceedings of the International AAAI Conference on Web and Social Media*, 2013. 7, 8

[3] Leman Akoglu, Hanghang Tong, and Danai Koutra. Graph based anomaly detection and description: A survey. *Data Mining and Knowledge Discovery*, 29(3):626–688, 2015. 7

[4] Jef Akst. I hate your paper. Many say the peer review system is broken. Here's how some journals are trying to fix it. *The Scientist*, 24(8):36, 2010. 8.1.1

[5] Noga Alon, Felix Fischer, Ariel Procaccia, and Moshe Tennenholtz. Sum of us: Strategyproof selection from the selectors. In *Conference on Theoretical Aspects of Rationality and Knowledge*, pages 101–110, 2011. 5

[6] Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, et al. Construction of the literature graph in semantic scholar. *arXiv preprint arXiv:1805.02262*, 2018. 7.1.2

[7] Joshua D Angrist, Parag A Pathak, and Christopher R Walters. Explaining charter school effectiveness. *American Economic Journal: Applied Economics*, 5(4):1–27, 2013. 3.7

[8] J. Scott Armstrong. Unintelligible management research and academic prestige. *Interfaces*, 10(2):80–86, 1980. 4

[9] Ines Arous, Jie Yang, Mourad Khayati, and Philippe Cudré-Mauroux. Peer grading the peer reviews: A dual-role approach for lightening the scholarly paper review process. In *Proceedings of the Web Conference 2021*, pages 1916–1927, 2021. 3.1

[10] Itai Ashlagi, Felix Fischer, Ian A. Kash, and Ariel D. Procaccia. Mix and match: A strategyproof mechanism for multi-hospital kidney exchange. *Games and Economic Behavior*, 91:284–296, 2015. 5

[11] Susan Athey, Dean Eckles, and Guido W. Imbens. Exact p-values for network interference. *Journal of the American Statistical Association*, 113(521):230–240, 2018. 3.6.2

[12] Haris Aziz, Omer Lev, Nicholas Mattei, Jeffrey S. Rosenschein, and Toby Walsh. Strategyproof peer selection: Mechanisms, analyses, and experiments. In *Proceedings of the*

AAAI Conference on Artificial Intelligence, volume 30, 2016. 5, 5.1.1

[13] Haris Aziz, Omer Lev, Nicholas Mattei, Jeffrey S. Rosenschein, and Toby Walsh. Strategyproof peer selection using randomization, partitioning, and apportionment. *Artificial Intelligence*, 275:295–309, 2019. 5

[14] Stefano Balietti, Robert L Goldstone, and Dirk Helbing. Peer review and competition in the art exhibition game. *Proceedings of the National Academy of Sciences*, 113(30): 8414–8419, 2016. 1.2, 5

[15] Richard E Barlow and Hugh D Brunk. The isotonic regression problem and its dual. *Journal of the American Statistical Association*, 67(337):140–147, 1972. 3.3.4

[16] Edward F. Barroga. Safeguarding the integrity of science communication by restraining 'rational cheating' in peer review. *Journal of Korean Medical Science*, 29(11):1450–1452, 2014. 1.2, 8.1.1

[17] Alina Beygelzimer, Yann Dauphin, Percy Liang, and Jennifer Wortman Vaughan. The NeurIPS 2021 consistency experiment. `https://blog.neurips.cc/2021/12/08/the-neurips-2021-consistency-experiment/`, 2021. Accessed May 17, 2023. 3, 4

[18] Eric Blais and Ryan O'Donnell. Lower bounds for testing function isomorphism. In *IEEE 25th Annual Conference on Computational Complexity*, pages 235–246. IEEE, 2010. 4.7

[19] Niclas Boehmer, Robert Bredereck, and André Nichterlein. Combating collusion rings is hard but possible. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4843–4850, 2022. 6.2.3, 7, 7.2

[20] Anna Bogomolnaia and Hervé Moulin. A new solution to the random assignment problem. *Journal of Economic Theory*, 100(2):295–328, 2001. 2

[21] Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X Charles, D Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research*, 14(11), 2013. 3, 3.7

[22] Nicolas Bousquet, Sergey Norin, and Adrian Vetta. A near-optimal mechanism for impartial selection. In *International Conference on Web and Internet Economics*, pages 133–146. Springer, 2014. 5, 5.1.1

[23] Brian Brubach, Karthik Abinav Sankararaman, Aravind Srinivasan, and Pan Xu. New algorithms, better bounds, and a novel model for online stochastic matching. In *24th Annual European Symposium on Algorithms*, 2016. 4

[24] Niv Buchbinder, Moran Feldman, Joseph Naor, and Roy Schwartz. Submodular maximization with cardinality constraints. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1433–1452. SIAM, 2014. 4.6.3

[25] Eric Budish, Yeon-Koo Che, Fuhito Kojima, and Paul Milgrom. Implementing random assignments: A generalization of the Birkhoff-von Neumann theorem. In *Cowles Summer Conference*, 2009. 2.2.1, 2.2.2, 2.5.3, 2.5.3, 2.2, 4.7, 4.7

[26] Guillaume Cabanac and Thomas Preuss. Capitalizing on order effects in the bids of peer-

reviewed conferences to secure reviews by expert referees. *Journal of the American Society for Information Science and Technology*, 64(2):405–415, 2013. 6.1

[27] Stephen J. Ceci and Douglas P. Peters. Peer review: A study of reliability. *Change: The Magazine of Higher Learning*, 14(6):44–48, 1982. 4

[28] Moses Charikar. Greedy approximation algorithms for finding dense components in a graph. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 84–95. Springer, 2000. 7.2.2

[29] Laurent Charlin and Richard S. Zemel. The Toronto Paper Matching System: An automated paper-reviewer assignment system. In *ICML Workshop on Peer Reviewing and Publishing Models*, 2013. 1.1, 2.1, 3.1, 3.6.1, 3.6.6, 4.1, 5.1.1, 5.1.2, 7.1.2, 8.3.1

[30] Laurent Charlin, Richard S Zemel, and Craig Boutilier. A framework for optimizing paper matching. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 11, pages 86–95, 2011. 2.1, 2.4.1, 3.1, 3.6.1, 4.1, 5.1.2, 8.3.1

[31] Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S Weld. SPECTER: Document-level representation learning using citation-informed transformers. In *Annual Meeting of the Association for Computational Linguistics*, pages 2270–2282, 2020. 3.7

[32] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, 2009. Chapter 26.3. 2.5.3

[33] David Roxbee Cox. *Planning of experiments*. Wiley, 1958. 3.6.2

[34] Liyi Dai, Ching-Hua Chen, and John R. Birge. Convergence properties of two-stage stochastic programming. *Journal of Optimization Theory and Applications*, 106(3):489–509, 2000. 4.2

[35] David J Deming, Justine S Hastings, Thomas J Kane, and Douglas O Staiger. School choice, school quality, and postsecondary attainment. *American Economic Review*, 104 (3):991–1013, 2014. 3.7

[36] Komal Dhull, Steven Jecmen, Pravesh Kothari, and Nihar B Shah. The price of strategyproofing peer assessment. In *The 9th AAAI Conference on Human Computation and Crowdsourcing*, volume 2, 2022. 1.3, 3.1, 3.6.1, 5

[37] John Dickerson, Karthik Sankararaman, Aravind Srinivasan, and Pan Xu. Allocation problems in ride-sharing platforms: Online matching with offline reusable resources. In *AAAI Conference on Artificial Intelligence*, volume 32, 2018. 4

[38] John P. Dickerson, Ariel D. Procaccia, and Tuomas Sandholm. Dynamic matching via weighted myopia with application to kidney exchange. In *AAAI Conference on Artificial Intelligence*, 2012. 4

[39] Jorge Díez Peláez, Óscar Luaces Rodríguez, Amparo Alonso Betanzos, Alicia Troncoso, and Antonio Bahamonde Rionda. Peer assessment in MOOCs using preference learning via matrix factorization. In *NeurIPS Workshop on Data Driven Education*, 2013. 5

[40] Efim A Dinic. Algorithm for solution of a problem of maximum flow in networks with power estimation. In *Soviet Math. Doklady*, volume 11, pages 1277–1280, 1970. 2.5.3

[41] Shaddin Dughmi and Arpita Ghosh. Truthful assignment without money. In *Proceedings of the 11th ACM conference on Electronic Commerce*, pages 325–334, 2010. 5

[42] Carsten Eickhoff and Arjen P. de Vries. Increasing cheat robustness of crowdsourcing tasks. *Information Retrieval*, 16:121–137, 2012. 8

[43] Thorsten Eisenhofer, Erwin Quiring, Jonas Möller, Doreen Riepel, Thorsten Holz, and Konrad Rieck. No more reviewer #2: Subverting automatic paper-reviewer assignment using adversarial learning. *arXiv preprint arXiv:2303.14443*, 2023. 1.2

[44] Bruno Escoffier, Laurent Gourvès, Jérôme Monnot, and Olivier Spanjaard. Two-stage stochastic matching and spanning tree problems: Polynomial instances and approximation. *European Journal of Operational Research*, 205(1):19–30, 2010. 4

[45] Dhivya Eswaran, Stephan Günnemann, Christos Faloutsos, Disha Makhija, and Mohit Kumar. ZooBP: Belief propagation for heterogeneous networks. *Proceedings of the VLDB Endowment*, 10(5):625–636, 2017. 7, 8

[46] Uriel Feige, Vahab S. Mirrokni, and Jan Vondrák. Maximizing non-monotone submodular functions. *SIAM Journal on Computing*, 40(4):1133–1153, 2011. 4.6.3

[47] Jon Feldman, Aranyak Mehta, Vahab Mirrokni, and Shan Muthukrishnan. Online stochastic matching: Beating 1-1/e. In *50th Annual IEEE Symposium on Foundations of Computer Science*, pages 117–126. IEEE, 2009. 4

[48] Yiding Feng, Rad Niazadeh, and Amin Saberi. Two-stage stochastic matching with application to ride hailing. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 2862–2877. SIAM, 2021. 4

[49] Tanner Fiez, Nihar B. Shah, and Lillian Ratliff. A SUPER* algorithm to optimize paper bidding in peer review. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2020. 2.5.4, 5.1.1, 1

[50] Alessandro Di Fiore and Marcio Souza. Are peer reviews the future of performance evaluations? *Harvard Business Review*, Jan 2021. ISSN 0017-8012. 5

[51] Felix Fischer and Max Klimm. Optimal impartial selection. *SIAM Journal on Computing*, 44(5):1263–1285, 2015. 5, 5.1.1

[52] Peter A Flach, Sebastian Spiegler, Bruno Golénia, Simon Price, John Guiver, Ralf Herbrich, Thore Graepel, and Mohammed J Zaki. Novel tools to streamline the conference review process: Experiences from SIGKDD'09. *ACM SIGKDD Explorations Newsletter*, 11(2):63–67, 2010. 1.1, 2.1, 2.4.1, 3.1, 3.6.1, 4.1, 8.3.1

[53] Ujwal Gadiraju, Ricardo Kawase, Stefan Dietze, and Gianluca Demartini. Understanding malicious behavior in crowdsourcing platforms: The case of online surveys. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015. 8

[54] Naveen Garg, Telikepalli Kavitha, Amit Kumar, Kurt Mehlhorn, and Julián Mestre. Assigning papers to referees. *Algorithmica*, 58(1):119–136, 2010. 1.1, 2.5.1

[55] Shayan Oveis Gharan and Jan Vondrák. Submodular maximization by simulated annealing. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 1098–1116. SIAM, 2011. 4.6.3

[56] Alexandre Gilotte, Clément Calauzènes, Thomas Nedelec, Alexandre Abraham, and Simon Dollé. Offline A/B testing for recommender systems. In *ACM International Conference on Web Search and Data Mining*, pages 198–206, 2018. 3

[57] V.P. Godambe and V.M. Joshi. Admissibility and Bayes estimation in sampling finite populations. *The Annals of Mathematical Statistics*, 36(6):1707–1722, 1965. 3.2

[58] Andrew V. Goldberg. Finding a maximum density subgraph. *University of California, Berkeley EECS Department Technical Report*, 1984. 7, 7.2.2

[59] Judy Goldsmith and Robert H. Sloan. The AI conference paper assignment problem. *AAAI Workshop*, WS-07-10:53–57, 12 2007. 3.1, 3.6.1, 4.1, 8.3.1

[60] Judy Goldsmith and Robert H. Sloan. The AI conference paper assignment problem. In *AAAI Workshop on Preference Handling for Artificial Intelligence*, pages 53–57, 2007. 1.1, 2.1, 2.4.1, 5.1.2

[61] Longhua Guo, Jie Wu, Wei Chang, Jun Wu, and Jianhua Li. K-loop free assignment in conference review systems. In *2018 International Conference on Computing, Networking and Communications (ICNC)*, pages 542–547. IEEE, 2018. 6.2.3, 7, 7.2

[62] András Hajnal and Endre Szemerédi. Proof of a conjecture of P. Erdos. *Combinatorial Theory and its Applications*, 2:601–623, 1970. 5.3

[63] Horace He. OpenReview explorer. `https://github.com/Chillee/OpenReviewExplorer`, 2020. Accessed May 26, 2021. 4.3.1, 5.3.1

[64] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. In *The Collected Works of Wassily Hoeffding*, pages 409–426. Springer, 1994. 4.7

[65] Ron Holzman and Hervé Moulin. Impartial nominations for a prize. *Econometrica*, 81 (1):173–196, 2013. 5, 5.1.1

[66] Bryan Hooi, Hyun Ah Song, Alex Beutel, Neil Shah, Kijung Shin, and Christos Faloutsos. FRAUDAR: Bounding graph fraud in the face of camouflage. In *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 895–904, 2016. 7, 7.3.2

[67] Bryan Hooi, Kijung Shin, Hemank Lamba, and Christos Faloutsos. TellTail: Fast scoring and detection of dense subgraphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4150–4157, 2020. 7, 7.2.2

[68] Daniel G Horvitz and Donovan J Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47 (260):663–685, 1952. 3.2

[69] Aanund Hylland and Richard Zeckhauser. The efficient allocation of individuals to positions. *Journal of Political Economy*, 87(2):293–314, 1979. 2

[70] Guido W. Imbens and Charles F. Manski. Confidence intervals for partially identified parameters. *Econometrica*, 2004. 3.3.3, 3.3.4, 3.6.4

[71] Terne Thorn Jakobsen and Anna Rogers. What factors should paper-reviewer assignments rely on? Community perspectives on issues and ideals in conference peer-review. In *Con-*

ference of the North American Chapter of the Association for Computational Linguistics, pages 4810–4823, 2022. 3.1

[72] Steven Jecmen, Hanrui Zhang, Ryan Liu, Nihar B. Shah, Vincent Conitzer, and Fei Fang. Mitigating manipulation in peer review via randomized reviewer assignments. In *Advances in Neural Information Processing Systems*, 2020. 1.3, 6, 6.2.7

[73] Steven Jecmen, Nihar B Shah, Fei Fang, and Vincent Conitzer. Tradeoffs in preventing manipulation in paper bidding for reviewer assignment. In *ML Evaluation Standards Workshop at ICLR*, 2022. 1.3, 9

[74] Steven Jecmen, Hanrui Zhang, Ryan Liu, Fei Fang, Vincent Conitzer, and Nihar B Shah. Near-optimal reviewer splitting in two-phase paper reviewing and conference experiment design. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 10, pages 102–113, 2022. 1.3, 9

[75] Steven Jecmen, Minji Yoon, Vincent Conitzer, Nihar B Shah, and Fei Fang. A dataset on malicious paper bidding in peer review. In *Proceedings of the ACM Web Conference 2023*, pages 3816–3826, 2023. 1.3

[76] Albert Xin Jiang and Kevin Leyton-Brown. Bidding agents for online auctions with hidden bids. *Machine Learning*, 67:117–143, 2007. 6.4

[77] Anson Kahng, Yasmine Kotturi, Chinmay Kulkarni, David Kurokawa, and Ariel D. Procaccia. Ranking wily people who rank each other. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 5, 5.1.1

[78] Maryam Karimzadehgan, ChengXiang Zhai, and Geneva Belford. Multi-aspect expertise matching for review assignment. In *17th ACM Conference on Information and Knowledge Management*, pages 1113–1122, 2008. 4.1, 4.3.1

[79] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Springer, 1972. 2.3.1, 2.6, 4.2, 4.7

[80] Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. An optimal algorithm for online bipartite matching. In *22nd Annual ACM Symposium on Theory of Computing*, pages 352–358, 1990. 4

[81] Irit Katriel, Claire Kenyon-Mathieu, and Eli Upfal. Commitment under uncertainty: Two-stage stochastic matching problems. *Theoretical Computer Science*, 408(2-3):213–223, 2008. 4

[82] Samir Khan, Martin Saveski, and Johan Ugander. Off-policy evaluation beyond overlap: Partial identification through smoothness. *arXiv preprint arXiv:2305.11812*, 2023. 3.6.4, 3.7

[83] Henry A. Kierstead, Alexandr V. Kostochka, Marcelo Mydlarz, and Endre Szemerédi. A fast algorithm for equitable coloring. *Combinatorica*, 30(2):217–224, 2010. 5.3

[84] Alan J. King and R. Tyrrell Rockafellar. Asymptotic theory for solutions in statistical estimation and stochastic programming. *Mathematics of Operations Research*, 18(1):148–162, 1993. 4.2

[85] Ari Kobren, Barna Saha, and Andrew McCallum. Paper matching with local fairness

constraints. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1247–1257, 2019. 1.1, 2.1, 2.4.1, 3.1, 3.6.1, 3.7, 4.1, 8.3.1

[86] Nan Kong and Andrew J. Schaefer. A factor 1/2 approximation algorithm for two-stage stochastic matching problems. *European Journal of Operational Research*, 172(3):740–746, 2006. 4

[87] Dmytro Korzhyk, Vincent Conitzer, and Ronald Parr. Complexity of computing optimal Stackelberg strategies in security resource allocation games. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010. 2

[88] Elias Koutsoupias. Scheduling without payments. *Theory of Computing Systems*, 54(3): 375–387, 2014. 5

[89] Ariel Kulik, Kanthi Sarpatwar, Baruch Schieber, and Hadas Shachnai. Generalized assignment via submodular optimization with reserved capacity. *arXiv preprint arXiv:1907.01745*, 2019. 4.6.3, 4.7

[90] Srijan Kumar, Bryan Hooi, Disha Makhija, Mohit Kumar, Christos Faloutsos, and V. S. Subrahmanian. Rev2: Fraudulent user prediction in rating platforms. *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 2018. 7, 8

[91] David Kurokawa, Omer Lev, Jamie Morgenstern, and Ariel D. Procaccia. Impartial peer review. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015. 5

[92] John Langford. Bidding problems. `https://hunch.net/?p=407`, 2008. Accessed February 4, 2024. 1.2

[93] Neil Lawrence and Corinna Cortes. The NIPS experiment. `http://inverseprobability.com/2014/12/16/the-nips-experiment`, 2014. Accessed February 4, 2024. 3, 4

[94] Mathias Lecuyer, Joshua Lockerman, Lamont Nelson, Siddhartha Sen, Amit Sharma, and Aleksandrs Slivkins. Harvesting randomness to optimize distributed systems. In *ACM Workshop on Hot Topics in Networks*, pages 178–184, 2017. 3

[95] Euiwoong Lee and Sahil Singla. Maximum matching in the online batch-arrival model. *ACM Transactions on Algorithms*, 16(4):1–31, 2020. 4

[96] Kevin Leyton-Brown, Mausam, Yatin Nandwani, Hedayat Zarkoob, Chris Cameron, Neil Newman, and Dinesh Raghu. Matching papers and reviewers at large conferences. *arXiv preprint arXiv:2202.12273*, 2022. 1.1, 2, 2.3.2, 3.1, 3.6.1, 3.6.6, 3.7, 6, 6.2.3, 6.2.4, 7.2

[97] Baochun Li and Y. Thomas Hou. The new automated IEEE INFOCOM review assignment system. *IEEE Network*, 30(5):18–24, 2016. 5.1.2

[98] Jing Wu Lian, Nicholas Mattei, Renee Noble, and Toby Walsh. The conference paper assignment problem: Using order weighted averages to assign indivisible goods. In *Thirty-Second AAAI Conference on Artificial Intelligence*, volume 32, 2018. 2.4.1, 3.7

[99] Michael L Littman. Collusion rings threaten the integrity of computer science research. *Communications of the ACM*, 64(6):43–44, 2021. 1.2, 7, 8

[100] Xiang Liu, Torsten Suel, and Nasir Memon. A robust model for paper reviewer assignment. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 25–32, 2014. 1.1, 3.1

[101] Gurobi Optimization LLC. Gurobi optimizer reference manual, 2020. URL `http://www.gurobi.com`. 2.4, 2.5.4, 4.6.1

[102] Cheng Long, Raymond Chi-Wing Wong, Yu Peng, and Liangliang Ye. On good and fair paper-reviewer assignment. In *2013 IEEE 13th International Conference on Data Mining*, pages 1145–1150. IEEE, 12 2013. 1.1, 2.4.1, 3.1, 3.6.1

[103] Samuel Madden and David DeWitt. Impact of double-blind reviewing on SIGMOD publication rates. *ACM SIGMOD Record*, 35(2):29–32, 2006. 4

[104] Charles F Manski. Nonparametric bounds on treatment effects. *The American Economic Review*, 80(2):319–323, 1990. 3, 3.3.3

[105] Emaad Manzoor and Nihar B. Shah. Uncovering latent biases in text: Method and application to peer review. In *INFORMS Workshop on Data Science*, 2020. 4

[106] Nicholas Mattei and Toby Walsh. PrefLib: A library for preferences http://www. preflib. org. In *Algorithmic Decision Theory: Third International Conference*, pages 259–270. Springer, 2013. 2.4, 4.1, 4.3.1, 4.6.1, 7.1.2, 8

[107] Nicholas Mattei, Paolo Turrini, and Stanislav Zhydkov. PeerNomination: Relaxing exactness for increased accuracy in peer selection. In *International Joint Conference on Artificial Intelligence*, 2020. 5, 5.7

[108] Reshef Meir, Jérôme Lang, Julien Lesca, Natan Kaminsky, and Nicholas Mattei. A market-inspired bidding scheme for peer review paper assignment. In *Games, Agents, and Incentives Workshop at AAMAS*, 2020. 2, 4.1, 4.3.1, 4.6.1, 5.1.1

[109] Michael R Merrifield and Donald G Saari. Telescope time without tears: A distributed approach to peer review. *Astronomy & Geophysics*, 50(4):4–16, 2009. 5, 6.2.6

[110] David Mimno and Andrew McCallum. Expertise modeling for matching papers with reviewers. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 500–509, 2007. 1.1, 3.1, 5.1.1

[111] Adrian Mulligan, Louise Hall, and Ellen Raphael. Peer review in a changing world: An international study measuring the attitudes of researchers. *Journal of the Association for Information Science and Technology*, 64(1):132–161, 2013. 1

[112] Parinaz Naghizadeh and Mingyan Liu. Incentives, quality, and risks: A look into the NSF proposal review pilot. *arXiv preprint arXiv:1307.6528*, 2013. 5

[113] Graham Neubig, John Wieting, Arya McCarthy, Amanda Stent, Natalie Schluter, and Trevor Cohn. ACL reviewer matching code. `https://github.com/acl-org/reviewer-paper-matching`, 2020. Accessed May 17, 2023. 1.1, 3.1, 3.6.6

[114] David Nicholas, Anthony Watkinson, Hamid R. Jamali, Eti Herman, Carol Tenopir, Rachel Volentine, Suzie Allard, and Kenneth Levine. Peer review: Still king in the digital age. *Learned Publishing*, 28(1):15–21, 2015. 1

[115] Victor M. Panaretos and Yoav Zemel. Statistical aspects of Wasserstein distances. *Annual Review of Statistics and Its Application*, 6:405–431, 2019. 4.4.1

[116] Shashank Pandit, Duen Horng Chau, Samuel Wang, and Christos Faloutsos. NetProbe: A fast and scalable system for fraud detection in online auction networks. In *Proceedings of the 16th International Conference on World Wide Web*, pages 201–210, 2007. 7

[117] Mario Paolucci and Francisco Grimaldo. Mechanism change in a simulation of peer review: From junk support to elitism. *Scientometrics*, 99(3):663–688, 2014. 1.2, 8.1.1

[118] Ferdinando Patat, Wolfgang Kerzendorf, Dominic Bordelon, Glen Van de Ven, and Tyler Pritchard. The distributed peer review experiment. *The Messenger*, 177:3–13, 2019. 4

[119] Justin Payan and Yair Zick. I will have order! Optimizing orders for fair reviewer assignment. In *International Joint Conference on Artificial Intelligence*, 2022. 3.6.1, 3.7

[120] Chris Piech, Jonathan Huang, Zhenghao Chen, Chuong Do, Andrew Ng, and Daphne Koller. Tuned models of peer assessment in MOOCs. *arXiv preprint arXiv:1307.2579*, 2013. 5

[121] Elizabeth Pier, Joshua Raclaw, Anna Kaatz, Markus Brauer, Molly Carnes, Mitchell Nathan, and Cecilia Ford. Your comments are meaner than your score: Score calibration talk influences intra- and inter-panel variability during scientific grant peer review. *Research Evaluation*, 26(1):1–14, 2017. 4

[122] Jean Pouget-Abadie, Guillaume Saint-Jacques, Martin Saveski, Weitao Duan, Souvik Ghosh, Ya Xu, and Edoardo M Airoldi. Testing for arbitrary interference on experimentation platforms. *Biometrika*, 106(4):929–940, 2019. 3.6.2

[123] B Aditya Prakash, Ashwin Sridharan, Mukund Seshadri, Sridhar Machiraju, and Christos Faloutsos. EigenSpokes: Surprising patterns and scalable community chipping in large graphs. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 435–448. Springer, 2010. 7

[124] Eric Price. The NIPS experiment. `http://blog.mrtz.org/2014/12/15/the-nips-experiment.html`, 2014. Accessed May 17, 2023. 3, 4

[125] Ariel D. Procaccia and Moshe Tennenholtz. Approximate mechanism design without money. *ACM Transactions on Economics and Computation (TEAC)*, 1(4):1–26, 2013. 5

[126] Herbert Robbins. A remark on Stirling's formula. *The American Mathematical Monthly*, 62(1):26–29, 1955. 4.7

[127] Marko A Rodriguez and Johan Bollen. An algorithm to determine peer-reviewers. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 319–328, 2008. 1.1, 3.1

[128] Thomas Rothvoss. A direct proof for Lovett's bound on the communication complexity of low rank matrices. *arXiv preprint arXiv:1409.6366*, 2014. 4.4.1, 4.7

[129] Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5):688, 1974. 3.2

[130] Noveen Sachdeva, Yi Su, and Thorsten Joachims. Off-policy bandits with deficient sup-

port. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 965–975, 2020. 3

[131] Martin Saveski, Jean Pouget-Abadie, Guillaume Saint-Jacques, Weitao Duan, Souvik Ghosh, Ya Xu, and Edoardo M. Airoldi. Detecting network effects: Randomizing over randomized experiments. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1027–1035, 2017. 3.6.2

[132] Martin Saveski, Steven Jecmen, Nihar B Shah, and Johan Ugander. Counterfactual evaluation of peer-review assignment policies. In *Advances in Neural Information Processing Systems*, 2023. 1.3, 3

[133] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. Recommendations as treatments: Debiasing learning and evaluation. In *International Conference on Machine Learning*, pages 1670–1679. PMLR, 2016. 3

[134] Nihar B. Shah. Challenges, experiments, and computational solutions in peer review. *Communications of the ACM*, 65(6):76–87, 2022. 1, 2.1, 5, 6, 6.4, 7.1.1, 8.3.1

[135] Nihar B. Shah, Joseph K. Bradley, Abhay Parekh, Martin Wainwright, and Kannan Ramchandran. A case for ordinal peer-evaluation in MOOCs. In *NeurIPS Workshop on Data Driven Education*, 2013. 5

[136] Nihar B. Shah, Behzad Tabibian, Krikamol Muandet, Isabelle Guyon, and Ulrike Von Luxburg. Design and analysis of the NIPS 2016 review process. *The Journal of Machine Learning Research*, 19(1):1913–1946, 2018. 1.1, 2.4, 2.4.2, 3.1, 4.3.1, 5, 6.2.1, 6.1b, 6.2.5, 7.1.1, 7.1.2, 8.1.1, 8.3.1

[137] Ivan Stelmakh, Nihar B. Shah, and Aarti Singh. On testing for biases in peer review. In *Advances in Neural Information Processing Systems*, volume 32, pages 5286–5296, 2019. 4

[138] Ivan Stelmakh, Nihar Shah, and Aarti Singh. PeerReview4All: Fair and accurate reviewer assignment in peer review. *The Journal of Machine Learning Research*, 22(1):7393–7458, 2021. 1.1, 2.5.1, 3.1, 3.6.1, 3.7

[139] Ivan Stelmakh, Nihar B Shah, and Aarti Singh. Catch me if I can: Detecting strategic behaviour in peer assessment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4794–4802, 2021. 4

[140] Ivan Stelmakh, Nihar B Shah, Aarti Singh, and Hal Daumé III. Prior and prejudice: The novice reviewers' bias against resubmissions in conference peer review. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW1):1–17, 2021. 4

[141] Ivan Stelmakh, Nihar B. Shah, Aarti Singh, and Hal Daumé III. A novice-reviewer experiment to address scarcity of qualified reviewers in large conferences. In *AAAI Conference on Artificial Intelligence*, volume 35, pages 4785–4793, 2021. 3.1

[142] Ivan Stelmakh, Charvi Rastogi, Nihar B Shah, Aarti Singh, and Hal Daumé III. A large scale randomized controlled trial on herding in peer-review discussions. *PLOS One*, 18 (7):e0287443, 2023. 3, 4

[143] Ivan Stelmakh, John Wieting, Graham Neubig, and Nihar B. Shah. A gold standard dataset

for the reviewer assignment problem. *arXiv preprint arXiv:2303.16750*, 2023. 3.1, 7.1.2, 7.4.1

[144] Wenbin Tang, Jie Tang, and Chenhao Tan. Expertise matching via constraint-based optimization. In *International Conference on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 34–41, 2010. 1.1, 2.1, 2.4.1, 3.1, 3.6.1, 5.1.2

[145] Camillo J. Taylor. On the optimal assignment of conference papers to reviewers. *University of Pennsylvania Department of Computer and Information Science Technical Report*, 2008. 1.1, 3.1, 3.6.1, 5.1.2

[146] Misha Teplitskiy, Hardeep Ranu, Gary Gray, Michael Menietti, Eva Guinan, and Karim R. Lakhani. Do experts listen to other experts? Field experimental evidence from peer review. `https://www.hbs.edu/ris/Publication%20Files/19-107_06115731-d0ae-4a11-ab1d-ecaec2118921.pdf`, 2019. 4

[147] Stefan Thurner and Rudolf Hanel. Peer-review in a world with rational scientists: Toward selection of the average. *The European Physical Journal B*, 84(4):707–711, 2011. 5

[148] Andrew Tomkins, Min Zhang, and William D Heavlin. Reviewer bias in single-versus double-blind peer review. *Proceedings of the National Academy of Sciences*, 114(48): 12708–12713, 2017. 3, 4, 5

[149] Hong Diep Tran, Guillaume Cabanac, and Gilles Hubert. Expert suggestion for conference program committees. In *11th International Conference on Research Challenges in Information Science*, pages 221–232, May 2017. 1.1, 3.1

[150] Charalampos Tsourakakis, Francesco Bonchi, Aristides Gionis, Francesco Gullo, and Maria Tsiarli. Denser than the densest subgraph: Extracting optimal quasi-cliques with quality guarantees. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 104–112, 2013. 7, 7.2.2, 7.2.2, 7.3.2

[151] Anthony K.H. Tung. Impact of double blind reviewing on SIGMOD publication: A more detailed analysis. *ACM SIGMOD Record*, 35(3):6–7, 2006. 4

[152] Johan Ugander, Brian Karrer, Lars Backstrom, and Jon Kleinberg. Graph cluster randomization: Network exposure to multiple universes. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 329–337, 2013. 3.6.2

[153] Jan Van Den Brand, Yin Tat Lee, Yang P Liu, Thatchaphol Saranurak, Aaron Sidford, Zhao Song, and Di Wang. Minimum cost flows, MDPs, and ℓ_1-regression in nearly linear time for dense instances. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 859–869, 2021. 2.2.1, 5.2.3, 5.2.4

[154] T. N. Vijaykumar. Potential organized fraud in ACM/IEEE computer architecture conferences. `https://medium.com/@tnvijayk/potential-organized-fraud-in-acm-ieee-computer-architecture-conferences-ccd61169370d`, 2020. Accessed May 17, 2023. 1.2, 8

[155] Mark Ware. Peer review: Benefits, perceptions and alternatives. *Publishing Research Consortium*, 4:1–20, 2008. 1

[156] K.N. Wexley and Richard Klimoski. Performance appraisal: An update. *Research in*

Personnel and Human Resources Management, 2:35–79, 01 1984. 5

[157] Paul Wilson. Academic fraud: Solving the crisis in modern academia. *Exchanges: The Interdisciplinary Research Journal*, 7(3):14–44, 2020. 7

[158] Alex Wood-Doughty and Cameron Bruggeman. The incentives platform at Lyft. In *ACM International Conference on Web Search and Data Mining*, pages 1654–1654, 2022. 3, 3.7

[159] Ruihan Wu, Chuan Guo, Felix Wu, Rahul Kidambi, Laurens Van Der Maaten, and Kilian Weinberger. Making paper reviewing robust to bid manipulation attacks. In *International Conference on Machine Learning*, pages 11240–11250. PMLR, 2021. 6, 6.2.5, 6.1, 6.4, 7, 7.1.2, 7.4.1, 8, 8.6

[160] Yichong Xu, Han Zhao, Xiaofei Shi, Jeremy Zhang, and Nihar B. Shah. On strategyproof conference peer review. In *International Joint Conference on Artificial Intelligence*, pages 616–622, 2018. 2.4, 3.6.1, 4.1, 4.3.1, 5, 5.3.1, 5.4.2, 7.1.2, 7.4.1, 8, 1

[161] Yixuan Even Xu, Steven Jecmen, Zimeng Song, and Fei Fang. A one-size-fits-all approach to improving randomness in paper assignment. In *Advances in Neural Information Processing Systems*, 2023. 2.7

[162] Mihalis Yannakakis. Node-and edge-deletion NP-complete problems. In *ACM Symposium on Theory of Computing*, pages 253–264, 1978. 5.2.5, 5.6