A Unified Framework for Paper Assignment

Michael Cui

CMU-CS-26-115
May 2026

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Fei Fang, Chair
Nihar Shah

Submitted in partial fulfillment of the requirements
for the degree of Masters of Science.

Keywords: peer review, paper assignment, optimization, robustness, diversity, machine learning systems, linear programming, artificial intelligence

To everyone who made this work possible.

Abstract

Assigning submitted papers to appropriate reviewers is a fundamental component of the peer-review process in large academic conferences. In modern conference settings, this task has become increasingly challenging due to the scale of submissions and the need to satisfy multiple competing objectives simultaneously. In particular, program chairs must balance reviewer expertise, diversity considerations, and robustness to strategic behavior, while also ensuring that the assignment process remains practical at scale. This thesis aims to improve the paper-assignment process by making it more effective, more robust, and more practical for real-world peer review.

This thesis studies the problem of large-scale paper assignment from an optimization perspective. It examines the limitations of existing assignment methods, which often optimize only a subset of the relevant objectives or become computationally impractical in realistic conference settings. To address these limitations, the thesis presents Robust Assignment via Marginal Perturbation (RAMP), a unified framework for scalable, robust, and diversity-aware reviewer assignment. The proposed framework combines a linearized perturbed-maximization objective with soft constraints that incorporate multiple practical desiderata into a single optimization procedure, together with an attribute-aware sampling method for converting fractional assignments into integral ones.

In addition to presenting the algorithmic framework, this thesis discusses the practical challenges and lessons that arose in deploying the method for major AI conferences, including AAAI 2026, AAMAS 2026, and EC 2026. It also describes an interface that enables future conference organizers to run the matching process directly, helping bridge the gap between optimization research and real conference workflows.

Acknowledgments

I am deeply grateful to my advisor, Fei Fang, for her guidance, support, and encouragement throughout this thesis. Her advice helped shape both the technical direction of this work and my understanding of how algorithmic research can be used in real-world systems.

I would also like to thank Nihar Shah for serving on my thesis committee and for his valuable feedback on this work. I am grateful to my collaborators Chenxin Dai and Yixuan Xu for their ideas, discussions, and contributions to the broader project on paper assignment. I also thank the conference organizers and collaborators who helped with the deployment experiences discussed in this thesis. In particular, I thank Matthew Taylor, Chad Jenkins, and Kevin Leyton-Brown for their valuable input and practical feedback, which helped me better understand the challenges of applying matching algorithms in real conference workflows.

Lastly, I am thankful to my friends and family for their patience, support, and encouragement throughout my time at Carnegie Mellon. This thesis would not have been possible without them.

This work was supported by NSF IIS-2200410.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Peer Review

Peer review is a central mechanism for ensuring the quality, credibility, and integrity of scientific research. In most academic venues, submitted work is evaluated by experts in the relevant field before it is accepted for publication. This process serves as a filter that helps identify contributions that are novel, technically sound, and significant, while also helping maintain the reliability of the scientific record and supporting the cumulative advancement of knowledge [7, 11].

In a typical conference setting, the peer-review process begins when authors submit their papers to a venue. Program chairs or senior committee members then oversee the process of assigning each submission to a set of reviewers. These reviewers read the paper, evaluate its strengths and weaknesses, and provide written reviews and recommendations. Based on these reviews, along with discussion among reviewers and area chairs when applicable, the conference ultimately decides which papers to accept. Although many aspects of this process involve human judgment, the quality of those judgments depends heavily on whether the right reviewers were assigned in the first place.

Paper-reviewer matching therefore plays a foundational role in peer review. Before any scientific contribution can be evaluated, the conference must determine which reviewers are most appropriate for each paper. A good assignment should place each paper in the hands of reviewers with relevant expertise, while also balancing reviewer workload and respecting conflicts of interest. If the matching is poor, even an otherwise well-functioning review process can be compromised: papers may be judged by reviewers who lack the necessary background, conflicts may be overlooked, or the workload may be distributed unfairly across the committee. In this sense, paper assignment is not a peripheral administrative step, but one of the key infrastructure problems underlying effective peer review.

Beyond its role in selecting qualified reviewers, paper matching also affects the broader fairness and legitimacy of the review process. Conferences often seek to ensure that each paper receives a balanced and independent evaluation, which may require considering additional properties such as diversity of reviewer expertise, institutional separation, or seniority coverage. As conferences have grown in scale, these concerns have made reviewer assignment increasingly complex, turning it into a major computational and organizational challenge rather than a purely

manual task.

Peer review also contributes to the improvement of research. Reviewers provide critical feedback on methodology, assumptions, and presentation, often leading to revisions that strengthen the clarity, rigor, and completeness of a paper prior to publication. At the same time, the review process helps establish and reinforce scientific norms by shaping what communities regard as sufficient evidence, acceptable methodology, and meaningful contribution. In this way, peer review not only evaluates individual submissions, but also influences the evolution of research practices over time.

Taken together, these roles make peer review a foundational component of modern scientific infrastructure. Paper–reviewer matching is a central part of that infrastructure, since it determines how expertise, effort, and judgment are allocated across submitted work. For this reason, improving large-scale paper assignment is an important step toward improving the quality, fairness, and robustness of peer review itself.

1.2 Challenges in Large-Scale Paper Assignment

At the scale of modern academic conferences, manual reviewer assignment is no longer feasible. Large venues may receive thousands or even tens of thousands of submissions, each of which must be assigned to multiple reviewers while respecting reviewer capacities, conflicts of interest, and conference-specific requirements. The number of possible paper–reviewer pairs is therefore enormous, making purely manual assignment both prohibitively time consuming and highly error prone. As a result, modern conferences necessarily rely on automated matching systems, that is, paper-assignment algorithms, to generate reviewer assignments in a systematic and reproducible way.

The first challenge in large-scale paper assignment is achieving high matching quality while respecting the many constraints of the review process. A good assignment should place each paper in the hands of reviewers with the appropriate technical expertise, but reviewer expertise is only imperfectly observed. Most conference management systems estimate paper–reviewer suitability using a combination of bidding signals, publication records, keywords, topical representations, or learned similarity measures. These signals can be informative, but they are inevitably noisy and incomplete [2, 3, 10, 14]. At the same time, the assignment must satisfy feasibility constraints: each paper requires a fixed number of reviewers, each reviewer has limited capacity, and many reviewer–paper pairs must be excluded due to conflicts of interest. Additional considerations such as balancing reviewer loads or ensuring local fairness further complicate the problem [8, 13, 15]. Thus, the challenge is not merely to maximize similarity, but to do so while respecting the full set of constraints imposed by real conference workflows.

The second challenge is robustness to strategic behavior and collusion. As publication outcomes at top venues play a major role in academic reputation and career advancement, reviewer assignment procedures cannot assume that all participants behave passively or truthfully. Deterministic or overly predictable matching rules may create opportunities for reviewers to manipulate bids, seek assignment to particular papers, or exploit reciprocal structures in the assignment graph. Related concerns also arise when reviewers assigned to the same paper are too closely connected through coauthorship or other professional ties. Prior work has addressed some of

these issues through randomized assignment, probability-limited matching, and explicit anti-collusion constraints [5, 9, 18]. However, robustness mechanisms often come at a cost, either in matching quality or in increased optimization complexity.

The third challenge is scalability. Even if an assignment method produces high-quality and robust matches, it may still be unusable in practice if it is too slow or too memory-intensive for conference deployment. Modern computer science conferences now routinely receive more than $10{,}000$ submissions in a single review cycle [19], and the associated paper–reviewer graph can be extremely large. Methods that perform well on small or medium-sized datasets may therefore become impractical at conference scale, especially when additional robustness or anti-collusion constraints are included [9, 18]. For this reason, runtime is not merely an implementation detail, but one of the core requirements of any paper-assignment system intended for real-world use.

Taken together, these considerations show that modern paper assignment is not just an administrative task, but a major algorithmic challenge. Effective reviewer assignment requires methods that can simultaneously maintain matching quality, provide robustness against collusion and strategic behavior, and scale to the size of modern conference review processes. This thesis is motivated by the need for a framework that addresses all three.

1.3 Thesis Contributions and Organization

Given the importance of peer review, this thesis investigates scalable, robust, and diversity-aware methods for paper assignment. It builds on recent work on RAMP, a unified framework that combines perturbed similarity maximization, soft structural constraints, and attribute-aware sampling into a single assignment pipeline [1].

The primary contribution of this thesis is to present reviewer assignment as a genuinely multi-objective problem. Rather than treating assignment solely as the task of maximizing reviewer–paper similarity, it is framed as balancing reviewer expertise with feasibility constraints, diversity objectives, structural robustness, and computational efficiency. In this context, the thesis formalizes the optimization framework underlying RAMP, shows how practical assignment desiderata can be expressed through soft objective components and linear constraints, and analyzes the algorithmic mechanisms that enable scalability at conference scale, including piecewise-linear approximation, sparsification, and attribute-aware sampling [5, 18]. It further provides an empirical evaluation of the framework, examining tradeoffs among assignment quality, diversity, robustness, and runtime, and discusses the requirements for deployment in real conference workflows, including interpretability, hyperparameter tuning, human oversight, and integration with existing review-management systems.

The remainder of this thesis is organized as follows:

- Chapter 2 reviews the background and related work necessary to study paper assignment. It introduces the role of peer review, explains classical similarity-based reviewer assignment, and surveys prior work on fairness, randomization, robustness, and collusion-aware matching [5, 9, 13, 16, 18].
- Chapter 3 presents the formal problem setting and the unified framework used throughout the thesis. It develops the optimization model, introduces the diversity and robustness

components, and describes the approximation, sampling, and sparsification techniques that make the framework practical at conference scale.

- Chapter 4 presents the experimental methodology and empirical evaluation. It describes the datasets, baselines, metrics, and implementation details, and analyzes the observed tradeoffs among assignment quality, diversity, robustness, and runtime.
- Chapter 5 discusses deployment in real conference workflows. It draws on practical experience from running the framework for conferences such as AAAI, AAMAS, and EC, and examines issues including sparsification, incomplete information, the role of bidding, collusion mitigation, and conference-specific operational requirements.
- Chapter 6 discusses an interface for paper matching designed to support future conference organizers in running and inspecting the matching process directly.
- Chapter 7 outlines future directions, including both technical extensions of the framework and broader opportunities for improving practical conference matching systems.
- Chapter 8 concludes the thesis by summarizing the main findings, discussing limitations, and outlining future work.

Chapter 2

Background and Related Work

2.1 Overview of Peer Review Process

The peer review process is usually structured through a hierarchy of roles, with each group contributing to a different stage of evaluation and decision-making. At the highest level are the Program Chairs, who are responsible for designing the overall review procedure and making the final acceptance and rejection decisions. Supporting them are the Area Chairs (ACs), who oversee the process at an intermediate level and provide recommendations that contribute to the final outcome. Senior Program Committee members (SPCs) play a coordinating role by facilitating reviewer discussions and preparing meta-reviews that synthesize the main points raised during evaluation. Program Committee members (PCs), in turn, act as the primary reviewers and are responsible for reading submissions and writing individual reviews. This organizational structure shows that peer review is not treated as a simple aggregation of reviewer scores, but rather as a layered process in which assessment, discussion, synthesis, and decision-making are distributed across multiple levels of responsibility [9].

Most conferences also adapt a two-phase review process intended to allocate reviewing effort more efficiently. In Phase 1, each paper is assigned a certain number of reviewers. If most reviewers are sufficiently confident and recommend rejection, the paper is rejected at this stage. Nevertheless, authors still receive several full reviews, which makes this approach more informative than desk rejection systems that often provide little or no feedback. Papers that are not rejected in Phase 1 advance to Phase 2, where they receive two or more additional reviewers. After these reviews are submitted, authors are given the opportunity to provide rebuttals. Reviewers from both phases then read one another's evaluations as well as the author responses, participate in discussion under the supervision of SPCs and ACs, and may revise their reviews before recommendations are finalized. The Program Chairs then use this material, together with the recommendations of the ACs, to make the final decisions. The purpose of this two-phase model is to focus reviewing resources on papers near the acceptance threshold.

For the purposes of this thesis, when we use the word "reviewer", it refers to PCs, which make up the bulk of the reviewers; however, we will also show that it can be generalized to SPCs or ACs alike. Similarly, we focus on only one phase of the review process, but it is obvious that this framework can be generalized to two-phase review processes.

2.2 Paper Assignment using Similarity Scores

The most classical formulation of reviewer assignment models the process as an optimization problem over a bipartite graph between papers and reviewers. Each paper-reviewer pair is associated with a similarity score that estimates the reviewer's suitability for the paper, and the goal is to maximize the total similarity of the final assignment subject to reviewer load and paper coverage constraints [15]. In its simplest form, this yields a linear program or network-flow problem that can often be solved efficiently.

A large body of work has focused on constructing high-quality similarity scores. Different systems estimate expertise using combinations of author-supplied keywords, publication records, text similarity, topic models, bidding behavior, and other signals [2, 3, 10, 14]. These similarity estimates then serve as the primary input to the assignment optimization problem.

In modern conference management systems, similarity scores (often referred to as *affinity scores*) are typically computed using text-based representations of both papers and reviewers. Systems such as *OpenReview* construct these representations from sources including paper titles, abstracts, and reviewer publication profiles, and map them into a shared vector space [12]. Similarity is then computed using measures such as cosine similarity between these vector representations. Earlier approaches relied on keyword overlap or topic models such as Latent Dirichlet Allocation (LDA), while more recent methods use neural embeddings derived from pretrained language models trained on large scholarly corpora, such as SPECTER, Sentence-BERT (SBERT), or related transformer-based encoders. In addition, auxiliary signals such as reviewer bids or co-authorship information may be incorporated to refine these estimates. These affinity scores provide a scalable and flexible proxy for reviewer expertise, and form the basis of most modern assignment systems.

This similarity-based perspective is attractive for several reasons. It is simple, interpretable, and computationally tractable. It also aligns naturally with the basic intuition that papers should be reviewed by experts. In the absence of additional constraints, similarity maximization often provides a strong baseline for assignment quality.

However, similarity-based assignment is limited in several important ways. First, similarity scores are only proxies for true expertise and may be noisy or incomplete. Second, maximizing total similarity may produce assignments that are acceptable in aggregate but undesirable in structure, for example by repeatedly assigning tightly connected reviewers to the same papers or by producing little diversity in reviewer backgrounds. Third, deterministic similarity-based assignments can be predictable, which may make them vulnerable to manipulation or collusion in settings where reviewers strategically influence bids or other assignment signals [5, 16, 18].

Throughout this thesis, we assume that similarity scores are provided by external models and are treated as fixed inputs to the assignment problem. Accordingly, the focus is not on improving the estimation of similarity itself, but on understanding and addressing the limitations that arise when such scores are imperfect, including their impact on assignment quality, diversity, robustness, and scalability. In the deployment part of this paper, we talked about approaches to take while performing paper matching for conferences, where similarity scores were not perfect.

2.3 Optimization Approaches

A wide range of optimization methods have been proposed for paper assignment. Classical approaches formulate assignment as a linear or network-flow problem with paper-demand, reviewer-capacity, and conflict-of-interest constraints [15]. These methods are attractive because they are efficient in important special cases and provide strong practical baselines.

Subsequent work has shown that the choice of objective function can substantially affect the quality profile of the final assignment. In particular, fairness-oriented methods emphasize minimum assignment quality or local balance rather than pure total similarity [8, 13]. These formulations highlight that reviewer assignment should not be evaluated only by average match quality, but also by how assignment quality is distributed across papers.

More recent work has explored randomization as a tool for robustness. Jecmen et al. propose Probability-Limited Randomized Assignment (PLRA), which computes a probabilistic assignment subject to upper bounds on pairwise marginal probabilities and then samples a deterministic matching from those marginals [5]. This reduces assignment predictability and therefore limits certain opportunities for manipulation. Xu et al. extend this idea through perturbed maximization, which encourages probability mass to be distributed across several strong reviewers rather than concentrated on a single one [18].

At the same time, other work has modeled integrity considerations more explicitly. Leyton-Brown et al. introduce optimization constraints for reducing collusion and improving diversity [9], while related work has studied collusion-aware reviewer assignment using coauthorship and bidding structure [16]. These approaches show that robustness and diversity can be represented directly within the optimization problem, but they also highlight the associated computational cost, especially when mixed-integer formulations are used at conference scale.

Taken together, this line of work suggests that effective reviewer assignment requires more than a single improvement in objective design. Rather, it requires a framework that can combine expertise-based matching, randomized robustness, and explicit structural constraints while remaining computationally feasible in large conference settings.

2.4 Summary of Prior Work

The literature on reviewer assignment points to four broad lessons. First, similarity-based optimization provides a strong and efficient baseline for matching papers to qualified reviewers [2, 3, 10, 14, 15]. Second, fairness-oriented objectives show that assignment quality should be evaluated not only in aggregate, but also in how it is distributed across papers [8, 13]. Third, randomization can improve robustness by reducing predictability and limiting certain forms of strategic manipulation [5, 18]. Fourth, explicit robustness and diversity constraints make important practical concerns directly representable, but often at a substantial computational cost [9, 16].

These observations suggest that modern paper assignment should be evaluated against a broader set of practical desiderata, rather than only by total reviewer–paper similarity. Table 2.1 summarizes the main requirements considered in this thesis.

The central challenge is that existing assignment methods tend to satisfy only some of these

Requirement	What it means
Quality	Papers should be assigned to reviewers with relevant expertise, as measured by reviewer–paper similarity scores.
Runtime	The assignment algorithm should finish within a practical amount of time for large conferences.
COI	Reviewers must not be assigned to papers with which they have a conflict of interest.
Paper demand	Each paper must receive the required number of reviewers.
Reviewer load	No reviewer should be assigned more papers than their reviewing capacity allows.
Seniority	Each paper should receive at least one experienced reviewer when required by the conference.
Coauthor distance	Reviewers assigned to the same paper should ideally not be past coauthors or closely connected in the coauthorship graph.
Diversity	Reviewers assigned to the same paper should come from different regions, subject areas, institutions, or other relevant groups.
2-cycles	Pairs of reviewers who bid positively on one another's papers should not be reciprocally assigned to those papers.

Table 2.1: Practical desiderata in large-scale paper assignment. Hard constraints must be satisfied by every feasible assignment, while soft rewards and penalties allow the assignment procedure to trade off quality, diversity, robustness, and runtime.

desiderata. Classical similarity maximization is efficient and high-quality, but does not address randomization or structural robustness. MILP approaches can encode many additional requirements, but they may be too slow at large scale. Randomized and perturbed methods improve robustness to manipulation, but do not directly handle coauthorship, diversity, seniority, or bid-based 2-cycle requirements. This motivates the need for a unified framework that can combine these properties while remaining scalable.

These observations suggest that the main challenge is no longer the absence of useful ideas, but the lack of a unified and scalable framework that can integrate them cleanly. Table 2.2 summarizes this tradeoff by comparing representative approaches across efficiency, randomization, and structural robustness properties.

The framework studied in this thesis is motivated by precisely this gap. The next chapter formalizes the reviewer-assignment problem and introduces the unified optimization framework used throughout the remainder of the thesis.

Alg.	Time	Random	Coauthor	Div.	2Cyc.
Default	✓	×	×	×	×
MILP [9]	×	×	✓	✓	✓
PLRA [5]	✓	✓	×	×	×
PM [18]	✓*	✓†	×	×	×
RAMP [1]	✓	✓†	✓	✓	✓

* partially; † improved.

Table 2.2: Comparison of properties satisfied by different paper-assignment algorithms

Chapter 3

Problem Formulation and Unified Framework

3.1 Conference Assignment Setting

We consider the paper-assignment problem for a large academic conference. Let $\mathcal{P}$ denote the set of submitted papers and $\mathcal{R}$ denote the set of reviewers. Each paper $p \in \mathcal{P}$ requires ℓ_p reviews, and each reviewer $r \in \mathcal{R}$ can review at most u_r papers. In addition, some reviewer–paper pairs are forbidden because of conflicts of interest. The goal is to compute an assignment that is both high-quality and practically suitable for the needs of a modern conference.

In classical formulations, the primary objective is to assign each paper to reviewers with high expertise, typically represented through a similarity matrix between papers and reviewers [2, 3, 10, 14, 15]. In large-scale peer review, however, conference organizers often care about more than expertise alone. They may wish to promote reviewer diversity, reduce undesirable structural patterns, and limit opportunities for strategic manipulation or collusion [5, 9, 16, 18].

The source paper on which this thesis builds is motivated by exactly this gap. It argues that large conferences need assignment procedures that balance reviewer expertise, diversity, robustness, and computational efficiency within a single scalable framework [5, 9, 18]. The purpose of this chapter is to formalize that setting and present the unified framework used throughout the rest of the thesis.

3.2 Preliminaries

For each paper–reviewer pair $(p, r) \in \mathcal{P} \times \mathcal{R}$, let $S_{p,r} \geq 0$ denote a similarity score representing the suitability of reviewer r for paper p. These scores may be derived from bidding behavior, publication records, text similarity, or related expertise signals [2, 3, 10, 14]. Let $\mathcal{F}$ denote the set of paper–reviewer pairs with conflicts of interest; any feasible assignment must exclude these pairs.

We use decision variables $x_{p,r}$ to represent assignment decisions. In a deterministic assignment, $x_{p,r} \in \{0, 1\}$ indicates whether reviewer r is assigned to paper p. In randomized formulations, $x_{p,r}$ may instead be interpreted as a marginal assignment probability. This distinction is

important because recent work has shown that randomness can improve robustness and reduce predictability in peer-review assignment [5, 18].

Throughout this chapter, the core design problem is to determine how these assignment variables should be optimized so that the resulting assignment satisfies paper demand and reviewer capacity, respects conflicts of interest, and also supports broader goals such as diversity and anti-collusion robustness.

3.3 Classical and Randomized Assignment Formulations

The classical reviewer-assignment problem can be expressed as a linear program that maximizes total similarity subject to paper-demand, reviewer-capacity, and conflict-of-interest constraints. A standard formulation is

$$\max_{x \in [0,1]^{|\mathcal{P}| \times |\mathcal{R}|}} \quad \sum_{p \in \mathcal{P}} \sum_{r \in \mathcal{R}} S_{p,r} x_{p,r} \tag{3.1}$$

$$\text{s.t.} \quad \sum_{r \in \mathcal{R}} x_{p,r} = \ell_p \qquad \forall p \in \mathcal{P}, \tag{3.2}$$

$$\sum_{p \in \mathcal{P}} x_{p,r} \leq u_r \qquad \forall r \in \mathcal{R}, \tag{3.3}$$

$$x_{p,r} = 0 \qquad \forall (p,r) \in \mathcal{F}. \tag{3.4}$$

This formulation captures the basic expertise-driven assignment problem and is a natural starting point for algorithmic reviewer matching [15].

However, deterministic similarity-maximizing assignments can be predictable and may fail to address strategic behavior or structural concerns. To mitigate this issue, Jecmen et al. propose Probability-Limited Randomized Assignment (PLRA), which computes a probabilistic assignment while imposing upper bounds on marginal assignment probabilities [5]. The resulting fractional assignment is then rounded into a deterministic matching through sampling, which limits the probability that any reviewer can secure assignment to a particular paper.

Xu et al. extend this idea through perturbed maximization, in which the assignment objective takes the form

$$\sum_{p,r} S_{p,r} f(x_{p,r}), \tag{3.5}$$

where f is a nondecreasing concave function [18]. The concavity of f rewards spreading probability mass across several strong reviewers rather than concentrating it entirely on a single reviewer. This introduces additional randomness and robustness while preserving expertise as the underlying signal.

These randomized approaches are important because they demonstrate that reviewer assignment need not be fully deterministic. At the same time, they do not by themselves fully address practical diversity goals or explicit structural anti-collusion requirements. This motivates the more general framework introduced next.

3.4 A Unified Mathematical Program for Paper Assignment

The central idea of the source paper is to formulate reviewer assignment as a single optimization program that generalizes classical similarity-based matching, PLRA, and perturbed maximization while also allowing conference organizers to encode additional soft constraints [5, 9, 18].

Let $x_{p,r} \in [0, Q]$ denote the fractional probability of assigning reviewer r to paper p, where $Q \in (0, 1]$ upper-bounds the marginal assignment probability. Let $f : [0, Q] \to \mathbb{R}$ be a fixed concave, nondecreasing perturbation function. The unified program can be written as

$$\max_{\mathbf{x},\mathbf{s}} \quad \sum_{p\in\mathcal{P}} \sum_{r\in\mathcal{R}} S_{p,r} f(x_{p,r}) + \sum_{k=1}^{K} \mathcal{O}^k(\mathbf{x}, s^k) \tag{3.6}$$

$$\text{s.t.} \quad \sum_{r\in\mathcal{R}} x_{p,r} = \ell_p \qquad \forall p \in \mathcal{P}, \tag{3.7}$$

$$\sum_{p\in\mathcal{P}} x_{p,r} \le u_r \qquad \forall r \in \mathcal{R}, \tag{3.8}$$

$$(x, s^1, \ldots, s^K) \in \Xi^1 \cap \cdots \cap \Xi^K. \tag{3.9}$$

The first term is the similarity-based objective, while each $\mathcal{O}^k$ denotes an additional soft-objective component with associated auxiliary variables s^k and feasible region Ξ^k. This structure makes the framework modular: conference organizers can add, remove, or reweight components to reflect different organizational priorities without changing the core optimization mechanism [9].

This formulation subsumes several prior methods as special cases. If all soft components are removed and $f(x) = x$ with $Q = 1$, the formulation reduces to the classical deterministic assignment problem. If $f(x) = x$ and $Q < 1$, it recovers the PLRA-style relaxation. If f is concave, it recovers perturbed maximization [5, 18]. The unified framework therefore provides a natural bridge between earlier expertise-based randomized assignment methods and richer conference-specific requirements.

Reviewer diversity. Often times, program chairs would want some diversity in the reviewers. For instance, a conference might seek geographic diversity to ensure that each paper receives perspectives from different parts of the world. This has the added benefit of reducing the likelihood of collusion, since reviewers and authors are less likely to know each other if they are geographically separated. Other conferences, especially those that are interdisciplinary, might want each paper to receive reviews from different areas of expertise, so that subject-area diversity is captured. Finally, some conferences may want diversity in reviewer seniority, so that each paper gets at least one senior reviewer while also including junior reviewers.

$\mathcal{O}^{\mathsf{div}}$ and its associated constraints capture reviewer diversity. Program chairs may wish each paper to receive reviews from a diverse set of reviewers, according to some notion of *region*.

Let $\mathcal{G}$ denote the set of regions under consideration for diversity (e.g., geographic regions, areas of expertise, or levels of seniority). For each reviewer $r \in \mathcal{R}$, let

$$\text{region}(r) \in \mathcal{G}$$

denote the region to which r belongs. We introduce a reward parameter $\lambda_{\mathsf{div}} \geq 0$ that controls the balance between diversity and similarity. For each paper $p \in \mathcal{P}$ and each region $g \in \mathcal{G}$, we introduce a slack variable $s^{\mathsf{div}}_{p,g}$ and impose the following constraints:

$$0 \leq s^{\mathsf{div}}_{p,g} \leq 1, \tag{3.10}$$

$$s^{\mathsf{div}}_{p,g} \leq \sum_{r \in \mathcal{R}:\, \mathrm{region}(r)=g} x_{p,r}. \tag{3.11}$$

Intuitively, $s^{\mathsf{div}}_{p,g}$ captures up to one unit of review probability that paper p receives from region g.

Finally, we add the following term to the overall objective:

$$\mathcal{O}^{\mathsf{div}} \sum_{p \in \mathcal{P}} \sum_{g \in \mathcal{G}} \lambda_{\mathsf{div}}\, s^{\mathsf{div}}_{p,g}. \tag{3.12}$$

These variables and constraints together define Ξ^{div}.

Co-authorship distance. The co-authorship component $\mathcal{O}^{\mathsf{co}}$ and its associated constraints encode co-authorship structure. Reviewers assigned to the same paper should ideally not have co-authored previously, nor be very close in the co-authorship graph, in order to ensure independence of opinions and reduce the risk of collusion.

We introduce a penalty parameter $\lambda_{\mathsf{co}} \geq 0$. For each reviewer $r \in \mathcal{R}$, define the (closed) co-authorship neighborhood

$$\mathcal{N}(r)\{r\} \cup \{r' \in \mathcal{R} \mid r \text{ and } r' \text{ are past coauthors}\}.$$

For each paper $p \in \mathcal{P}$ and reviewer $r \in \mathcal{R}$, we introduce a slack variable $s^{\mathsf{co}}_{p,r}$ and impose

$$s^{\mathsf{co}}_{p,r} \geq 1, \tag{3.13}$$

$$s^{\mathsf{co}}_{p,r} \geq \sum_{r' \in \mathcal{N}(r)} x_{p,r'}. \tag{3.14}$$

The second constraint ensures that $s^{\mathsf{co}}_{p,r}$ is lower bounded by the total assignment probability allocated to reviewer r and all reviewers in r's co-authorship neighborhood.

We then add the following term to the objective:

$$\mathcal{O}^{\mathsf{co}} - \sum_{p \in \mathcal{P}} \sum_{r \in \mathcal{R}} \lambda_{\mathsf{co}}\, s^{\mathsf{co}}_{p,r}. \tag{3.15}$$

These variables and constraints together define Ξ^{co}.

This penalty discourages assigning direct past coauthors to the same paper and also indirectly penalizes reviewer pairs at co-authorship distance 2, since such pairs share a common reviewer in their neighborhoods. As a result, it helps avoid assigning reviewers drawn from tightly connected subcommunities.

Moreover, we observe empirically that applying this penalty only to reviewer–paper pairs (p, r) for which reviewer r submits a positive bid on paper p substantially sparsifies the optimization problem without noticeably degrading solution quality, and we adopt this strategy in our implementation.

2-cycles. The 2-cycle component $\mathcal{O}^{\mathsf{cyc}}$ penalizes bid-based 2-cycle violations. Ideally, pairs of reviewers who bid positively on each other's papers should not be assigned to review those papers, providing an additional safeguard against bid-based collusion.

Let $\mathcal{C}$ denote the set of all 2-cycles (r_1, r_2, p_1, p_2) such that reviewer r_1 bids positively on reviewer r_2's paper p_2 and reviewer r_2 bids positively on reviewer r_1's paper p_1. We introduce a penalty parameter $\lambda_{\mathsf{cyc}} \geq 0$ and add the following term to the objective:

$$\mathcal{O}^{\mathsf{cyc}} - \sum_{(r_1,r_2,p_1,p_2)\in\mathcal{C}} \lambda_{\mathsf{cyc}} \left(x_{p_2,r_1} + x_{p_1,r_2}\right)^2. \tag{3.16}$$

The auxiliary variables introduced when linearizing this quadratic term, together with their associated linear constraints, are included in Ξ^{cyc}.

The quadratic expression $\left(x_{p_2,r_1} + x_{p_1,r_2}\right)^2$ is convex, and therefore the penalization term $-\lambda_{\mathsf{cyc}}\left(x_{p_2,r_1} + x_{p_1,r_2}\right)^2$ is concave, preserving the overall concavity of the objective.

Expanding the square yields

$$\left(x_{p_2,r_1} + x_{p_1,r_2}\right)^2 = x_{p_2,r_1}^2 + 2x_{p_2,r_1}x_{p_1,r_2} + x_{p_1,r_2}^2,$$

where

- the cross term $2x_{p_2,r_1}x_{p_1,r_2}$ captures the probability that both assignments occur simultaneously, which is precisely the behavior we seek to penalize;
- the remaining squared terms depend only on individual assignment variables and introduce no additional interactions. These terms can be absorbed into the per-assignment perturbed-maximization objective f, increasing the penalty on large marginal assignment probabilities for reviewer–paper pairs involved in a 2-cycle without altering the structure of the optimization problem.

Seniority coverage. At the scale of large modern-day conferences, program chairs may require that each paper receive at least one experienced reviewer to ensure the quality and reliability of the reviews. While this objective could in principle be implemented in a manner similar to the geographic diversity constraints above, it is often treated as a hard requirement that should be satisfied with a 100% guarantee. Accordingly, our framework is able to support this case via a simple two-stage procedure: we run the matching algorithm twice, first on the set of senior reviewers only to guarantee coverage, and then again on the set of junior reviewers to fill the remaining slots.

3.5 Piecewise Linearization of the Optimization Program

The objective function above, like that in PM [18], contains non-linear concave terms that reduce efficiency. These include the perturbed similarity terms $S_{p,r}f(x_{p,r})$ and the concave quadratic penalties in $\mathcal{O}^{\mathsf{2cycle}}$. For any scalar $z \geq 0$ that appears inside such a concave function $\varphi(z)$, we approximate $\varphi(z)$ by a piecewise-linear function $g(z)$ on segments of length $\mathsf{SegLen} > 0$. For notational simplicity, we describe the construction for a generic concave function f and a scalar variable $x \in [0, Q]$; the same construction is applied to all concave terms in the objective.

Formally, we define $g(x)$ by linearly interpolating f on the grid $\{0, \mathsf{SegLen}, 2\mathsf{SegLen}, \ldots\}$:

$$g(x) \frac{f\Big(\Big\lfloor \frac{x}{\mathsf{SegLen}} \Big\rfloor \mathsf{SegLen}\Big) \cdot \Big(\Big\lceil \frac{x}{\mathsf{SegLen}} \Big\rceil \mathsf{SegLen} - x\Big) + f\Big(\Big\lceil \frac{x}{\mathsf{SegLen}} \Big\rceil \mathsf{SegLen}\Big) \cdot \Big(x - \Big\lfloor \frac{x}{\mathsf{SegLen}} \Big\rfloor \mathsf{SegLen}\Big)}{\mathsf{SegLen}}.$$

This definition simply linearly interpolates f between the two grid points that bracket x. As $\mathsf{SegLen} \to 0$, continuity of f implies that g converges uniformly to f on $[0, Q]$.

Replacing all concave terms by their piecewise-linear approximations yields a separable concave piecewise-linear objective. Any one-dimensional concave piecewise-linear function can be written as the minimum of finitely many affine functions, so there exist an index set J and coefficients $\{a_j, b_j\}_{j \in J}$ with

$$g(x) = \min_{j \in J} \{a_j x + b_j\}.$$

For each pair (p, r), we introduce an auxiliary variable $t_{p,r}$ and enforce

$$t_{p,r} \leq a_j x_{p,r} + b_j, \quad \forall j \in J.$$

Then maximizing the linear objective $\sum_{p,r} S_{p,r}\, t_{p,r}$ is equivalent to maximizing $\sum_{p,r} S_{p,r}\, g(x_{p,r})$ under the assumption that $S_{p,r} \geq 0$, since at optimality $t_{p,r}$ will be driven to $\min_{j \in J}\{a_j x_{p,r} + b_j\} = g(x_{p,r})$. An analogous construction is used for the concave quadratic terms in $\mathcal{O}^{\mathsf{2cycle}}$, with their own auxiliary variables and affine pieces. All constraints remain linear, so the entire approximation becomes a linear program.

This construction is closely related to the piecewise-linearization used in [18]. In their setting, the authors interpret the approximation via a network-flow formulation and empirically find that directly solving the original quadratic program with Gurobi can be competitive. In contrast, we apply the same piecewise-linearization idea but solve the resulting linear program directly with Gurobi, which is more efficient in our setting.

In our implementation, we use $\mathsf{SegLen} = 0.1$. An ablation study on this approximation is provided later.

3.6 Constraint-aware Sampling

In the previous subsection, the optimization program produces a fractional (probabilistic) assignment $x_{p,r}$. The next step is to generate a final integral assignment based on these marginal probabilities. Following [5, 18], we employ the Birkhoff–von Neumann (BVN) decomposition for sampling, applied to the bipartite graph induced by the nonzero entries of x.

Among our soft constraints, those induced by $\mathcal{O}^{\mathsf{div}}$ and $\mathcal{O}^{\mathsf{coauthor}}$ are specifically designed to promote diversity and reduce collusion. However, the stochastic nature of sampling can lead to additional violations of these constraints. For example, if two reviewers from the same region each have probability 0.5 of being assigned to a paper p, both may end up being selected after sampling.

To mitigate such effects, we modify the BVN sampling procedure. The BVN algorithm iteratively finds a loop

$$(r_0 \to p_0 \to r_1 \to p_1 \to \cdots \to r_{\ell-1} \to p_{\ell-1} \to r_0),$$

where, for each $i \in \{0, 1, \ldots, \ell - 1\}$, both x_{p_i,r_i} and $x_{p_i,r_{(i+1) \bmod \ell}}$ remain fractional. A step size Δ is then drawn in the same manner as in Algorithm 1 of [5]. The value of Δ may be positive or negative. Each x_{p_i,r_i} is updated by $+\Delta$ and each $x_{p_i,r_{(i+1) \bmod \ell}}$ is updated by $-\Delta$. After these adjustments are applied to all edges in the loop, the algorithm proceeds to the next iteration.

The original BVN algorithm does not specify how to select the loop at each iteration; any valid loop suffices for correctness. The simplest implementation, as in [5], uses a depth-first search (DFS) to find such a loop. In our variant, we bias this DFS toward improving diversity. When extending a partial loop

$$(r_0 \to p_0 \to r_1 \to p_1 \to \cdots \to r_k \to p_k),$$

and searching for the next reviewer r_{k+1} adjacent to p_k, we proceed as follows:

1. First, we give priority to reviewers r_{k+1} who are past coauthors of r_k (i.e., $r_{k+1} \in \mathcal{N}(r_k)$);
2. If no such reviewer is available, we prioritize reviewers from the same diversity region as r_k (i.e., $\text{region}(r_{k+1}) = \text{region}(r_{r_k})$).

We prioritize co-authorship distance over region similarity because, intuitively, two past coauthors are more similar than two reviewers who merely share a region (e.g., geographic location).

This biased sampling method naturally reduces the likelihood that two similar reviewers are assigned to the same paper. To illustrate this, consider again the example of two reviewers from the same region, each having a probability of 0.5 of being assigned to paper p. One approach is to increase one reviewer's assignment probability by 0.5 while decreasing the other's by 0.5. This ensures that exactly one of the two reviewers is assigned to the paper, and therefore no diversity violation can occur. By contrast, suppose we instead increase or decrease both reviewers' probabilities by 0.5 together, each with probability 0.5. A diversity violation can occur only in the case where both probabilities are increased, so we have that the constraint is violated with probability $1/4$

A natural question is why we do not adopt the sampling algorithm from [5], which introduces the notion of institutions and ensures, through its sampling procedure, that two reviewers from the same institution are not assigned to the same paper.That algorithm, however, is designed to handle only a single constraint (institutional diversity) and does not easily accommodate multiple simultaneous constraints. In contrast, our sampling framework flexibly supports multiple soft diversity constraints.

The pseudocode for the algorithm is as below. This procedure is built upon the sampling framework of Jecmen et al. [5].

The core mechanism involves iteratively decomposing the fractional assignment matrix into paths and cycles to shift probability mass until an integral solution is reached. Our key modification to the original algorithm lies in the search subroutine: whereas the standard approach selects paths and cycles arbitrarily, our method utilizes an attribute-aware search (Algorithm 2). This subroutine prioritizes linking "similar" reviewers (e.g., past co-authors or those from the same region) within the same update step. By coupling the rounding decisions for these reviewers, the algorithm minimizes the variance that leads to violations of soft diversity and co-authorship constraints.

Algorithm 1 presents the main iterative rounding loop, and Algorithm 2 details the priority-based Depth-First Search (DFS) strategy.

Note that if conference organizers wish to impose additional diversity constraints, these can be incorporated into the same sampling framework with minimal modification.

Algorithm 1 Attribute-Aware Sampling (Path or Cycle)

Input: Fractional assignment matrix $\mathbf{x} \in [0,1]^{|\mathcal{P}|\times|\mathcal{R}|}$.
Output: Integral assignment matrix $\mathbf{X} \in \{0,1\}^{|\mathcal{P}|\times|\mathcal{R}|}$.
while exists (p, r) such that $0 < x_{p,r} < 1$ **do**
  Construct the bipartite graph $G = (\mathcal{P} \cup \mathcal{R}, E)$ where $E = \{(p, r) \mid 0 < x_{p,r} < 1\}$
  $C \leftarrow$ FindAttributeAwareChain$(G, \mathbf{x})$ (See Algorithm 2)
  Partition edges of C into C_{odd} and C_{even} based on their position in the sequence.
  Calculate maximum feasible steps:
    $\alpha \leftarrow \min\big(\{1 - x_e \mid e \in C_{odd}\}$
      $\cup\{x_e \mid e \in C_{even}\}\big)$
    $\beta \leftarrow \min\big(\{x_e \mid e \in C_{odd}\}$
      $\cup\{1 - x_e \mid e \in C_{even}\}\big)$
  if C is a Path starting at u and ending at v **then**
    Update α, β ensuring reviewer load constraints at u, v are not violated.
  end if
  With probability $\frac{\beta}{\alpha+\beta}$:
    $x_e \leftarrow x_e + \alpha$ for $e \in C_{odd}$
    $x_e \leftarrow x_e - \alpha$ for $e \in C_{even}$
  Else (with probability $\frac{\alpha}{\alpha+\beta}$):
    $x_e \leftarrow x_e - \beta$ for $e \in C_{odd}$
    $x_e \leftarrow x_e + \beta$ for $e \in C_{even}$
end while
return $\mathbf{x}$

This biased sampling intuitively suppresses the probability that two "similar" reviewers are assigned to the same paper. Returning to the earlier example, where two reviewers from the same region each have a 0.5 probability of being assigned to paper p: increasing one probability by 0.5 and decreasing the other by 0.5 is clearly preferable to jointly increasing or decreasing both by 0.5 with probability 0.5. In the former case, no diversity violation occurs, whereas in the latter case, the constraint is violated with probability $1/4$.

A natural question is why we do not adopt the sampling algorithm from [5], which ensures that two reviewers from the same institution are not assigned to the same paper. That algorithm, however, is designed to handle only a single constraint (institutional diversity) and does not easily accommodate multiple simultaneous constraints. In contrast, our sampling framework flexibly supports multiple soft diversity constraints.

Note that if conference organizers wish to impose additional diversity constraints, these can be incorporated into the same sampling framework with minimal modifications

Algorithm 2 FindAttributeAwareChain (Subroutine)

Input: Bipartite graph G, Assignment $\mathbf{x}$.
Output: A sequence of edges forming a Cycle or Path.
visited $\leftarrow \emptyset$, stack $\leftarrow []$.
Selection of Start Node:
    If there exists a node v with odd degree in G (fractional load), let $v_{start} = v$.
    Else, pick arbitrary v_{start} with degree > 0.
Push v_{start} to stack.
while stack is not empty **do**
    $u \leftarrow$ stack.peek()
    $V_{\text{adj}} \leftarrow \{v \mid (u, v) \in E\} \setminus \{\text{parent}(u)\}$
    if $V_{\text{adj}} = \emptyset$ **then**
        {Path found}
        return stack as a Path.
    end if
    if $u \in \mathcal{P}$ **then**
        {**Attribute-Awareness:** Prioritize similar reviewers}
        Let r_{prev} be the reviewer preceding u in stack.
        Construct ordered list L from V_{adj} based on priority:
            1. $v \in \mathcal{N}(r_{prev})$ (Co-authors)
            2. $\text{Region}(v) = \text{Region}(r_{prev})$ (Same Region)
            3. Others
    else
        {Reviewer node: standard traversal}
        Construct list L from V_{adj}
    end if
    Pick first v in L.
    if $v \in$ visited **then**
        {Cycle found}
        Extract cycle portion from stack (from first occurrence of v to u).
        return Cycle.
    else
        Push v to stack, mark v visited.
    end if
end while

Chapter 4

Evaluation of the Framework

4.1 Datasets

To evaluate the proposed framework, I follow the experimental design of the source paper and consider four datasets spanning both real and synthetic conference-review settings. The purpose of this evaluation is not only to compare assignment quality, but also to study how diversity, robustness, and runtime behave under conference-scale conditions.

The first dataset, denoted **Large**, is a synthetic large-scale conference instance constructed to reflect the scale and sparsity structure of modern major computer science venues. In the source paper, this dataset contains $20{,}000$ papers and $22{,}000$ reviewers, and is designed to model the large assignment instances now encountered in practice [19]. In addition to this synthetic large-scale benchmark, the paper evaluates on two smaller real-world conference datasets, **AAMAS 2015** and **ICLR 2018**, both of which have been used in prior work on reviewer assignment [18]. A fourth dataset, **S2ORC**, is derived from a synthetic conference construction based on the Semantic Scholar Open Research Corpus and is particularly useful for studying collusion-related structures and bidding behavior [16].

These datasets jointly support several goals. The smaller datasets allow comparison with prior methods under more classical benchmark settings, while the Large dataset tests whether the framework remains computationally feasible at modern conference scale. The S2ORC dataset is especially valuable because it provides a setting in which robustness-related phenomena such as bid-based collusion can be modeled more explicitly [16].

For datasets that do not originally contain all reviewer metadata required by the framework, such as diversity attributes or bid information, the source paper augments them using synthetic generation procedures aligned with the large-scale benchmark construction. This allows the evaluation to compare algorithms on a common set of quality, diversity, and robustness metrics.

4.2 Baselines

The source paper compares the proposed framework, Robust Assignment via Marginal Perturbation (RAMP), against four baselines representing different points in the reviewer-assignment literature.

Dataset	# Papers	# Reviewers	Source
Large	20,000	22,000	Synthetic
AAMAS 2015	601	213	Real
ICLR 2018	911	2,435	Real
S2ORC	2,446	2,483	Synthetic

Table 4.1: Datasets used in evaluation

The first baseline is the **Default** assignment method, namely the classical similarity-maximizing linear-program formulation with capacity and conflict constraints. This baseline is the strongest reference point for pure matching quality, since it optimizes reviewer expertise without introducing additional randomness or structural penalties [2, 3, 10, 14, 15].

The second baseline is the mixed-integer linear programming approach, labeled **MILP**, based on prior work that introduces explicit diversity and anti-collusion constraints into the assignment formulation [9]. This baseline is important because it represents a direct constraint-based alternative to RAMP. However, as emphasized in the source paper, its main drawback is computational cost at larger scales.

The third baseline is **PLRA**, or Probability-Limited Randomized Assignment, which introduces controlled randomness by limiting pairwise marginal assignment probabilities and then sampling a deterministic matching from the resulting distribution [5]. PLRA is especially relevant as a robustness-oriented baseline.

The fourth baseline is **PM**, or perturbed maximization, which generalizes the PLRA perspective by replacing the linear similarity objective with a concave perturbation that encourages spreading assignment probability across multiple strong reviewers [18]. PM is particularly relevant because RAMP builds directly on this perturbed-objective viewpoint.

Taken together, these baselines provide a useful spectrum. Default emphasizes quality, MILP emphasizes explicit constraints, PLRA emphasizes randomized robustness, and PM emphasizes randomized robustness with a richer objective. RAMP is designed to combine the advantages of these lines of work while avoiding their most important limitations, especially the scalability limitations of explicit mixed-integer formulations [5, 9, 18].

4.3 Evaluation Metrics

The evaluation in the source paper uses metrics that reflect the multiple goals of modern reviewer assignment rather than only total matching quality.

The first metric is **Quality**, which measures the overall objective value of the produced assignment relative to the best achievable quality, typically represented by the Default assignment. This metric captures how much matching utility is preserved when diversity and robustness objectives are introduced.

The second metric is **Runtime**, which measures the computational time required to solve the assignment problem. Runtime is critical because a method that performs well on paper but cannot be executed at conference scale is of limited practical use.

The third metric is **Diversity**, computed as the average number of distinct regions represented among the reviewers assigned to each paper. This directly measures the success of the diversity objective introduced in the framework.

Two further metrics capture structural robustness. **Coauthors** counts the number of coauthor pairs assigned to the same paper, and **2-Cycles** counts the number of bid-based reciprocal structures discussed earlier in the thesis. Lower values on these metrics indicate stronger protection against undesirable reviewer groupings and collusion-related patterns [9, 16].

The source paper also reports two metrics related to randomness: **Support Size**, the number of paper–reviewer pairs with positive marginal probability, and **Entropy**, which summarizes how dispersed the fractional assignment is. These metrics are useful because one of the main goals of randomized assignment is to increase uncertainty and reduce the concentration of assignment probability on a small number of deterministic matches [5, 18].

Together, these metrics give a more complete picture of assignment behavior than matching quality alone. They make it possible to analyze the tradeoffs among quality, diversity, robustness, and computational efficiency that are central to the source paper and to this thesis.

4.4 Comparison with Prior Methods

Table 4.2 compares RAMP against four baselines on the Large dataset. RAMP is the only method that achieves substantial improvements over Default across all robustness-related metrics as well as all diversity-related metrics, including coauthor pairs, two-cycles, and geographic diversity.

In addition, RAMP completes the large-scale experiment in under 20 minutes, representing an 89% reduction in runtime compared to MILP, which is the only baseline that also performs strongly across all robustness-related metrics. These gains are achieved while incurring only a 2.6% reduction in assignment Quality compared to Default. These trends are consistent across datasets.

Algorithm	**Time (s)** $\downarrow$	**Quality** $\uparrow$	**Support** $\uparrow$	**Entropy** $\uparrow$	**Div.** $\uparrow$	**Coauth.** $\downarrow$	**2Cyc.** $\downarrow$
Default	585.0	1.000	80000	0	0.615	163	86
MILP	17257.5	0.985	80000	0	0.916	0	0
PM	1765.4	0.980	779680	155316	0.616	150	117
PLRA	600.1	0.999	103576	16355	0.614	172	86
RAMP	1096.1	0.974	537266	143587	0.895	21	1

Table 4.2: Performance comparison of paper-assignment algorithms on the large synthetic dataset. Metrics marked with $\uparrow$ indicate that higher values are preferred, while those marked with $\downarrow$ indicate that lower values are preferred.

We postpone the display of experiment results on other datasets to 4.8, but the results found are very similar to 4.2

4.5 Hyperparameters, Quality, and Constraint Satisfaction

One natural question is: how do the choices of hyperparameters for our soft constraints impact key attributes such as quality, runtime, and how well different constraints are satisfied? Table 4.3 presents the the trade-offs induced by tuning hyperparameters λ_* in soft constraints on S2ORC: we start with all hyperparameters set to 0, and tune individual hyperparameters.

As discussed earlier, some conferences may want taht each paper receive at least one "senior" or experienced reviewer. This can be handled by the Diversity constraint that we talked about earlier as well. The attribute Seniority then measures the amount of papers with at least one senior reviewer.

Increasing individual penalties improves the targeted constraint with only a minor impact on quality and runtime. For example, increasing λ_{cyc} eliminates two-cycles with less than a 0.2% quality loss. We first set all hyperparameters to zero, and tune the parameters one at the time. Note that quality is relative to the default algorithm on the same dataset.

Setting	**Time (s)** $\downarrow$	**Quality** $\uparrow$	**Support** $\uparrow$	**Entropy** $\uparrow$	**Div.** $\uparrow$	**Seniority** $\uparrow$	**Coauth.** $\downarrow$	**2Cyc.** $\downarrow$
all $= 0.0$	113.047	0.977	41,231	12,308.055	0.835	0.822	76	101
$\lambda_{\text{div}} = 0.2$	124.968	0.971	43,373	12,863.558	0.949	0.816	55	88
$\lambda_{\text{co}} = 0.2$	113.043	0.976	41,392	12,362.858	0.832	0.825	30	104
$\lambda_{\text{cyc}} = 0.15$	112.330	0.975	41,351	12,292.501	0.846	0.822	80	0

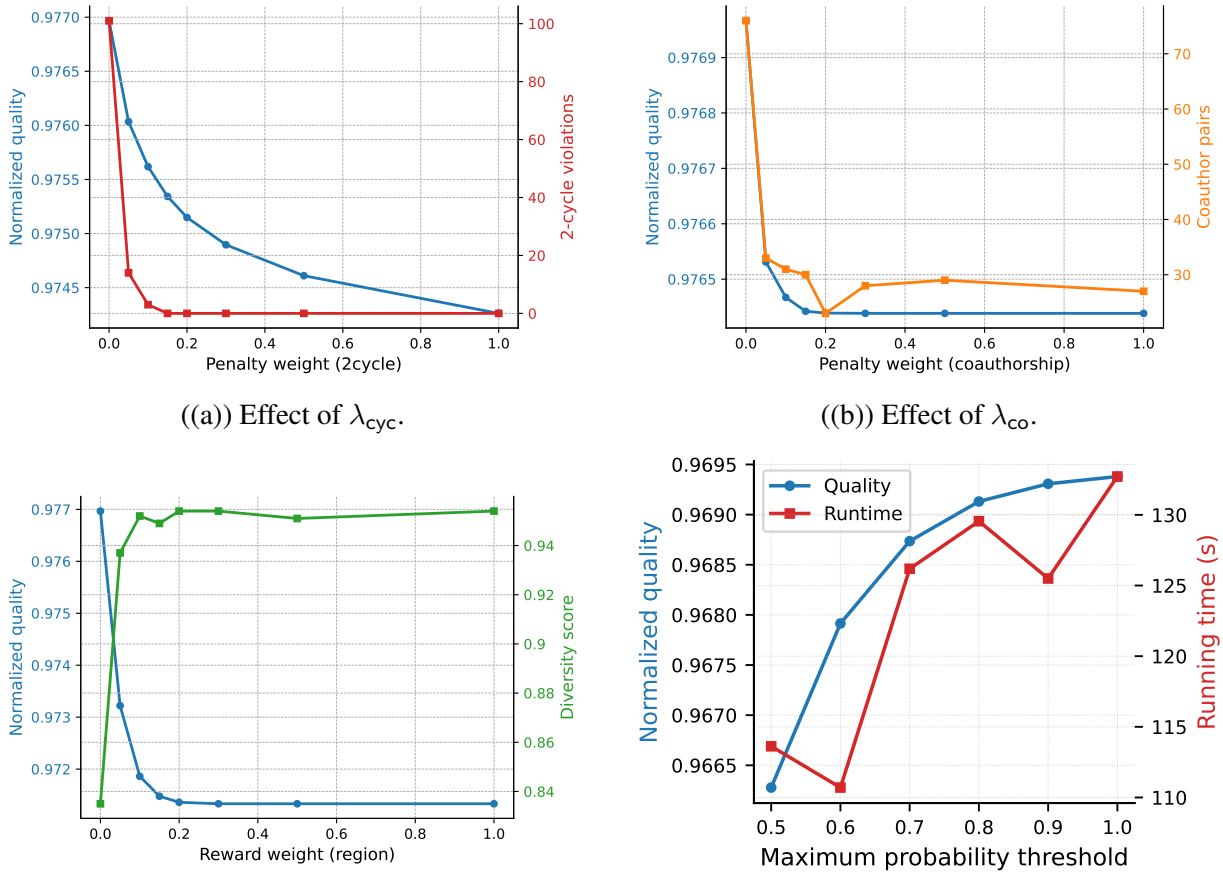
Table 4.3: Hyperparameter tuning on the S2ORC dataset. Arrows indicate desired direction for each metric: higher ($\uparrow$) or lower ($\downarrow$).

As discussed earlier, the tradeoffs between quality and individual hyperparameters are illustrated in Figures 4.1(a), 4.1(b), and 4.1(c). Normalized quality refers to the quality achieved by the algorithm relative to the maximum attainable quality, which is obtained using the Default algorithm. The fluctuations observed in these curves arise from the randomized nature of the algorithm, which produces different matches between runs.

In addition to the parameters above, the hyperparameter Q can also be tuned. Results are plotted in Figure 4.1(d) Experiments on the S2ORC dataset indicate that both runtime and solution quality increase as Q grows; however, these effects are negligible in magnitude. In practice, program chairs are therefore more likely to tune Q to mitigate collusion concerns rather than to optimize runtime or quality. The choice of perturbation function is similarly tunable and was explored extensively in [18].

4.6 Ablation Studies

Piecewise-linear Approximation. To assess the contribution of the piecewise-linear (PWL) approximation, we compare our full algorithm against a variant that operates directly on the nonlinear objective and constraint functions, without any linearization. Both versions were evaluated on the Large dataset, using identical hyperparameter settings to ensure a fair comparison. The quantitative results are presented in Table 4.4, with only the most relevant attributes.



((a)) Effect of λ_{cyc}.

((b)) Effect of λ_{co}.

((c)) Effect of λ_{div}.

((d)) Effect of Q.

Figure 4.1: Hyperparameter trade-offs on the S2ORC dataset.

Table 4.4 shows that removing the PWL approximation increases runtime by over $16\times$, while leaving assignment quality and constraint metrics essentially unchanged. This confirms that PWL is critical for computational efficiency without sacrificing solution quality.

Attribute-aware sampling algorithm. We examine the impact of the attribute-aware sampling algorithm introduced in Section 3.6. We conducted experiments on the Large dataset, comparing our method with and without attribute-aware sampling enhancement. Similarly, both variants were run under identical hyperparameter settings to ensure a controlled comparison. Table 4.5 summarizes the results across several key metrics, while other metrics showed no significant difference.

As shown, attribute-aware sampling leaves runtime and quality unchanged, but substantially improves some diversity-related metrics. In particular, geographic diversity increases from 0.697 to 0.896, and coauthor pairs drop from 199 to 26, demonstrating that the sampling procedure effectively enforces diversity constraints without affecting efficiency.

Variant	Time (s) ↓	Quality ↑	Support ↑	Entropy ↑	Diversity ↑	Coauthors ↓	2Cycles ↓
PWL	**1088.9**	1.0	537266	143587	0.896	26	0
Non-PWL	16569.9	0.999	**825421**	158028	0.893	27	2

Table 4.4: Full comparison of PWL approximation vs non-PWL variant on the large synthetic dataset. PWL achieves similar quality with much lower runtime

Method	Time (s) ↓	Quality ↑	Support ↑	Entropy ↑	Diversity ↑	Coauthors ↓	2Cycles ↓
Attribute-Aware	1088.9	1.0	537266	143587	**0.896**	**26**	0
Vanilla	1083.2	1.0	537266	143587	0.697	199	0

Table 4.5: Comparison of attribute-aware sampling vs. vanilla sampling on the large synthetic dataset. Quality values are normalized to 1.0. Attribute-aware sampling improves diversity and reduces coauthor overlaps while maintaining identical quality.

4.7 Handling Seniority Requirements

Another desired property is to have at least one senior reviewer reviewing every paper.

Earlier, we proposed handling seniority in the same manner as geographic diversity, namely through soft constraints. An alternative approach is a two-stage algorithm, in which senior reviewers are assigned first, followed by junior reviewers. This strategy is widely used in practice by many conferences and has been discussed previously in [9].

From Table 4.6, we observe that although the soft-constraint approach achieves marginally better solution quality and runtime, it is less effective at satisfying seniority requirements. Modern large-scale conferences, however, may impose strict seniority constraints, preferring to assign at least one senior reviewer to each paper even at the cost of slightly reduced quality or increased runtime.

Method	Running time (s)	Quality	Support size	Diversity	Seniority	Coauthors
$\lambda_{\text{sen}} = 0.2$	**122.733**	1.000	43,645	0.949	0.815	23
Two-stage	142.080	0.998	42,946	0.971	**1.000**	28

Table 4.6: Comparison of single-stage and two-stage methods for handling seniority requirements on the S2ORC dataset

4.8 Extra Results

In addition to the large synthetic dataset reported in Table 4.2, we evaluated the same set of algorithms on the ICLR, S2ORC, and AAMAS datasets. Table 4.7, Table 4.8, and Table 4.9 present the full metrics, including runtime, quality, support size, entropy, diversity, coauthor counts, and 2-cycle counts. All metrics follow the same convention as Table 4.2: metrics with

↑ indicate that higher values are preferred, while metrics with ↓ indicate that lower values are preferred.

Overall, the relative performance of the algorithms is consistent with the observations in Table 4.2. In particular, RAMP maintains competitive quality while improving diversity and reducing coauthor overlaps and 2-cycles compared to baseline methods.

Algorithm	**Time (s)** ↓	**Quality** ↑	**Support** ↑	**Entropy** ↑	**Div.** ↑	**Coauth.** ↓	**2Cyc.** ↓
Default	26.006	1.0	3644	0.0	0.735	19	1
MILP	66.945	0.993	3644	0.0	0.900	4	0
PM	81.846	0.988	12956	3339.650	0.727	21	1
PLRA	27.303	0.995	4585	692.393	0.731	22	2
RAMP	34.115	0.982	12041	3381.906	0.918	6	0

Table 4.7: Comparison of algorithms on the ICLR dataset.

Algorithm	**Time (s)** ↓	**Quality** ↑	**Support** ↑	**Entropy** ↑	**Div.** ↑	**Coauth.** ↓	**2Cyc.** ↓
Default	77.031	1.0	9784	0.0	0.733	329	170
MILP	174.841	0.980	9784	0.0	0.990	8	12
PM	221.858	0.978	49113	12643.684	0.741	242	150
PLRA	74.435	0.997	12676	2009.879	0.737	323	162
RAMP	128.582	0.969	43645	12901.315	0.951	19	1

Table 4.8: Comparison of algorithms on the S2ORC dataset.

Algorithm	**Time (s)** ↓	**Quality** ↑	**Support** ↑	**Entropy** ↑	**Div.** ↑	**Coauth.** ↓	**2Cyc.** ↓
Default	3.096	1.0	2104	0.0	0.746	18	13
MILP	4.720	0.991	2104	0.0	0.872	0	7
PM	8.773	0.979	5609	1422.213	0.743	29	11
PLRA	3.136	0.992	2655	403.450	0.752	21	13
RAMP	4.530	0.965	5650	1628.143	0.930	3	1

Table 4.9: Comparison of algorithms on the AAMAS dataset.

Chapter 5

Deployment

A central goal of this thesis is not only to study reviewer assignment as an optimization problem, but also to understand how such methods behave in real conference workflows. To that end, the algorithm developed in this thesis was run for several major conference matching processes: AAAI 2026 during August through October 2025, AAMAS 2026 during October through November 2025, and EC 2026 during January through March 2026. These deployments provided an opportunity to evaluate the framework under realistic operational constraints rather than only in offline experiments.

In practice, deployment required substantially more than solving an optimization problem on a fixed dataset. Each conference had its own timeline, review structure, organizer preferences, data pipeline, and exceptional constraints. As a result, the matching process was not a single one-shot computation, but an iterative workflow involving repeated runs, parameter adjustments, data cleaning, and discussion with organizers. In several cases, the most important questions were not purely algorithmic, but organizational: which constraints should be treated as hard requirements, which should be represented softly, and which should remain subject to manual review.

5.1 Sparsification

In large conferences, the similarity matrix S is extremely sparse in terms of "useful" entries. To exploit this, our algorithm restricts attention to a sparse subset of candidate paper–reviewer pairs.

For starters, we only initialize assignment variables $x_{p,r}$ where $S_{p,r}$ is available or that the paper is bid on. This means that when solving the linear program, there will not be a massive amount of assignment variables.

Furthermore, we can perform the following, for each paper $p \in \mathcal{P}$, we retain only the top K_{paper} reviewers ranked by $S_{p,r}$, and for each reviewer $r \in \mathcal{R}$, we retain only the top K_{rev} papers. The union of these pairs defines the support of the assignment variables $x_{p,r}$; all remaining pairs are removed from the optimization program.

In our experiments, we set $K_{\text{paper}} = K_{\text{rev}} = 1000$. For a conference-scale instance with roughly $20{,}000$ papers and reviewers, this yields a highly sparse problem in which the number of active variables is several orders of magnitude smaller than $|\mathcal{P}| \cdot |\mathcal{R}|$. Accordingly, we include only these variables and the constraints involving them in the solver.

By comparison, the NeurIPS 2024/2025 paper-matching system applies a similar sparsification to similarity scores but retains all variables in the optimization, treating non-retained pairs as zero-weighted entries. Explicitly removing these variables leads to substantial reductions in both memory usage and runtime in our setting.

In large conferences, it might also be the case that some reviewers or papers do not have a similarity score with other papers or reviewers. However, in our algorithm, we only initialize assignment variables $x_{p,r}$ for p, r that have a non-negative similarity score or bid for the sake of saving memory and decreasing runtime. While running the algorithm for AAAI and AAMAS, this led to problems.

The solution was rather straightforward. Before we built the similarity matrix, for each paper, we randomly sample k reviewers with with these papers have a ϵ similarity score boost, and we do the same for each paper. This basically adds an edge to consider. In AAAI, we used $k = 500$; even so, its effects on runtime were minimal.

5.2 Subject Area Scores

In AAAI, one of the problems we encountered after Phase 1 was that reviewers complained about being matched with papers not directly related to their area of interest. After some more digging, we realized that this was because the OpenReview similarity scores, though text-based, do not effectively capture subject area information of papers and reviewers.

For Phase 2, our solution was to change the way similarity scores are calculated. We would still use the text-based similarity score from OpenReview, but we would also use a subject area score derived the subject area information that paper submissions contained, as well as reviewer subject area that was scraped from OpenReview. The score calculations were done using Jaccard similarity, where p is paper area and r is reviewer area.

$$S_{\text{Jaccard}}(p, r) = \frac{|p \cap r|}{|p \cup r|} = \frac{\sum_i p_i r_i}{\sum_i \mathbf{1}[p_i = 1 \text{ or } r_i = 1]}$$

We also realized that this was not the case for other conferences such as NeurIPS and AAMAS because their sizes were smaller, and all papers were in similar areas. In the future, for larger conferences, we suggest requiring papers and reviewers to submit a desired subject area.

5.3 Bids, Similarity Score, and Collusion

A much talked about topic was bidding, and how much of a role bidding should play. The general rules were: if a conference is smaller scale and a tighter community, bids should generally be trusted and honored. Then the calculations for the aggregate score would be

$$\text{aggregate score} = \alpha * \text{similarity score} + (1 - \alpha) * \text{bid score}$$

, where $\alpha \in [0, 1]$. Bidscore values would be -1.0, 0.5, 0.0, 0.5, 1.0 for *not willing*, *not entered*, *in a pinch*, *willing*, and *eager*.

In larger conferences such as AAAI, we combine reviewer expertise with reviewer interest by modifying the base expertise score using the reviewer's bid. The resulting aggregate score is defined as

$$\text{aggregate score} = (\text{base aggregated score})^{\text{bidscore}}.$$

The bidscore values are set to 20, 1, 0.67, 0.4, and 0.25, corresponding respectively to *not willing*, *not entered*, *in a pinch*, *willing*, and *eager*. Since the base score lies in $[0, 1]$, exponents greater than 1 decrease the score, exponents less than 1 increase it, and an exponent of 1 leaves it unchanged. In this way, bids can influence the final assignment score without fully overriding reviewer expertise: a reviewer who is manifestly unqualified cannot become highly ranked solely by bidding, while bids can still meaningfully distinguish among reviewers who are already reasonably qualified.

Another way we can mitigate collusion is to make the following observation: reviewers can only collude if they bid on each other's papers. Since we can cap individual assignment variables $x_{p,r} \in [0, 1]$ at some maximum probability $Q \in [0, 1]$, we can choose to cap $x_{p,r}$ where reviewer r bid on p so that bids do play a role, but we limit the probability that they are assigned. For $x_{p,r}$ where r does not bid on p, we can simply keep $Q = 1$, since we are not as worried about collusion in the case where bidding does not happen.

5.4 Others

While performing matchings for these conferences, we realized that the optimal values of these hyperparameters ($\lambda_{\mathsf{div}}, \lambda_{\mathsf{co}}$, ...) were very much data dependent. The scale of aggregate scores depend on how similarity score, subject area, and bids were processed. As such, it is often the case that many rounds of tuning need to happen to find a comfortable spot on the pareto frontier.

Other special requests can also be handled by the framework. For instnace, while performing matching for EC, they may desire that reviewers for a paper are diverse - some have backgrounds in economics and others in computer science. This can be handled by adjusting diversity: instead of using a score for geographic regions, we can use subject areas.

Chapter 6

User Interface for Paper Matching

The previous chapters describe the optimization framework and its evaluation. In practice, however, deploying a paper-matching algorithm requires more than solving the optimization problem. Conference organizers must prepare input files, choose hyperparameters, run the solver, inspect diagnostic information, and compare candidate assignments. These steps can be difficult for users who are not familiar with the codebase or with the details of the optimization formulation.

To reduce this deployment burden, we implemented a local web interface for running the paper-matching pipeline. The interface is not intended to replace the matching algorithm. Instead, it exposes the existing backend in a more accessible and inspectable form. It allows users to upload data, configure algorithmic settings, launch matching runs, monitor execution, and inspect the resulting outputs. The interface discussed in this section can be found in this GitHub repo.

6.1 Motivation

The deployment experience described in Chapter 5 showed that usability is an important part of practical paper assignment. Even if an algorithm produces good assignments, it may be difficult for conference organizers to use if it requires manual editing of configuration files, direct command-line interaction, or detailed knowledge of solver settings. In addition, real deployments often require several runs with different parameters before organizers are satisfied with the final assignment.

This is especially important for RAMP because the framework exposes several meaningful choices. Organizers may need to choose whether to include diversity rewards, coauthorship penalties, 2-cycle penalties, seniority handling, bid-based marginal caps, or different sparsification settings. These choices affect the tradeoff between assignment quality, robustness, diversity, and runtime. Without an interface, it is difficult for non-specialist users to understand these choices and compare the resulting assignments.

The purpose of the interface is therefore to make the matching workflow easier to run and easier to inspect. Rather than requiring users to interact directly with the research code, the interface provides a structured workflow for configuring and executing the matcher.

6.2 Design Goals

The interface was designed around four main goals.

First, it should reduce setup burden. Users should be able to upload the required CSV files or select a bundled example dataset without modifying the backend code. The interface should then generate the necessary configuration files automatically.

Second, it should make the algorithmic parameters more understandable. Parameters such as the marginal probability cap, diversity reward, coauthorship penalty, and 2-cycle penalty should be visible in one place and accompanied by short descriptions. This helps users understand what they are changing when they adjust a parameter.

Third, it should support iterative experimentation. In deployment, it is common to run the matcher multiple times with different settings. The interface therefore stores recent runs, records the settings used for each run, and presents output summaries so that users can compare candidate assignments.

Fourth, it should improve transparency. The interface displays execution logs, summary statistics, diagnostic plots, and output files using descriptive labels. This helps users distinguish between intermediate outputs, such as fractional assignment probabilities, and the final sampled assignment used in practice.

6.3 System Workflow

The interface follows the main stages of the matching pipeline:

Upload data $\rightarrow$ Configure run $\rightarrow$ Solve optimization problem $\rightarrow$ Sample assignment $\rightarrow$ Inspect outputs.

In the upload stage, the user provides the data needed to construct the matching instance. These inputs may include paper metadata, reviewer metadata, similarity scores, conflicts of interest, bids, reviewer capacities, and reviewer attributes such as region or seniority. The interface explains the expected format of each file so that users can identify missing or incorrectly formatted inputs.

In the configuration stage, the user selects the algorithmic settings for the run. These include the optimization backend, sparsification settings, probability caps, and weights for the soft objectives. For example, a user can increase the diversity reward to encourage more diverse reviewer sets, or increase the coauthorship penalty to discourage assigning past coauthors to the same paper.

After the configuration is submitted, the interface generates the backend configuration file and invokes the matching pipeline. The backend constructs the optimization instance, solves the fractional assignment problem, and then runs the sampling procedure to produce an integral assignment. The interface monitors this process and displays logs so that users can see whether the run is still executing, has finished successfully, or has failed.

Finally, the interface presents the resulting artifacts. These include the final assignment, fractional assignment probabilities, solver logs, summary statistics, and diagnostic outputs. Presenting these files through the interface makes the matching process more inspectable than simply returning a directory of raw output files.

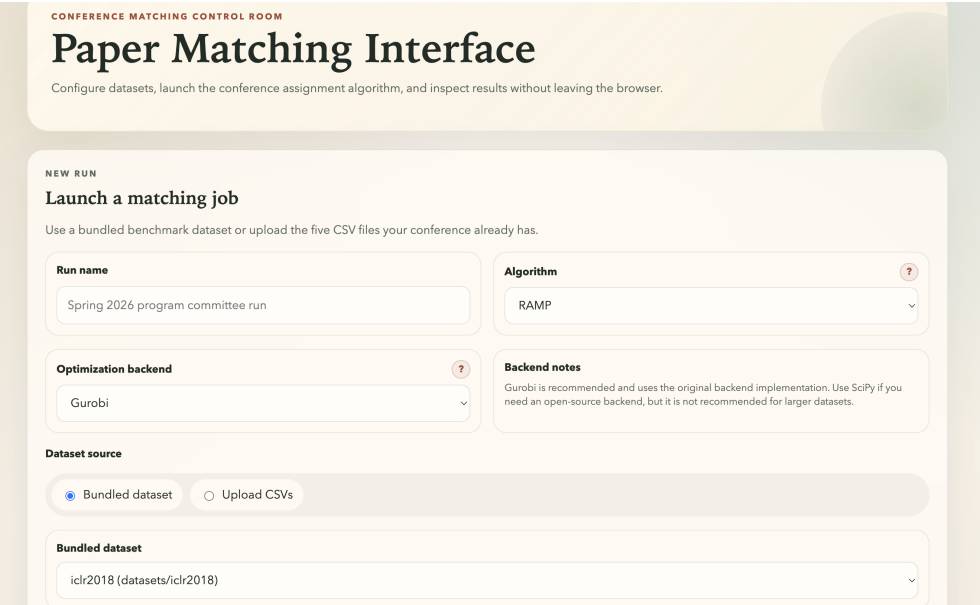


Figure 6.1: User selects algorithm and uploads dataset here

6.4 Implementation

The interface is implemented as a local web application layered on top of the existing matching backend. This design preserves consistency with the core implementation: the same backend code is used whether the matcher is launched manually or through the interface. The interface mainly handles input collection, configuration generation, process execution, monitoring, and output presentation.

The primary solver backend is Gurobi, which remains the recommended option for large datasets and complex formulations. Since Gurobi is a commercial solver, the interface also supports a SciPy-based backend for smaller-scale experiments and settings where commercial solver infrastructure is not available. However, the SciPy backend is not intended to replace Gurobi for large conference-scale instances, where solver scalability is a central concern.

The interface also supports limited experiment management. Each run is stored with its configuration and output summaries. This makes it easier to compare runs with different hyperparameter settings. For example, a user can compare a run with a higher diversity reward against a run with a stronger coauthorship penalty and inspect how the metrics change.

6.5 Discussion

The interface serves as a systems component that complements the optimization framework studied in this thesis. It demonstrates how a research-oriented matching algorithm can be made more usable for operational workflows. In particular, it helps reduce setup burden, exposes important hyperparameters, supports repeated experiments, and makes the outputs of the matcher easier to inspect.

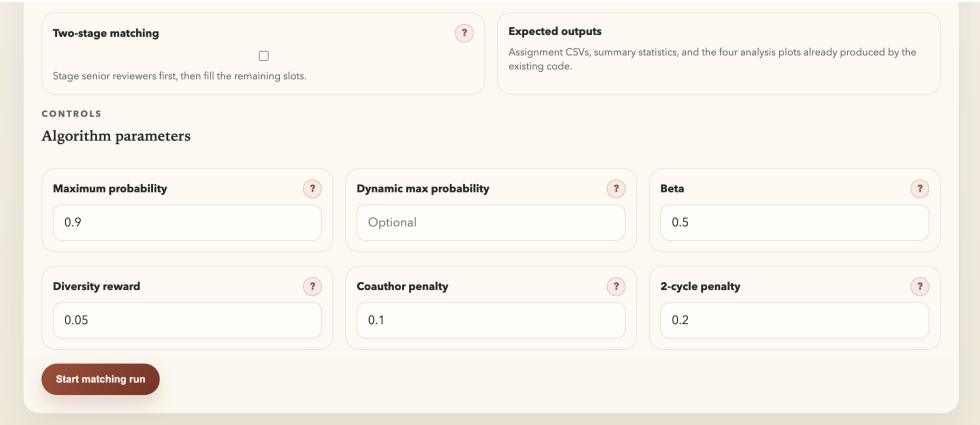


Figure 6.2: User can configure hyperparameters, and refer to the question mark if unsure about any of them

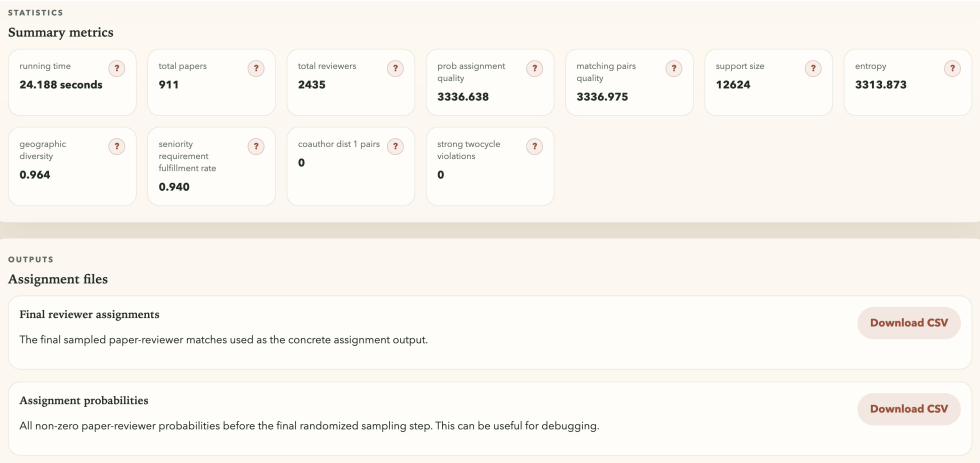


Figure 6.3: When the run finishes, user receives a summary statistic as well as the output files. Plots are also included in the interface (though not displayed).

At the same time, the interface is only an initial step toward a complete deployment system. It still assumes that users can prepare input files in the expected format, and it does not yet integrate directly with conference-management systems such as OpenReview. As a result, some manual data preparation is still required. A natural future direction is to connect the interface more directly to conference-management platforms so that data ingestion, validation, matching, and export can be handled with less manual effort.

Chapter 7

Conclusion

7.1 Summary of Contributions

This thesis studied the problem of assigning papers to reviewers in large academic conferences. Paper assignment is an important part of the peer-review process because the quality of the final reviews depends heavily on whether papers are matched with appropriate reviewers. A simple assignment method can maximize reviewer–paper similarity while satisfying basic constraints such as reviewer capacity, paper coverage, and conflicts of interest. However, modern conferences often require more than this. Organizers may also care about reviewer diversity, seniority coverage, robustness to strategic behavior, coauthorship structure, bid-based collusion risks, and runtime.

The main goal of this thesis was to present and study RAMP, a unified framework for scalable and robust paper assignment. RAMP builds on prior work in randomized paper assignment and perturbed maximization, and extends these ideas by adding soft objectives for practical conference requirements. In particular, the framework allows objectives such as diversity, coauthorship-distance reduction, and bid-based 2-cycle avoidance to be incorporated into one optimization problem. This makes the framework more flexible than assignment methods that only optimize similarity.

A key idea in the framework is to treat some conference requirements as soft objectives rather than hard constraints. Hard constraints are still used for requirements that must always be satisfied, such as conflicts of interest, paper coverage, and reviewer load limits. Other goals, such as improving diversity or reducing coauthor pairs assigned to the same paper, are represented through rewards or penalties in the objective. This allows the assignment process to trade off matching quality against other desirable properties. In practice, this is useful because conference organizers may not want to sacrifice too much reviewer expertise in order to improve a secondary metric.

This thesis also discussed the algorithmic steps that make the framework practical for larger instances. Piecewise linearization is used to approximate nonlinear concave terms in the objective, making it possible to solve the resulting problem as a linear program. Sparsification is used to reduce the number of paper–reviewer pairs considered by the optimizer. Attribute-aware sampling is then used to convert a fractional assignment into an integral assignment while trying to

preserve some of the structure encouraged by the optimization problem.

The experimental results show that RAMP can improve diversity and reduce undesirable structures such as coauthor pairs and 2-cycles, while only slightly reducing assignment quality compared with the default similarity-maximizing baseline. The results also show that the piecewise-linear approximation is important for runtime and that attribute-aware sampling improves the final integral assignment compared with vanilla sampling. These results suggest that RAMP provides a useful middle ground: it is more expressive than simple similarity maximization, but more scalable than approaches that rely heavily on mixed-integer constraints.

Finally, this thesis discussed deployment lessons from applying these ideas to real conference workflows. These experiences show that paper assignment is not only an optimization problem. In practice, the quality of the final assignment also depends on the input similarity scores, the availability of reviewer metadata, how bids are incorporated, how missing profiles are handled, and how organizers choose hyperparameters. The interface described in Chapter 6 was included as a first step toward making the matching pipeline easier for future conference organizers to run and inspect.

Overall, this thesis shows that large-scale paper assignment can benefit from a unified framework that combines similarity maximization with robustness, diversity, and practical deployment considerations. While the framework does not solve every problem in peer-review assignment, it provides a useful foundation for building more flexible and scalable matching systems.

7.2 Future Work

There are several directions for future work.

One important direction is improving integration with conference-management systems such as OpenReview. In the current workflow, organizers or researchers may need to manually prepare files containing papers, reviewers, conflicts of interest, bids, similarity scores, and reviewer metadata. This manual process can be time consuming and error prone. A more integrated system could automatically read the necessary data from the conference platform, run the matching algorithm, and export the final assignment back to the platform. This would make the framework easier to use in real conference settings.

A second direction is improving similarity-score construction. This thesis mainly focuses on the assignment problem given a similarity matrix, but the deployment experience shows that the quality of the similarity scores is extremely important. Future work could combine text-based similarity with structured subject-area information, publication histories, reviewer bids, keywords, and other metadata. This may be especially useful for large interdisciplinary conferences, where general text similarity may not be enough to capture the expertise needed for specific papers.

Lastly, future work could also improve how assignment systems handle possible collusion. RAMP includes penalties for coauthorship structure and bid-based 2-cycles, but these are only two possible signals of collusion risk. Other signals, such as unusual bidding patterns, repeated co-reviewing relationships, institutional connections, or citation relationships, may also be useful. In fact, there are already many work along this line. Jecmen, Shah, Fang, and Akoglu's work explores the problem of detecting collusion rings, where groups of reviewers and authors coor-

dinate bids to influence paper assignments. Their findings suggest that standard fraud detection algorithms, which typically focus on bidding patterns alone, are insufficient for detecting collusion and manipulation. This highlights the importance of splitting techniques that can partition reviewers and papers in a way that reduces the chance of collusion by ensuring that reviewers are not assigned papers they can influence through coordinated manipulation [6]. Furthermore, Hsieh, Raghunathan, and Shah demonstrate that even similarity-based assignment components, which are generally thought to be resistant to manipulation, are vulnerable to manipulation by coordinated colluders who adjust their profiles. This work emphasizes the need for more robust assignment algorithms that take both bids and profile-based manipulation into account [4]. On the mitigation front, Wu et al. propose algorithms that incorporate splitting techniques to dynamically partition papers and reviewers to reduce the influence of dishonest or collusive bids while maintaining assignment quality. These methods help reduce the effectiveness of coordinated bids by making the assignment process less predictable and by limiting collusion opportunities through reviewer splitting [17]. These efforts are an important part of the growing body of work that addresses the challenges posed by collusion and strategic behavior in peer review systems.

These signals could be used to adjust similarity scores, impose stricter marginal caps, or add new penalty terms to the optimization problem.

In summary, this thesis presents RAMP as a flexible framework for paper assignment and studies how it can address several practical requirements of large conferences. Future work can build on this foundation by improving similarity scores, strengthening sampling methods, reducing deployment burden, and evaluating the effect of assignment algorithms on the peer-review process as a whole.

Bibliography

[1] Michael Cui, Chenxin Dai, Yixuan Even Xu, and Fei Fang. A unified framework for scalable and robust paper assignment. *arXiv preprint arXiv:2601.14402*, 2026. doi: 10.48550/arXiv.2601.14402. 1.3, **??**

[2] 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. 1.2, 2.2, 2.4, 3.1, 3.2, 4.2

[3] Naveen Garg, Telikepalli Kavitha, Amit Kumar, Kurt Mehlhorn, and Julián Mestre. Assigning papers to referees. *Algorithmica*, 58(1):119–136, 2010. doi: 10.1007/s00453-009-9386-0. URL https://doi.org/10.1007/s00453-009-9386-0. 1.2, 2.2, 2.4, 3.1, 3.2, 4.2

[4] Jhih-Yi (Janet) Hsieh, Aditi Raghunathan, and Nihar B. Shah. Vulnerability of text-matching in ml/ai conference reviewer assignments to collusions. In *34th USENIX Security Symposium (USENIX Security 2025)*, 2025. URL https://arxiv.org/abs/2412.06606. 7.2

[5] Steven Jecmen, Hanrui Zhang, Ryan Liu, Nihar B. Shah, Vincent Conitzer, and Fei Fang. Mitigating manipulation in peer review via randomized reviewer assignments. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546. 1.2, 1.3, 2.2, 2.3, 2.4, **??**, 3.1, 3.2, 3.3, 3.4, 3.4, 3.6, 3.6, 3.6, 4.2, 4.3

[6] Steven Jecmen, Nihar B. Shah, Fei Fang, and Leman Akoglu. On the detection of reviewer-author collusion rings from paper bidding. *arXiv:2402.07860*, 2024. URL https://arxiv.org/abs/2402.07860. 7.2

[7] Tom Jefferson et al. Effects of editorial peer review: a systematic review. *JAMA*, 2007. 1.1

[8] Ari Kobren, Barna Saha, and Andrew McCallum. Paper matching with local fairness constraints. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 1247–1257, 2019. doi: 10.1145/3292500.3330899. URL https://doi.org/10.1145/3292500.3330899. 1.2, 2.3, 2.4

[9] Kevin Leyton-Brown, Mausam, Yatin Nandwani, Hedayat Zarkoob, Chris Cameron, Neil Newman, and Dinesh Raghu. Matching papers and reviewers at large conferences. *Artif. Intell.*, 331(C), June 2024. ISSN 0004-3702. doi: 10.1016/j.artint.2024.104119. URL https://doi.org/10.1016/j.artint.2024.104119. 1.2, 1.3, 2.1, 2.3, 2.4,

??, 3.1, 3.4, 3.4, 4.2, 4.3, 4.7

[10] Jing Wu Lian, Nicholas Mattei, Renee Noble, and Toby Walsh. The conference paper assignment problem: Using order weighted averages to assign indivisible goods. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1138–1145, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17396. 1.2, 2.2, 2.4, 3.1, 3.2, 4.2

[11] National Academy of Sciences. *On Being a Scientist: A Guide to Responsible Conduct in Research*. 2009. 1.1

[12] OpenReview. How to compute affinity scores, 2023. https://docs.openreview.net/how-to-guides/paper-matching-and-assignment/how-to-compute-affinity-scores. 2.2

[13] Ivan Stelmakh, Nihar B. Shah, and Aarti Singh. Peerreview4all: Fair and accurate reviewer assignment in peer review. In Aurélien Garivier and Satyen Kale, editors, *Proceedings of the 30th International Conference on Algorithmic Learning Theory*, volume 98 of *Proceedings of Machine Learning Research*, pages 828–856. PMLR, 22–24 Mar 2019. URL https://proceedings.mlr.press/v98/stelmakh19a.html. 1.2, 1.3, 2.3, 2.4

[14] Wenbin Tang, Jie Tang, and Chenhao Tan. Expertise matching via constraint-based optimization. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 34–41, 2010. doi: 10.1109/WI-IAT.2010.133. URL https://doi.org/10.1109/WI-IAT.2010.133. 1.2, 2.2, 2.4, 3.1, 3.2, 4.2

[15] Camillo J. Taylor. On the optimal assignment of conference papers to reviewers. Technical report, 2008. 1.2, 2.2, 2.3, 2.4, 3.1, 3.3, 4.2

[16] Ruihan Wu, Chuan Guo, Felix Wu, Rahul Kidambi, Laurens van der Maaten, and Kilian Q. Weinberger. Making paper reviewing robust to bid manipulation attacks. *ArXiv*, abs/2102.06020, 2021. URL https://api.semanticscholar.org/CorpusID:231879710. 1.3, 2.2, 2.3, 2.4, 3.1, 4.1, 4.3

[17] Ruihan Wu, Chuan Guo, Felix Wu, Rahul Kidambi, Laurens van der Maaten, and Kilian Q. Weinberger. Making paper reviewing robust to bid manipulation attacks. *arXiv:2102.06020*, 2021. URL https://arxiv.org/abs/2102.06020. 7.2

[18] Yixuan Even Xu, Steven Jecmen, Zimeng Song, and Fei Fang. A one-size-fits-all approach to improving randomness in paper assignment. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc. 1.2, 1.3, 2.2, 2.3, 2.4, **??**, 3.1, 3.2, 3.3, 3.4, 3.4, 3.5, 3.6, 4.1, 4.2, 4.3, 4.5

[19] Xiquan Zhao and Yangsen Zhang. Reviewer assignment algorithms for peer review automation: A survey. *Information Processing and Management*, 59(6):103028, 2022. ISSN 0306-4573. doi: 10.1016/j.ipm.2022.103028. URL https://doi.org/10.1016/j.ipm.2022.103028. Open access under CC BY 4.0 license. 1.2, 4.1