Mitigating Manipulation in Peer Review via Randomized Reviewer Assignments

Steven Jecmen, Hanrui Zhang, Ryan Liu,
Nihar B. Shah, Vincent Conitzer, and Fei Fang

Abstract

We consider three important challenges in conference peer review: (i) reviewers maliciously attempting to get assigned to certain papers to provide positive reviews, possibly as part of quid-pro-quo arrangements with the authors; (ii) "torpedo reviewing," where reviewers deliberately attempt to get assigned to certain papers that they dislike in order to reject them; (iii) reviewer de-anonymization on release of the similarities and the reviewer-assignment code. On the conceptual front, we identify connections between these three problems and present a framework that brings all these challenges under a common umbrella. We then present a (randomized) algorithm for reviewer assignment that can optimally solve the reviewer-assignment problem under any given constraints on the probability of assignment for any reviewer-paper pair. We further consider the problem of restricting the joint probability that certain suspect pairs of reviewers are assigned to certain papers, and show that this problem is NP-hard for arbitrary constraints on these joint probabilities but efficiently solvable for a practical special case. Finally, we experimentally evaluate our algorithms on datasets from past conferences, where we observe that they can limit the chance that any malicious reviewer gets assigned to their desired paper to 50% while producing assignments with over 90% of the total optimal similarity.

1   Introduction

Peer review, the evaluation of work by others working in the same field as the producer of the work or with similar competencies, is a critical component of scientific research. It is regarded favorably by a significant majority of researchers and is seen as being essential to both improving the quality of published research and validating the legitimacy of research publications [1–3]. Due to the wide adoption of peer review in the publication process in academia, the peer-review process can be very high-stakes for authors, and the integrity of the process can significantly influence the careers of the authors (especially due to the prominence of a "rich get richer" effect in academia [4]).

However, there are several challenges that arise in peer review relating to the integrity of the review process. In this work, we address three such challenges for peer review in academic conferences where a number of papers need to be assigned to reviewers at the same time.

(1) Untruthful favorable reviews. In order to achieve a good reviewer assignment, peer review systems must solicit some information about their reviewers' knowledge and interests (e.g., through bidding). This can be manipulated, and in fact this is known to have happened in at least one ACM conference [5]: "*Another SIG community has had a collusion problem where the investigators found that a group of PC members and authors colluded to bid and push for each other's papers violating the usual conflict-of-interest rules.*" The problem of manipulation is not limited to the bidding system, as practically anything used to determine paper assignments (e.g., self-reported area of expertise, list of papers the reviewer has published) can potentially be manipulated [6, 7]. In some cases, unethical authors may enter into deals with potential reviewers for their paper, where the reviewer agrees to attempt to get assigned to the author's paper and give it a favorable review in exchange for some

outside reward (e.g., as part of a quid-pro-quo arrangement for the reviewer's own paper in another publication venue). To preserve the integrity of the reviewing process and maintain community trust, the paper assignment algorithm should guarantee the mitigation of these kinds of arrangements.

(2) Torpedo reviewing. In "torpedo reviewing," unethical reviewers attempt to get assigned to papers they dislike with the intent of giving them an overly negative review and blocking the paper from publication. This can have wide-reaching consequences [8]: "*Repeated indefinitely, this gives the power to kill off new lines of research to the 2 or 3 most close-minded members of a community, potentially substantially retarding progress for the community as a whole.*" One special case of torpedo reviewing has been called "rational cheating," referring to reviewers negatively reviewing papers that compete with their own authored work [9,10]. A paper assignment algorithm should guarantee to authors that their papers are unlikely to have been torpedo-reviewed.

(3) Reviewer de-anonymization in releasing assignment data. For transparency and research purposes, conferences may wish to release the paper-reviewer similarities and the paper assignment algorithm used after the conference. However, if the assignment algorithm is deterministic, this would allow for authors to fully determine who reviewed their paper, breaking the anonymity of the reviewing process. Even when reviewer and paper names are removed, identities can still be discovered (as in the case of the Netflix Prize dataset [11]). Consequently, a rigorous guarantee of anonymity is needed in order to release the data.

Although these challenges may seem disparate, we address all of them under a common umbrella framework. Our contributions are as follows. **Conceptually**, we formulate problems concerning the three aforementioned issues, and propose a framework to address them through the use of randomized paper assignments (Section 3). **Theoretically**, we design computationally efficient, randomized assignment algorithms that optimally assign reviewers to papers subject to given restrictions on the probability of assigning any particular reviewer-paper pair (Section 4). We further consider the more complex case of preventing suspicious *pairs* of reviewers from being assigned to the same paper (Section 5). We show that finding the optimal assignment subject to arbitrary constraints on the probabilities of reviewer-reviewer-paper assignments is NP-hard. In the practical special case where the program chairs want to prevent pairs of reviewers within the same subset of some partition of the reviewer set (e.g., by academic institution) from being assigned to the same paper, we present an algorithm that finds the optimal randomized assignment with this guarantee. **Empirically**, we test our algorithms on datasets from past conferences and show their practical effectiveness (Section 6). As a representative example, on data reconstructed from ICLR 2018, our algorithms can limit the chance of any reviewer-paper assignment to 50% while achieving 90.8% of the optimal total similarity. Our algorithms can continue to achieve this similarity while also preventing reviewers with close associations from being assigned to the same paper. We further demonstrate, using the ICLR 2018 dataset, that our algorithm successfully prevents manipulation of the assignment by a simulated malicious reviewer.

All of the code for our algorithms and our empirical results is freely available online.[1]

2   Related Literature

Many paper assignment algorithms for conference peer review have been proposed in past work. The widely-used Toronto Paper Matching System (TPMS) [12] computes a similarity score for each reviewer-paper pair based on analysis of the reviewers' past work and bids, and then aims to maximize the total similarity of the resulting assignment. The framework of "compute similarities and maximize total similarity" (and similar variants) encompasses

---

many paper assignment algorithms, where similarities can be computed in various ways from automated and manual analysis and reviewer bids [13–19]. We treat the bidding process and computation of similarities as given, and focus primarily on adjusting the optimization problem to address the three aforementioned challenges. There are also a number of recent works [20–34] which deal with other aspects of peer review.

Much prior work has studied the issue of preventing or mitigating strategic behavior in peer review. This work usually focuses on the incentives reviewers have to give poor reviews to other papers in the hopes of increasing their own paper's chances of acceptance [35–39]. Unlike the issues we deal with in this paper, these works consider only reviewers' incentives to get *their own paper accepted* and not other possible incentives. We instead consider arbitrary incentives for a reviewer to give an untruthful review, such as a personal dislike for a research area or misincentives brought about by author-reviewer collusion. Instead of aiming to remove reviewer incentives to be untruthful, our work focuses on mitigating the effectiveness of manipulating the reviewer assignment process.

Randomized assignments have been used to address the problem of fair division of indivisible goods such as jobs or courses [40, 41], as well as in the context of Stackelberg security games [42]. The paper [22] uses randomization to address the issue of miscalibration in ratings, such as those given to papers in peer review. The paper [43] considers fractional paper assignments in order to reason about reviewer incentives when bidding in peer review, where the fractional assignment can be interpreted as reviewers' perceived assignment probabilities under an abstracted assignment algorithm. Unlike this work, they do not tackle the problem of malicious reviewers nor actually randomize the assignment. To the best of our knowledge, the use of randomized reviewer-paper assignments to address the issues of malicious reviewers or reviewer de-anonymization in peer review has not been studied previously.

Additional coverage of related literature is presented in Appendix A.

3 Background and Problem Statements

We first define the standard paper assignment problem, followed by our problem setting. In the standard paper assignment setting, we are given a set $\mathcal{R}$ of n reviewers and a set $\mathcal{P}$ of d papers, along with desired reviewer load k (that is, the maximum number of papers any reviewer should be assigned) and desired paper load ℓ (that is, the exact number of reviewers any paper should be assigned to).[2] An assignment of papers to reviewers is a bipartite matching between the sets that obeys the load constraints on all reviewers and papers. In addition, we are given a similarity matrix $S \in \mathbb{R}^{n \times d}$ where S_{rp} denotes how good of a match reviewer r is for paper p. These similarities can be derived from the reviewers' bids on papers, prior publications, conflicts of interest, etc.

The standard problem of finding a maximum sum-similarity assignment [12, 13, 15, 17, 19] is then written as an integer linear program. The decision variables $M \in \{0,1\}^{n \times d}$ specify the assignment, where $M_{rp} = 1$ if and only if reviewer r is assigned to paper p. The objective is to maximize $\sum_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}} S_{rp} M_{rp}$ subject to the load constraints $\sum_{p \in \mathcal{P}} M_{rp} \leq k, \forall r \in \mathcal{R}$ and $\sum_{r \in \mathcal{R}} M_{rp} = \ell, \forall p \in \mathcal{P}$. Since the constraint matrix of the linear program (LP) relaxation of this problem is totally unimodular, the solution to the LP relaxation will be integral and so this problem can be solved as an LP. This method of assigning papers has been used by various conferences such as NeurIPS, ICML, ICCV, and SIGKDD (among others) [12, 17], as well as by popular conference management systems EasyChair (`easychair.org`) and

---

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

HotCRP (`hotcrp.com`).

When the assignment algorithm is deterministic in this way, a malicious reviewer who knows the algorithm may be able to effectively manipulate it in order to get assigned to a desired paper. To address this issue, we aim to provide a guarantee that regardless of the reviewer bids and similarities, this reviewer-paper pair has only a limited probability of being assigned. Additionally, consider the challenge of preserving anonymity in releasing conference data. If the assignment algorithm provides a guarantee that each reviewer-paper pair has only a limited probability of being assigned, then no reviewer's identity can be discovered with certainty. With this motivation, we now consider M as stochastic and aim to find a *randomized assignment*, a probability distribution over deterministic assignments. This naturally leads to the following problem formulation.

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

Since a randomized assignment is a distribution over deterministic assignments, all assignments M in the support of the randomized assignment must still obey the load constraints $\sum_{p \in \mathcal{P}} M_{rp} \leq k, \forall r \in \mathcal{R}$ and $\sum_{r \in \mathcal{R}} M_{rp} = \ell, \forall p \in \mathcal{P}$. The optimization objective is the expected sum-similarity across all paper-reviewer pairs, the natural analogue of the deterministic sum-similarity objective.

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

A second issue is that there may be dependencies within certain pairs of reviewers that cannot be accurately represented by constraints on individual reviewer-paper pairs. For example, we may have two reviewers a and b who are close collaborators, each of which we are not individually very concerned about assigning to paper p. However, we may believe that in the case where reviewer a has entered into a quid-pro-quo deal to dishonestly review paper p, reviewer b is likely to also be involved in the same deal. Therefore, one may want to strictly limit the probability that **both** reviewers a and b are assigned to paper p, regardless of the limits on the probability that either reviewer individually is assigned to paper p.

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

Definition 2 (Triplet-Constrained Problem). *The input to the problem is a similarity matrix S, a matrix $Q \in [0,1]^{n \times d}$, and a 3-dimensional tensor $T \in [0,1]^{n \times n \times d}$. The goal is to find a randomized assignment of papers to reviewers that maximizes $\mathbb{E}\left[\sum_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}} S_{rp} M_{rp}\right]$ subject to the constraints $\mathbb{P}[M_{rp} = 1] \leq Q_{rp}, \forall r \in \mathcal{R}, p \in \mathcal{P}$ and $\mathbb{P}[M_{ap} = 1 \wedge M_{bp} = 1] \leq T_{abp}, \forall a, b \in \mathcal{R} \text{ s.t. } a \neq b, p \in \mathcal{P}$.*

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

- **Untruthful favorable reviews:** By guaranteeing a limit on the probability that any malicious reviewer or pairs of reviewers can be assigned to the paper they want, we mitigate the effectiveness of any unethical deals between reviewers and authors by capping the probability that such a deal can be upheld. This guarantee holds regardless of how extreme a reviewers' manipulation of the assignment is and without any assumptions on reviewers' exact incentives. The entries of Q or T can be set by the program chairs based on their assessment of the risk of allowing the corresponding assignment.

- **Torpedo reviewing:** By limiting the probability that any reviewer or pair of reviewers can be assigned to a paper they wish to torpedo, we make it much more difficult for a small group of reviewers to shut down a new research direction or to take out competing papers.

- **Reviewer de-anonymization in releasing assignment data:** If all of the entries in Q are set to some reasonable constant value, only the distribution over assignments can be recovered and not the specific assignment that was actually used. This guarantees that for each paper, no reviewer's identity can be identified with high confidence, since every reviewer has only a limited chance to be assigned to that paper.

In Sections 4 and 5, we consider the Pairwise-Constrained Problem and Triplet-Constrained Problem respectively. We also consider several related problems in Appendices B and C.

4 Randomized Assignment with Reviewer-Paper Constraints

In this section we present our main algorithm to solve the Pairwise-Constrained Problem (Definition 1), thereby addressing the challenges identified earlier. Before delving into the details of the algorithm, the following theorem states our main result.

Theorem 1. *There exists an algorithm which returns an optimal solution to the Pairwise-Constrained Problem in poly(n, d) time.*

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

4.1 Finding the Fractional Assignment

Define a *fractional assignment matrix* as a matrix $F \in [0, 1]^{n \times d}$ that obeys the load constraints $\sum_{p \in \mathcal{P}} F_{rp} \leq k$ for all reviewers $r \in \mathcal{R}$ and $\sum_{r \in \mathcal{R}} F_{rp} = \ell$ for all papers $p \in \mathcal{P}$. Note that any deterministic assignment can be represented by a fractional assignment matrix with all entries in $\{0, 1\}$. Any randomized assignment is associated with a fractional assignment matrix where F_{rp} is the marginal probability that reviewer r is assigned to paper p. Furthermore, randomized assignments associated with the same fractional assignment matrix have the same expected sum-similarity. The paper [44] proves an extension of the Birkhoff-von Neumann theorem [45, 46] which shows that all fractional assignment matrices are implementable, i.e., they are associated with at least one randomized assignment. On the other hand, any probability matrix not obeying the load constraints cannot be implemented by a lottery over deterministic assignments, since all deterministic assignments do obey the constraints.

Therefore, finding the optimal randomized assignment is equivalent to solving the following LP, which we call $\mathcal{LP}1$:

$$\underset{F \in \mathbb{R}^{n \times d}}{\arg\max} \quad \sum_{p \in \mathcal{P}} \sum_{r \in \mathcal{R}} S_{rp} F_{rp} \tag{1}$$

$$\text{subject to} \quad 0 \le F_{rp} \le 1 \qquad \forall r \in \mathcal{R}, \forall p \in \mathcal{P} \tag{2}$$

$$\sum_{p \in \mathcal{P}} F_{rp} \le k \qquad \forall r \in \mathcal{R} \tag{3}$$

$$\sum_{r \in \mathcal{R}} F_{rp} = \ell \qquad \forall p \in \mathcal{P} \tag{4}$$

$$F_{rp} \le Q_{rp} \qquad \forall r \in \mathcal{R}, \forall p \in \mathcal{P}. \tag{5}$$

$\mathcal{LP}1$ has $O(dn)$ bounded variables and $O(d+n)$ other constraints. By [47], $\mathcal{LP}1$ can be solved with high probability in $\tilde{O}(dn(d+n) + (d+n)^{2.5})$ time.

4.2 Implementing the Probabilities

$\mathcal{LP}1$ only finds the optimal marginal assignment probabilities F (where F now refers to a solution to $\mathcal{LP}1$). It remains to show whether and how these marginal probabilities can be implemented as a randomization over deterministic paper assignments. The paper [44] provides a method for sampling a deterministic assignment from a fractional assignment matrix, which completes our algorithm once applied to the optimal solution of $\mathcal{LP}1$. Here we briefly sketch a simpler version of the sampling algorithm. In Appendix D, we describe the sampling algorithm in detail along with a supplementary algorithm to compute the full distribution over assignments.

The algorithm begins by constructing a complete bipartite graph with vertices representing reviewers and papers, and connecting source and sink vertices to the reviewer and paper vertices respectively. It also constructs a flow function over the edges, where the initial flow on each reviewer-paper edge is set equal to the marginal assignment probability for the adjacent reviewer and paper. The algorithm then proceeds in an iterative manner, modifying the flow function on each iteration. On each iteration, we first check if there exists a "fractional edge," an edge with non-integral flow. If no such edge exists, we can stop iterating and return the assignment represented by the current integral flows. If there does exist a fractional edge, we then find an arbitrary cycle of fractional edges, ignoring direction. On finding a cycle, we modify the flow function by "pushing" flow along the cycle in either the forwards or backwards direction until the flow on some edge becomes integral. We randomly choose the direction with probabilities proportional to the amount of flow pushed, so that the final sampled assignment satisfies the desired marginal probabilities.

Each iteration of this algorithm takes $O(d+n)$ time to find a cycle in the $O(d+n)$ vertices (if a list of fractional edges adjacent to each vertex is maintained), and it can take $O(dn)$ iterations to terminate since one edge becomes integral every iteration. Therefore, the sampling algorithm is overall $O(dn(d+n))$. The time complexity of our full algorithm, including both $\mathcal{LP}1$ and the sampling algorithm, is dominated by the complexity of solving the LP. Since standard paper assignment algorithms such as TPMS can be implemented by solving an LP of the same size, our algorithm is comparable in complexity. If a conference currently does solve an LP to find their assignment, whatever LP solver a conference currently uses for their paper assignment algorithm could be used in our algorithm as well.

5 Randomized Assignment with Constraints on Pairs of Reviewers

We now turn to the problem of controlling the probabilities that certain pairs of reviewers are assigned to the same paper, defined in Section 3 as the Triplet-Constrained Problem (Definition 2). As described in Section 3, solving the Triplet-Constrained Problem would allow the program chairs of a conference maximum flexibility in how they control the probabilities of the assignments of pairs of reviewers. Unfortunately, as the following theorem shows, this problem cannot be efficiently solved. The proof is stated in Appendix E.

Theorem 2. *The Triplet-Constrained Problem is NP-hard, by reduction from 3-Dimensional Matching.*

Since the most general problem of arbitrary constraints on reviewer-reviewer-paper triples is NP-hard, we must restrict ourselves to tractable special cases of interest. One such special case arises when the program chairs of a conference can partition the reviewers in such a way that they wish to prevent any two reviewers within the same subset from being assigned to the same paper. For example, reviewers can be partitioned by their primary academic institution. Since reviewers at the same institution are likely closely associated, program chairs may believe that placing them together as co-reviewers is more risky than would be implied by our concern about either reviewer individually. In this case, there may not even be any concern about the reviewers' motivations; the concern may simply be that the reviewers' opinions would not be sufficiently independent. Other partitions of interest could be the reviewer's geographical area of residence or research sub-field, as each of these defines a "community" of reviewers that may be more closely associated. This special case corresponds to instances of the Triplet-Constrained Problem where $T_{abp} = 0$ if reviewers a and b are in the same subset, and $T_{abp} = 1$ otherwise. We formally define this problem as follows:

Definition 3 (Partition-Constrained Problem)**.** *The input to the problem is a similarity matrix S, a matrix $Q \in [0,1]^{n \times d}$, and a partition of the reviewer set into subsets $I_1, \ldots, I_m \subseteq \mathcal{R}$. The goal is to find a randomized assignment of papers to reviewers that maximizes $\mathbb{E}\left[\sum_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}} S_{rp} M_{rp}\right]$ subject to the constraints that $\mathbb{P}[M_{rp} = 1] \leq Q_{rp}, \forall r \in \mathcal{R}, p \in \mathcal{P}$, and $\mathbb{P}[M_{ap} = 1 \wedge M_{bp} = 1] = 0, \forall a, b \in I_i, \forall i \in [m]$.*

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

Theorem 3. *There exists an algorithm which returns an optimal solution to the Partition-Constrained Problem in poly(n, d) time.*

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

5.1 Partition-Constrained Sampling Algorithm

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

Lemma 1. *Consider any fractional assignment matrix F and any partition of $\mathcal{R}$ into subsets $I_1, \ldots, I_m$.*

> *(i) There exists a sampling algorithm that implements the marginal assignment probabilities given by F and runs in $O(dn(d+n))$ time such that, for all papers $p \in \mathcal{P}$ and subsets $I \in \{I_1, \ldots, I_m\}$ where $\sum_{r \in I} F_{rp} \leq 1$, the algorithm never samples an assignment assigning two reviewers from subset I to paper p.*

> *(ii) For any sampling algorithm that implements the marginal assignment probabilities given by F, for all papers $p \in \mathcal{P}$ and subsets $I \in \{I_1, \ldots, I_m\}$ where $\sum_{r \in I} F_{rp} > 1$, the expected number of pairs of reviewers from subset I assigned to paper p is strictly positive.*

The algorithm which realizes Lemma 1 has an additional property, which holds *simultaneously* for all papers and subsets. We state the property in the following corollary and make use of it later:

Corollary 1. *For any fractional assignment matrix F, the sampling algorithm that realizes Lemma 1 minimizes the expected number of pairs of reviewers from subset I assigned to paper p simultaneously for all papers $p \in \mathcal{P}$ and subsets $I \in \{I_1, \ldots, I_m\}$ among all sampling algorithms implementing the marginal assignment probabilities given by F.*

We briefly sketch the sampling algorithm that realizes these results here, and describe it in detail along with proofs of the guarantees stated in Lemma 1 and Corollary 1 in Appendix F. This algorithm is a modification of the sampling algorithm from Theorem 1 sketched earlier in Section 4.2. We first provide some high-level intuition about the modifications. For any fractional assignment matrix F, for any subset I and paper p, the expected number of reviewers from subset I assigned to paper p is $\sum_{r \in I} F_{rp}$. This is equal to the initial load from subset I on paper p in the algorithm (that is, the sum of the flow on all edges from reviewers in subset I to paper p). Note that at the algorithm's conclusion, when all edges are integral, the load from subset I on paper p is equal to the number of reviewers from subset I assigned to paper p. Therefore, if the fractional assignment F is such that the initial expected number of reviewers from subset I assigned to paper p is no greater than 1 (as stated in part (i) of Lemma 1), we want to keep the load from subset I on paper p close to its initial value so that the final number of reviewers from subset I assigned to paper p is also no greater than 1. With this reasoning, we modify the algorithm from Theorem 1 so that in each iteration, it ensures that the total load on each paper from each subset is unchanged if originally integral and is never moved past the closest integer in either direction if originally fractional. The time complexity of this modified algorithm is identical to that of the original algorithm from Theorem 1.

5.2 Finding the Optimal Partition-Constrained Fractional Assignment

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

$$\sum_{r \in I} F_{rp} \leq 1 \quad \forall I \in \{I_1, \ldots, I_m\}, \forall p \in \mathcal{P}. \tag{6}$$

The solution to $\mathcal{LP}2$ when paired with the sampling algorithm from Section 5.1 never assigns two reviewers from the same subset to the same paper. Furthermore, since any

fractional assignment F not obeying Constraint (6) will have a strictly positive probability of assigning two reviewers from the same subset to the same paper, $\mathcal{LP}2$ finds the optimal fractional assignment with this guarantee. This completes the algorithm for the Partition-Constrained Problem.

Additionally, Corollary 1 shows that the sampling algorithm from Section 5.1 is optimal in the expected number of same-subset reviewer pairs, for any fractional assignment. If the guarantee of entirely preventing same-subset reviewer pairs is not strictly required, Constraint (6) in $\mathcal{LP}2$ can be loosened (constraining the subset loads to a higher value) without removing it entirely. For the resulting fractional assignment F, the sampling algorithm from Section 5.1 still minimizes the expected number of pairs of reviewers from any subset on any paper, as compared to any other sampling algorithm implementing F. Since the subset loads are still constrained, the expected number of same-subset reviewer pairs will be lower than in the solution to the Pairwise-Constrained Problem (at the cost of some similarity). We examine this tradeoff experimentally in Section 6.

6 Experiments

We test our algorithms on several real-world datasets. The first real-world dataset is a similarity matrix recreated from ICLR 2018 data in [35]; this dataset has $n = 2435$ reviewers and $d = 911$ papers. We also run experiments on similarity matrices created from reviewer bid data for three AI conferences from PrefLib dataset MD-00002 [48], with sizes $(n = 31, d = 54)$, $(n = 24, d = 52)$, and $(n = 146, d = 176)$ respectively. For all three PrefLib datasets, we transformed "yes," "maybe," and "no response" bids into similarities of 4, 2, and 1 respectively, as is often done in practice [27]. As done in [35], we set loads $k = 6$ and $\ell = 3$ for all datasets since these are common loads for computer science conferences (except on the PrefLib2 dataset, for which we set $k = 7$ for feasibility).

We run all experiments on a computer with 8 cores and 16 GB of RAM, running Ubuntu 18.04 and using Gurobi 9.0.2 [49] to solve the LPs. Our algorithm for the Pairwise-Constrained Problem takes an average of 41 seconds to complete on ICLR; our algorithm for the Partition-Constrained Problem takes an average of 45 seconds. As expected, the running time is dominated by the LP solving.

6.1 Quality of Resulting Assignments

We first study our algorithm for the Pairwise-Constrained Problem, as described in Section 4. In this setting, program chairs must make a tradeoff between the quality of the output assignments and guarding against malicious reviewers or reviewer de-anonymization by setting the values of the maximum-probability matrix Q. We investigate this tradeoff on real datasets. All results in this section are averaged over 10 trials with error bars plotted representing the standard error of the mean, although they are sometimes not visible since the variance is very low.

In Figure 1a, we set all entries of the maximum-probability-matrix Q equal to the same constant value q_0 (varied on the x-axis), and observe how the sum-similarity value of the assignment computed via our algorithm from Section 4 changes as q_0 increases from 0.1 to 1 with an interval of 0.1. We report the sum-similarity as a percentage of the unconstrained optimal solution's objective (which does not address the aforementioned challenges). We see that our algorithm trades off the maximum probability of an assignment gracefully against the sum-similarity on all datasets. For instance, with $q_0 = 0.5$, our algorithm achieves 90.8% of the optimal objective value on the ICLR dataset. In practice, this would allow program chairs to limit the chance that any malicious reviewer is assigned to their desired paper to 50% without suffering a significant loss of assignment quality.

(a) Pairwise-Constrained Problem

(b) Partition-Constrained Problem with three random subsets
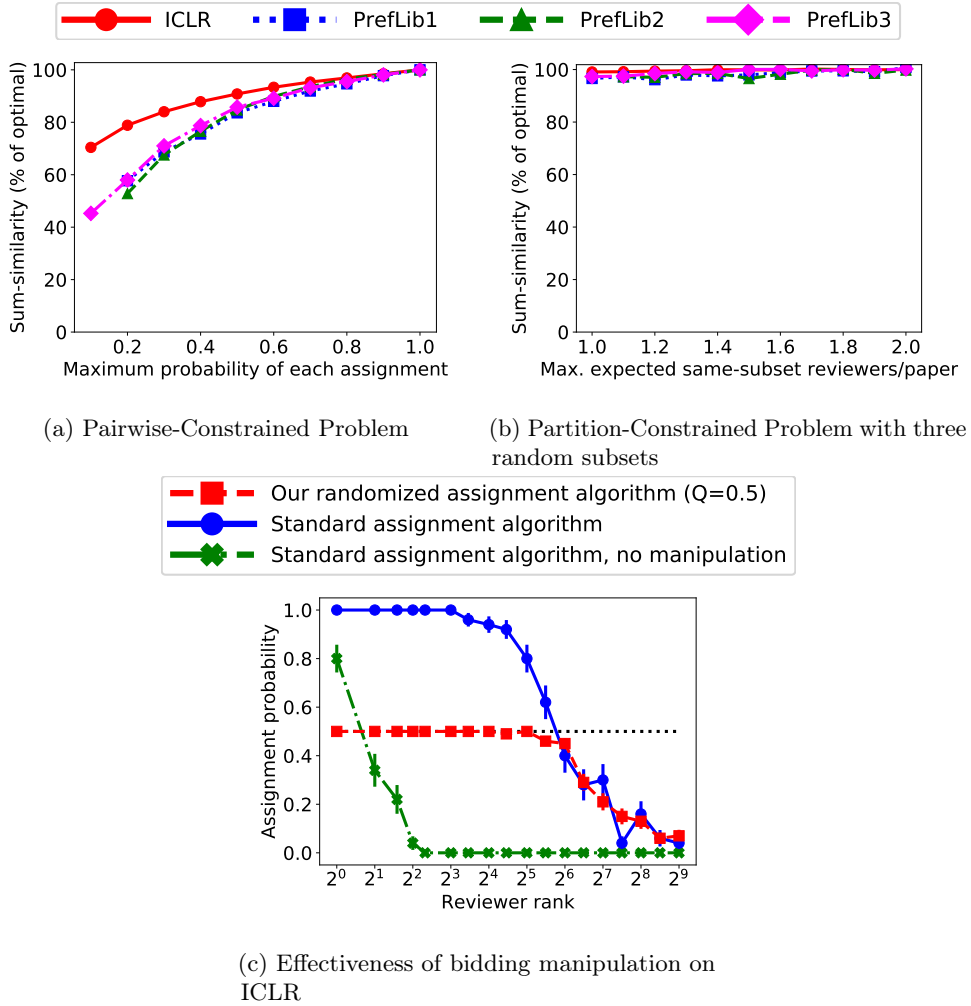

(c) Effectiveness of bidding manipulation on ICLR

Figure 1: Experimental results on four conference datasets.

We next test our algorithm for the Partition-Constrained Problem discussed in Section 5. In this algorithm, program chairs can navigate an additional tradeoff between the number of same-subset reviewers assigned to the same paper and the assignment quality; we investigate this tradeoff here. On ICLR, we fix $q_0 = 0.5$ and randomly assign reviewers to subsets of size 15 (representing academic institutions, for example), using this as our partition of $\mathcal{R}$ (since the dataset does not include any reviewer information). Our algorithm is able to achieve 100% of the optimal objective for the Pairwise-Constrained Problem with $q_0 = 0.5$ while preventing any pairs of reviewers from the same subset from being assigned to the same paper.

Since our algorithm achieves the full possible objective in this setting, we now run experiments with a considerably more restrictive partition constraint. In Figure 1b, we show an extreme case where we randomly assign reviewers to 3 subsets of equal size, again fixing $q_0 = 0.5$. We then gradually loosen the constraints on the expected number of same-subset reviewers assigned to the same paper by increasing the constant in Constraint (6) from 1 to 2 in increments of 0.1, shown on the x-axis. We plot the sum-similarity objective of the resulting assignment, expressed as a percentage of the optimal non-partition-constrained solution's

objective (i.e., the solution to the Pairwise-Constrained Problem with $q_0 = 0.5$). Even in this extremely constrained case, we still achieve 99.1% of the non-partition-constrained objective while entirely preventing same-subset reviewer pairs on ICLR.

In Appendix G, we present results for additional experiments on synthetic similarities. We also run experiments for a fairness objective, which we present in Appendix B.

6.2 Effectiveness at Preventing Manipulation

We now describe experiments evaluating the effectiveness of our algorithm at preventing manipulation on the ICLR dataset against a simulated reviewer bidding model. We assume that there is one malicious reviewer, who is attempting to maximize their chances of being assigned to a target paper solely through bidding (and not through other means). Since the ICLR similarities are reconstructed purely from the text similarity with each reviewers' past work and do not contain any bidding, we supplement them with synthetic bids. The malicious reviewer bids positively on their target paper and negatively on all other papers, while the other (honest) reviewers bid according to a simple randomized model constructed to match characteristics of the bidding observed in NeurIPS 2016 [27]. The function used to compute final similarities is based on the similarity function used in NeurIPS 2016 [27]. Details of the bidding are given in Appendix H, along with additional results.

In Figure 1c, we choose a target paper uniformly at random, and choose the malicious reviewer to be the reviewer with the xth highest pre-bid similarity with that paper (varying x on the x-axis, log-scaled). We then have all reviewers bid and compute the assignment with either the standard deterministic assignment algorithm described in Section 3 or our randomized assignment algorithm for the Pairwise-Constrained Problem, setting all entries of Q to 0.5. We then observe the probability that the malicious reviewer is assigned to the target paper. We report on the y-axis the average over 50 choices of target paper, giving an overall success rate for the manipulation under a uniform choice of papers (with error bars representing the standard error of the mean). For comparison, we also plot the case where only the honest reviewers bid and the malicious reviewer does not bid.

We see that under the standard assignment algorithm, the manipulation always succeeds for reviewers with high pre-bid similarity with the paper. In contrast, when the malicious reviewer does not bid, their assignment probability is much lower, especially for reviewers ranked in the top 4. This indicates that manipulation from reviewers in the same subject area as the paper is very powerful in standard assignment algorithms, potentially compromising the integrity of the assignment. However, our algorithm always limits the probability of successful manipulation to the desired level of 0.5.

7 Discussion

We have presented here a framework and a set of algorithms for addressing three challenges of practical importance to the peer review process: untruthful favorable reviews, torpedo reviewing, and reviewer de-anonymization on the release of assignment data. By design, our algorithms are quite flexible to the needs of the program chairs, depending on which challenges they are most concerned with addressing. Our empirical evaluations demonstrate some of the tradeoffs that can be made between total similarity and maximum probability of each paper-reviewer pair or number of reviewers from the same subset on the same paper. The exact parameters of the algorithm can be set based on how the program chairs weigh the relative importance of each of these factors.

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

additional special cases under which optimizing over constraints on the probabilities of reviewer-pair-to-paper assignments is feasible. Additionally, this work does not address the problem of reviewers colluding with each other to give dishonest favorable reviews after being assigned to each others' papers; we leave this issue for future work.

Acknowledgments

References

[1] Adrian Mulligan, Louise Hall, and Ellen Raphael. Peer review in a changing world: An international study measuring the attitudes of researchers. *Journal of the Association for Information Science and Technology*, 64(1):132–161, 2013.

[2] David Nicholas, Anthony Watkinson, Hamid R. Jamali, Eti Herman, Carol Tenopir, Rachel Volentine, Suzie Allard, and Kenneth Levine. Peer review: Still king in the digital age. *Learned Publishing*, 28(1):15–21, 2015.

[3] Mark Ware. Peer review: Benefits, perceptions and alternatives. *Publishing Research Consortium*, 4:1–20, 2008.

[4] Robert K. Merton. The Matthew effect in science. *Science*, 159:56–63, 1968.

[5] T. N. Vijaykumar. Potential organized fraud in ACM/IEEE computer architecture conferences. `https://medium.com/@tnvijayk/potential-organized-fraud-in-acm-ieee-computer-architecture-conferences-ccd61169370d`, 2020 (accessed August 17, 2020).

[6] Cat Ferguson, Adam Marcus, and Ivan Oransky. Publishing: The peer-review scam. *Nature News*, 515(7528):480, 2014.

[7] Jian Gao and Tao Zhou. Retractions: Stamp out fake peer review. *Nature*, 546(7656):33–33, 2017.

[8] John Langford. Bidding problems. `https://hunch.net/?p=407`, 2008 (accessed June 16, 2020).

[9] Edward F. Barroga. Safeguarding the integrity of science communication by restraining 'rational cheating' in peer review. *Journal of Korean Medical Science*, 29(11):1450–1452, 2014.

[10] Mario Paolucci and Francisco Grimaldo. Mechanism change in a simulation of peer review: From junk support to elitism. *Scientometrics*, 99(3):663–688, 2014.

[11] Arvind Narayanan and Vitaly Shmatikov. How to break anonymity of the Netflix Prize dataset. *arXiv preprint arXiv:cs/0610105*, 2006.

[12] Laurent Charlin and Richard Zemel. The Toronto Paper Matching System: An automated paper-reviewer assignment system. 2013.

[13] Laurent Charlin, Richard S. Zemel, and Craig Boutilier. A framework for optimizing paper matching. *arXiv preprint arXiv:1202.3706*, 2012.

[14] Cheng Long, Raymond Chi-Wing Wong, Yu Peng, and Liangliang Ye. On good and fair paper-reviewer assignment. In *2013 IEEE 13th International Conference on Data Mining*, pages 1145–1150. IEEE, 2013.

[15] Judy Goldsmith and Robert H. Sloan. The AI conference paper assignment problem. In *AAAI Workshop on Preference Handling for Artificial Intelligence*, pages 53–57, 2007.

[16] Wenbin Tang, Jie Tang, and Chenhao Tan. Expertise matching via constraint-based optimization. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 34–41. IEEE, 2010.

[17] 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.

[18] Jing Wu Lian, Nicholas Mattei, Renee Noble, and Toby Walsh. The conference paper assignment problem: Using order weighted averages to assign indivisible goods. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[19] Ari Kobren, Barna Saha, and Andrew McCallum. Paper matching with local fairness constraints. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019.

[20] Hong Ge, Max Welling, and Zoubin Ghahramani. A Bayesian model for calibrating conference review scores. `http://mlg.eng.cam.ac.uk/hong/unpublished/nips-review-model.pdf`, 2013 (accessed June 29, 2020).

[21] Ritesh Noothigattu, Nihar B. Shah, and Ariel D. Procaccia. Loss functions, axioms, and peer review. *arXiv preprint arXiv:1808.09057*, 2018.

[22] Jingyan Wang and Nihar B. Shah. Your 2 is my 1, your 3 is my 9: Handling arbitrary miscalibrations in ratings. In *18th International Conference on Autonomous Agents and MultiAgent Systems*, 2019.

[23] Tanner Fiez, Nihar B. Shah, and Lillian Ratliff. A SUPER* algorithm to optimize paper bidding in peer review. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2020.

[24] Magnus Roos, Jörg Rothe, and Björn Scheuermann. How to calibrate the scores of biased reviewers by quadratic programming. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[25] Ivan Stelmakh, Nihar B. Shah, and Aarti Singh. On testing for biases in peer review. In *Advances in Neural Information Processing Systems*, 2019.

[26] Neil Lawrence and Corinna Cortes. The NIPS experiment. `http://inverseprobability.com/2014/12/16/the-nips-experiment`, 2014 (accessed June 29, 2020).

[27] Nihar B. Shah, Behzad Tabibian, Krikamol Muandet, Isabelle Guyon, and Ulrike Von Luxburg. Design and analysis of the NIPS 2016 review process. *The Journal of Machine Learning Research*, 19(1):1913–1946, 2018.

[28] Andrew Tomkins, Min Zhang, and William D. Heavlin. Reviewer bias in single- versus double-blind peer review. *Proceedings of the National Academy of Sciences*, 114(48):12708–12713, 2017.

[29] Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard H. Hovy, and Roy Schwartz. A dataset of peer reviews (PeerRead): Collection, insights and NLP applications. *arXiv preprint arXiv:1804.09635*, 2018.

[30] Emaad Manzoor and Nihar B. Shah. Uncovering latent biases in text: Method and application to peer review. In *INFORMS Workshop on Data Science*, 2020.

[31] Ivan Stelmakh, Nihar B. Shah, Aarti Singh, and Hal Daumé III. Prior and prejudice: The bias against resubmissions in conference peer review. *arXiv preprint arXiv:2011.14646*, 2020.

[32] Ivan Stelmakh, Nihar B. Shah, and Aarti Singh. Catch me if I can: Detecting strategic behaviour in peer assessment. *arXiv preprint arXiv:2010.04041*, 2020.

[33] Naveen Garg, Telikepalli Kavitha, Amit Kumar, Kurt Mehlhorn, and Julián Mestre. Assigning papers to referees. *Algorithmica*, 58(1):119–136, 2010.

[34] Ivan Stelmakh, Nihar B. Shah, and Aarti Singh. PeerReview4All: Fair and accurate reviewer assignment in peer review. *arXiv preprint arXiv:1806.06237*, 2018.

[35] Yichong Xu, Han Zhao, Xiaofei Shi, Jeremy Zhang, and Nihar B. Shah. On strategyproof conference peer review. *arXiv preprint arXiv:1806.06266*, 2018.

[36] Haris Aziz, Omer Lev, Nicholas Mattei, Jeffrey S. Rosenschein, and Toby Walsh. Strategyproof peer selection using randomization, partitioning, and apportionment. *Artificial Intelligence*, 275:295–309, 2019.

[37] David Kurokawa, Omer Lev, Jamie Morgenstern, and Ariel D. Procaccia. Impartial peer review. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[38] Anson Kahng, Yasmine Kotturi, Chinmay Kulkarni, David Kurokawa, and Ariel D. Procaccia. Ranking wily people who rank each other. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[39] Ron Holzman and Hervé Moulin. Impartial nominations for a prize. *Econometrica*, 81(1):173–196, 2013.

[40] Aanund Hylland and Richard Zeckhauser. The efficient allocation of individuals to positions. *Journal of Political Economy*, 87(2):293–314, 1979.

[41] Anna Bogomolnaia and Hervé Moulin. A new solution to the random assignment problem. *Journal of Economic Theory*, 100(2):295–328, 2001.

[42] Dmytro Korzhyk, Vincent Conitzer, and Ronald Parr. Complexity of computing optimal Stackelberg strategies in security resource allocation games. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[43] Reshef Meir, Jérôme Lang, Julien Lesca, Natan Kaminsky, and Nicholas Mattei. A market-inspired bidding scheme for peer review paper assignment. In *Games, Agents, and Incentives Workshop at AAMAS*, 2020.

[44] Eric Budish, Yeon-Koo Che, Fuhito Kojima, and Paul Milgrom. Implementing random assignments: A generalization of the Birkhoff-von Neumann theorem. In *2009 Cowles Summer Conference*, 2009.

[45] Garrett Birkhoff. Three observations on linear algebra. *Univ. Nac. Tacuman, Rev. Ser. A*, 5:147–151, 1946.

[46] John Von Neumann. A certain zero-sum two-person game equivalent to the optimal assignment problem. *Contributions to the Theory of Games*, 2(0):5–12, 1953.

[47] Jan van den Brand, Yin Tat Lee, Yang P. Liu, Thatchaphol Saranurak, Aaron Sidford, Zhao Song, and Di Wang. Minimum cost flows, MDPs, and ℓ_1-regression in nearly linear time for dense instances. *arXiv preprint arXiv:2101.05719*, 2021.

[48] Nicholas Mattei and Toby Walsh. PrefLib: A library of preference data HTTP://PREFLIB.ORG. In *3rd International Conference on Algorithmic Decision Theory (ADT 2013)*, Lecture Notes in Artificial Intelligence. Springer, 2013.

[49] Gurobi Optimization LLC. Gurobi optimizer reference manual, 2020.

[50] Wenxin Ding, Nihar B. Shah, and Weina Wang. On the privacy-utility tradeoff in peer-review data analysis. *arXiv preprint arXiv:2006.16385*, 2020.

[51] Leman Akoglu, Hanghang Tong, and Danai Koutra. Graph based anomaly detection and description: A survey. *Data Mining and Knowledge Discovery*, 29(3):626–688, 2015.

[52] Bryan Hooi, Hyun Ah Song, Alex Beutel, Neil Shah, Kijung Shin, and Christos Faloutsos. FRAUDAR: Bounding graph fraud in the face of camouflage. In *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 895–904, 2016.

[53] Srijan Kumar, Bryan Hooi, Disha Makhija, Mohit Kumar, Christos Faloutsos, and V.S. Subrahmanian. REV2: Fraudulent user prediction in rating platforms. In *Eleventh ACM International Conference on Web Search and Data Mining*, pages 333–341, 2018.

[54] Guan Wang, Sihong Xie, Bing Liu, and S. Yu Philip. Review graph based online store review spammer detection. In *2011 IEEE 11th International Conference on Data Mining*, pages 1242–1247. IEEE, 2011.

[55] Leman Akoglu, Rishi Chandy, and Christos Faloutsos. Opinion fraud detection in online reviews by network effects. In *Seventh International Conference on Weblogs and Social Media*, 2013.

[56] Zhihai Yang, Lin Xu, Zhongmin Cai, and Zongben Xu. Re-scale AdaBoost for attack detection in collaborative filtering recommender systems. *Knowledge-Based Systems*, 100:74–88, 2016.

[57] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T. Hancock. Finding deceptive opinion spam by any stretch of the imagination. *arXiv preprint arXiv:1107.4557*, 2011.

[58] Mingdan Si and Qingshan Li. Shilling attacks against collaborative recommender systems: A review. *Artificial Intelligence Review*, 53(1):291–319, 2020.

[59] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, 2009. Chapter 26.3.

[60] Efim A Dinic. Algorithm for solution of a problem of maximum flow in networks with power estimation. In *Soviet Math. Doklady*, volume 11, pages 1277–1280, 1970.

[61] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Springer, 1972.

Steven Jecmen
Carnegie Mellon University
Email: `sjecmen@cs.cmu.edu`

Hanrui Zhang
Duke University
Email: `hrzhang@cs.duke.edu`

Ryan Liu
Carnegie Mellon University
Email: `ryanliu@andrew.cmu.edu`

Nihar B. Shah
Carnegie Mellon University
Email: `nihars@cs.cmu.edu`

Vincent Conitzer
Duke University
Email: `conitzer@cs.duke.edu`

Fei Fang
Carnegie Mellon University
Email: `feif@cs.cmu.edu`

Appendices

A Additional Related Literature

We here review some additional related literature.

A concurrent work [50] considers a different set of problems in releasing data in peer review while preserving reviewer anonymity. The data to be released here are some function of the scores and the reviewer-assignment, whereas we look to release the similarities and the assignment code. Moreover, the approach and techniques in [50] are markedly different—they consider post-processing the data for release using techniques such as differential privacy, whereas we consider randomizing the assignment for plausible deniability.

Outside of peer review, there is a line of prior work focused on detecting and mitigating manipulation in online reviews (such as those on Yelp or Amazon) [51–58]. These works typically make assumptions that are not applicable or require data that is not available in our setting. Several of these works analyze the graph of user reviews in order to detect fraudulent reviewers [51]. For example, [52] detects fraud from the review graph by assuming that products are trying to maximize the number of positive reviews they get, whereas in our setting it is important to mitigate the effectiveness of just a single author-paper collusion since the number of reviews per paper is fixed and small. Additionally, some of these works [53–55] are based on estimating the "true quality" of each item from the review graph, which is not possible in peer review since paper evaluations are subjective. Another direction of prior work uses machine learning to detect malicious behavior. Some research [56]
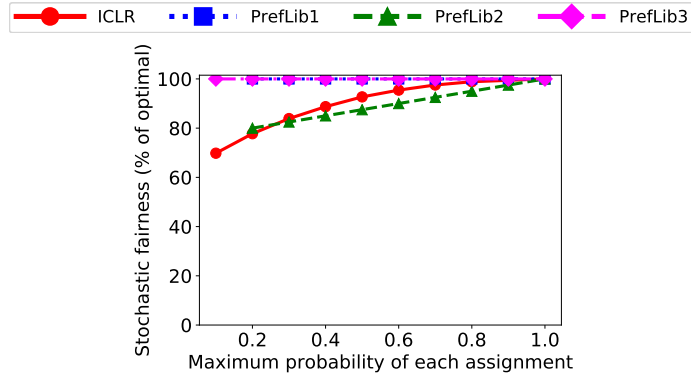
Figure 2: Experimental results for the Fair Pairwise-Constrained Problem.

detects fraud from reviewer statistics such as rating variance or number of ratings, but in the peer review setting, there is so little data for each reviewer that such features would be highly noisy or completely uninformative. Other works [57] attempt to detect malicious reviews from the review text, but in our setting, malicious reviews may not differ stylistically from genuine reviews. Finally, some work in recommender systems focuses on making product recommendations resistant to malicious reviews [58]; however, in peer review, the paper acceptance process must be done by hand since it must take into account reviewer opinions, arguments, and the reviewer discussion.

B Stochastic Fairness Objective

An alternate objective to the sum-similarity objective has been studied in past work [33, 34], aiming to improve the fairness of the assignment with respect to the papers. Rather than maximizing the sum-similarity across all papers, this objective maximizes the minimum total similarity assigned to any paper:

$$
\begin{aligned}
\underset{M \in \mathbb{R}^{n \times d}}{\arg\max} \quad & \min_{p \in \mathcal{P}} \sum_{r \in \mathcal{R}} S_{rp} M_{rp} \\
\text{subject to} \quad & M_{rp} \in \{0, 1\} && \forall r \in \mathcal{R}, \forall p \in \mathcal{P} \\
& \sum_{p \in \mathcal{P}} M_{rp} \leq k && \forall r \in \mathcal{R} \\
& \sum_{r \in \mathcal{R}} M_{rp} = \ell && \forall p \in \mathcal{P}.
\end{aligned}
$$

Due to the minimum in the objective, this problem is NP-hard [33]; the paper [34] presents an algorithm to find an approximate solution.

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

Definition 4 (Fair Pairwise-Constrained Problem). *The input to the problem is a similarity matrix S and a matrix $Q \in [0, 1]^{n \times d}$. The goal is to find a randomized assignment of papers to reviewers that maximizes $\min_{p \in \mathcal{P}} \mathbb{E}\left[\sum_{r \in \mathcal{R}} S_{rp} M_{rp}\right]$ subject to the constraints that $\mathbb{P}[M_{rp} = 1] \leq Q_{rp}, \forall r \in \mathcal{R}, p \in \mathcal{P}$.*

This problem definition is identical to that of the Pairwise-Constrained Problem (Definition 1), with the exception that the objective to maximize is now the stochastic fairness objective rather than the sum-similarity. Note that this objective is not equal to the "expected fairness" (i.e., $\mathbb{E}\left[\min_{p \in \mathcal{P}} \sum_{r \in \mathcal{R}} S_{rp} M_{rp}\right]$), but by Jensen's inequality it is an upper bound on the expected fairness.

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

Theorem 4. *There exists an algorithm which returns an optimal solution to the Fair Pairwise-Constrained Problem in poly(n, d) time.*

We now present our algorithm for solving the Fair Pairwise-Constrained Problem, thereby proving the theorem. It proceeds in a similar manner as the algorithm for the Pairwise-Constrained Problem presented in Section 4.

The algorithm first finds an optimal fractional assignment matrix, since the stochastic fairness objective depends only on the marginal probabilities in the fractional assignment matrix. The optimal fractional assignment is found by the following LP, which we call $\mathcal{LP}3$:

$$\underset{F \in \mathbb{R}^{n \times d}, x \in \mathbb{R}}{\arg \max} \quad x \tag{7}$$

$$\text{subject to} \quad 0 \leq F_{rp} \leq 1 \qquad \forall r \in \mathcal{R}, \forall p \in \mathcal{P} \tag{8}$$

$$\sum_{p \in \mathcal{P}} F_{rp} \leq k \qquad \forall r \in \mathcal{R} \tag{9}$$

$$\sum_{r \in \mathcal{R}} F_{rp} = \ell \qquad \forall p \in \mathcal{P} \tag{10}$$

$$F_{rp} \leq Q_{rp} \qquad \forall r \in \mathcal{R}, \forall p \in \mathcal{P} \tag{11}$$

$$x \leq \sum_{r \in \mathcal{R}} S_{rp} F_{rp} \qquad \forall p \in \mathcal{P}. \tag{12}$$

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

Once an optimal fractional assignment matrix has been found, it only remains to sample a deterministic assignment from it. This is done with the sampling algorithm described in Section 4.2, just as in the Pairwise-Constrained Problem.

We now present some empirical results for this algorithm on the four conference datasets described in Section 6. We set all entries of Q equal to the same constant value q_0 (varied on the x-axis), and observe how the stochastic fairness objective of the assignment changes as q_0 increases from 0.1 to 1 with an interval of 0.1. Since the expectation is inside a minimum in the objective, the objective cannot be estimated without bias by averaging together the stochastic fairness of sampled deterministic assignments. Due to this difficulty, we plot the exact objective of our randomized assignment (i.e., the optimal objective value of $\mathcal{LP}3$) rather than averaging over multiple samples, and report the objective as a percentage of the unconstrained optimal solution's objective (that is, the algorithm's solution when $q_0 = 1$). As Figure 2 shows, our algorithm finds a randomized assignment achieving 92.7% of the optimal fairness objective on the ICLR dataset when $q_0 = 0.5$.

C Bad-Assignment Probability Problem Variants

An input to both the Pairwise-Constrained Problem (Definition 1) and the Partition-Constrained Problem (Definition 3) is the matrix Q, where Q_{rp} denotes the maximum

probability with which reviewer r should be assigned to paper p. In practice, program chairs can set the values in this matrix based on their own beliefs about each reviewer-paper pair. However, it may be difficult for program chairs to translate their beliefs about the risk of assigning any reviewer-paper pair into appropriate values for Q. In this appendix, we define alternate versions of these problems that allow the program chairs to codify their beliefs in a different way.

Define the assignment of reviewer r to paper p as "bad" if reviewer r intends to untruthfully review paper p (either because they intend to give a dishonest favorable review or because they intend to torpedo-review). Further define a matrix $W \in [0,1]^{n \times d}$ of bad-assignment probabilities, where W_{rp} represents the probability that the assignment of reviewer r to paper p would be a bad assignment; we assume that the events of each reviewer-paper assignment being bad are all independent of each other. The "true value" of W may not be known, but it can be set based on the program chairs' beliefs about the reviewers and authors or potentially estimated based on some data from prior conferences. The problem variants we present in the following subsections make use of these bad-assignment probabilities.

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

C.1 Handling Bad Reviewer-Paper Assignments

We define an alternate version of the Pairwise-Constrained Problem using the bad-assignment probabilities:

Definition 5 (Bad-Assignment Probability Pairwise-Constrained Problem)**.** *The input to the problem is a similarity matrix S, a matrix $W \in [0,1]^{n \times d}$ of bad-assignment probabilities, and a value $\lambda \in [0,1]$. The goal is to find a randomized assignment of papers to reviewers that maximizes $\mathbb{E}\left[\sum_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}} S_{rp} M_{rp}\right]$ subject to the constraints that $W_{rp}\mathbb{P}[M_{rp} = 1] \leq \lambda, \forall r \in \mathcal{R}, p \in \mathcal{P}$.*

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

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

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

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

- Solve the Pairwise-Constrained Problem using the algorithm from Theorem 1, described in Section 4.

C.2   Handling Bad Pairs of Reviewers

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

C.2.1   Constraints on Disjoint Reviewer Sets

In the same way as done above for the Pairwise-Constrained Problem, we define an alternate version of the Partition-Constrained Problem:

Definition 6 (Bad-Assignment Probability Partition-Constrained Problem). *The input to the problem is a similarity matrix S, a matrix $W \in [0,1]^{n \times d}$ of bad-assignment probabilities, a value $\lambda \in [0,1]$, and a partition of the reviewer set into subsets $I_1, \ldots, I_m \subseteq \mathcal{R}$. The goal is to find a randomized assignment of papers to reviewers that maximizes $\mathbb{E}\left[\sum_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}} S_{rp} M_{rp}\right]$ subject to the constraints that $W_{rp}\mathbb{P}[M_{rp} = 1] \leq \lambda, \forall r \in \mathcal{R}, p \in \mathcal{P}$ and $\mathbb{P}[M_{ap} = 1 \wedge M_{bp} = 1] = 0, \forall a, b \in I_i, \forall i \in [m]$.*

Just as for the Bad-Assignment Probability Pairwise-Constrained Problem, we solve this problem by first transforming an instance of this problem into an equivalent instance of the Partition-Constrained Problem, done by constructing a matrix of maximum probabilities Q where $Q_{rp} = \min\left(\lambda/W_{rp}, 1\right), \forall r \in \mathcal{R}, p \in \mathcal{P}$. We then solve this instance using the algorithm in Section 5.

C.2.2   Constraints on the Expected Number of Bad Reviewers

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

The following problem states this goal:

Definition 7 (Bad-Assignment Probability Expectation-Constrained Problem). *The input to the problem is a similarity matrix S, a matrix $W \in [0,1]^{n \times d}$ of bad-assignment probabilities, a value $\lambda \in [0,1]$, and a value $\mu \in \mathbb{R}$. The goal is to find a randomized assignment of papers to reviewers that maximizes $\mathbb{E}\left[\sum_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}} S_{rp} M_{rp}\right]$ subject to the constraints that $W_{rp}\mathbb{P}[M_{rp} = 1] \leq \lambda, \forall r \in \mathcal{R}, p \in \mathcal{P}$ and $\sum_{r \in \mathcal{R}} W_{rp}\mathbb{E}[M_{rp}] \leq \mu, \forall p \in \mathcal{P}$.*

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

greater than μ:

$$\underset{F \in \mathbb{R}^{n \times d}}{\arg\max} \quad \sum_{p \in \mathcal{P}} \sum_{r \in \mathcal{R}} S_{rp} F_{rp} \tag{13}$$

$$\text{subject to} \quad 0 \leq F_{rp} \leq 1 \qquad\qquad \forall r \in \mathcal{R}, \forall p \in \mathcal{P} \tag{14}$$

$$\sum_{p \in \mathcal{P}} F_{rp} \leq k \qquad\qquad \forall r \in \mathcal{R} \tag{15}$$

$$\sum_{r \in \mathcal{R}} F_{rp} = \ell \qquad\qquad \forall p \in \mathcal{P} \tag{16}$$

$$F_{rp} W_{rp} \leq \lambda \qquad\qquad \forall r \in \mathcal{R}, \forall p \in \mathcal{P} \tag{17}$$

$$\sum_{r \in \mathcal{R}} F_{rp} W_{rp} \leq \mu \qquad\qquad \forall p \in \mathcal{P}. \tag{18}$$

Constraints (14-16) define the space of fractional assignment matrices, Constraint (17) ensures that the probability of each bad assignment occurring is limited at λ, and Constraint (18) ensures that the expected number of bad reviewer-paper assignments for each paper is at most μ. Therefore, $\mathcal{LP}4$ finds the optimal fractional assignment for the Bad-Assignment Probability Expectation-Constrained Problem. This fractional assignment can then be sampled from using the sampling algorithm in Section 4.2.

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

D Algorithms for the Pairwise-Constrained Problem

In Section 4, we briefly sketched the sampling algorithm that realizes Theorem 1, a simplified version of the algorithm from [44]. Here, we first present this sampling algorithm in detail, thus solving the Pairwise-Constrained Problem (Definition 1). We then present a supplementary algorithm to compute the full distribution over deterministic assignments, which [44] does not. Knowing the full distribution may be useful in order to compute other properties of the randomized assignment not calculable from F directly.

D.1 Sampling Algorithm

Recall that $\mathcal{LP}1$ finds the optimal marginal assignment probabilities F for the Pairwise-Constrained Problem. In this subsection, we present the sampling algorithm that implements

Algorithm 1 Sampling algorithm for the Pairwise-Constrained Problem.

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

Ouput: Deterministic assignment matrix M

Algorithm:

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

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

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

5: **while** $\exists e \in E$ such that $f(e) \notin \mathbb{Z}$ **do**
6:      Find a cycle of edges (ignoring direction) $C = \{e_1, \ldots, e_k\}$ such that $f(e_i) \notin \mathbb{Z}, \forall i \in [k]$

7:      $A \leftarrow \{e \in C | \ e$ is directed in the same direction as e_1 along the cycle$\}$
8:      $B \leftarrow C \setminus A$
9:      $\alpha \leftarrow \min \left(\min_{e \in A} f(e), \min_{e \in B} h(e) - f(e) \right)$
10:      **for** $e \in A$ **do**
11:          $f_1(e) \leftarrow f(e) - \alpha$
12:      **end for**
13:      **for** $e \in B$ **do**
14:          $f_1(e) \leftarrow f(e) + \alpha$
15:      **end for**
16:      $\beta \leftarrow \min \left(\min_{e \in A} h(e) - f(e), \min_{e \in B} f(e) \right)$
17:      **for** $e \in A$ **do**
18:          $f_2(e) \leftarrow f(e) + \beta$
19:      **end for**
20:      **for** $e \in B$ **do**
21:          $f_2(e) \leftarrow f(e) - \beta$
22:      **end for**
23:      $\gamma \leftarrow \frac{\beta}{\alpha + \beta}$
24:      With probability γ, $f \leftarrow f_1$; else $f \leftarrow f_2$
25: **end while**
26: $M_{rp} = f((r,p)), \forall (r,p) \in \mathcal{R} \times \mathcal{P}$

these marginal probabilities as a randomization over deterministic paper assignments. The sampling algorithm proceeds as follows. We begin by constructing a directed graph $G = (V, E)$ for our problem, along with a capacity function $h : E \to \mathbb{Z}$ (Lines 1-3). First, construct one vertex for each reviewer, one vertex for each paper, and source and destination vertices s, t. Add an edge from the source vertex to each reviewer's vertex with capacity k. Add an edge from each paper's vertex to the destination vertex with capacity ℓ. Finally, add an edge from each reviewer to each paper with capacity 1. We also construct a flow function $f : E \to \mathbb{R}$, which obeys the flow conservation constraints $\sum_{e \in E \cap (V \times \{v\})} f(e) = \sum_{e \in E \cap (\{v\} \times V)} f(e), \forall v \in V \setminus \{s, t\}$ and the capacity constraints $f(e) \leq h(e), \forall e \in E$ (Line 4). A (possibly fractional) assignment F can be represented as a flow on this graph, where the flow from reviewer i to

paper j corresponds to the probability reviewer i is assigned to paper j and the other flows are set uniquely by flow conservation. Due to the load constraints on assignments, the flows on the edges from the papers to the destination must be equal to those edges' capacities and the flows on the edges from the source to the reviewers must be less than or equal to the capacities.

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

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

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

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

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

D.2   Decomposition Algorithm

We here provide a decomposition algorithm to compute a full distribution over deterministic assignments for a given fractional assignment matrix (which the prior work [44] does not). For simplicity, we assume here that all reviewer loads are met with equality (that is, $\sum_{p \in \mathcal{P}} F_{rp} = k$ for all $r \in \mathcal{R}$); the extension to the case when reviewer loads are met with inequality is simple.

We first define certain concepts necessary for the algorithm. We then present a subroutine of the algorithm and prove its correctness. We then present the overall algorithm and prove

its correctness. Finally, we analyze the time complexity of the algorithm.

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

- A capacitated matching instance consists of a set of papers $\mathcal{P}$, a set of reviewers $\mathcal{R}$, and a capacity function $h : \mathcal{P} \cup \mathcal{R} \to \mathbb{Z}$. A solution to $(\mathcal{P}, \mathcal{R}, h)$ is a matrix $F \in [0, 1]^{n \times d}$, where for any $p \in \mathcal{P}$,

$$\sum_{r \in \mathcal{R}} F_{rp} = h(p),$$

  and for any $r \in \mathcal{R}$,

$$\sum_{p \in \mathcal{P}} F_{rp} = h(r).$$

  The solution F is integral if $F_{rp} \in \{0, 1\}$ for all $p \in \mathcal{P}$ and $r \in \mathcal{R}$.

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

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

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

1. Let $E \subseteq \mathcal{R} \times \mathcal{P}$ be $E = \{(r, p) \mid F_{rp} \in (0, 1)\}$, and let $M_0 \subseteq \mathcal{R} \times \mathcal{P}$ be $M_0 = \{(r, p) \mid F_{rp} = 1\}$. With this, define capacity function h' as, for any $p \in \mathcal{P}$,

$$h'(p) = h(p) - |\{(r, p) \mid r \in \mathcal{R}\} \cap M_0|$$

  and for any $r \in \mathcal{R}$,

$$h'(r) = h(r) - |\{(r, p) \mid p \in \mathcal{P}\} \cap M_0|.$$

2. Find a maximum matching $M \subseteq E$ on E subject to capacity constraints h'.

3. Set F_0 as

$$(F_0)_{rp} = \mathbb{I}\left[(r, p) \in M \cup M_0\right], \forall r \in \mathcal{R}, p \in \mathcal{P}.$$

  Set F' as

$$F'_{rp} = \frac{1}{(1 - \alpha_0)}(F_{rp} - \alpha_0 (F_0)_{rp}), \forall r \in \mathcal{R}, p \in \mathcal{P}.$$

  Set $\alpha_0 = \min(\{F_{rp} \mid (r, p) \in M\} \cup \{1 - F_{rp} \mid (r, p) \in E \setminus (M \cup M_0)\})$.

We prove the correctness of this subroutine in Lemma 3. Before we do, we restate a result from prior work [44] that we use in the proof, using our own notation.

Lemma 2 ([44, Thm. 1]).   *For any $(\mathcal{P}, \mathcal{R}, h)$ and any solution F to $(\mathcal{P}, \mathcal{R}, h)$, there exists some $z \in \mathbb{Z}$, integral solutions $\{F_1, \dots, F_z\}$ to $(\mathcal{P}, \mathcal{R}, h)$, and α lying on the z-dimensional simplex, such that $F = \sum_{i=1}^{z} \alpha_i F_i$.*

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

Lemma 3. *The decomposition subroutine finds F_0, α_0, and F', such that (i) F_0 is an integral solution to $(\mathcal{P}, \mathcal{R}, h)$, (ii) F' is a fractional solution to $(\mathcal{P}, \mathcal{R}, h)$, (iii) F' has strictly fewer fractional entries than F, and (iv) $F = \alpha_0 F_0 + (1 - \alpha_0) F'$.*

Proof. We first consider (i). The key step is to show that the maximum matching M found in step 2 is a perfect matching with respect to h', or equivalently, to show there is a perfect matching on E with respect to h'. Consider the capacitated matching instance $(\mathcal{P}, \mathcal{R}, h')$, and the solution F'' where

$$F''_{rp} = \begin{cases} F_{rp} & \text{if } F_{rp} < 1 \\ 0 & \text{otherwise.} \end{cases}$$

F'' is a solution to $(\mathcal{P}, \mathcal{R}, h')$ by the construction of h'. By Lemma 2, F'' is a convex combination of integral solutions to $(\mathcal{P}, \mathcal{R}, h')$. For some z, let $\{F_1, \ldots, F_z\}$ and α be such a decomposition of F'', where each F_i is an integral solution to $(\mathcal{R}, \mathcal{P}, h')$ and α_i is its associated weight. For each $i \in [z]$, let $M_i \subseteq \mathcal{R} \times \mathcal{P}$ be the set of (r, p) pairs where $(F_i)_{rp} = 1$. Since F_i is a solution to $(\mathcal{R}, \mathcal{P}, h')$, M_i is a perfect matching with respect to h'. By the definition of F'', $(r, p) \in E$ if and only if $F''_{rp} > 0$. Now since $F'' = \sum_{i=1}^{z} \alpha_i F_i$, $E = \bigcup_{i=1}^{z} M_i$. Since each M_i is a perfect matching with respect to h', E contains a perfect matching with respect to h' and so the maximum matching M found is in fact a perfect matching with respect to h'. Therefore, $M \cup M_0$ is a perfect matching with respect to h by the definition of h'. Therefore, F_0 is an integral solution to $(\mathcal{P}, \mathcal{R}, h)$.

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

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

and

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

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

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

and

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

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

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

Finally, (iv) holds by the construction of F_0 and F'. $\square$

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

1. If F is integral, return solution $\{F\}$ and weight 1.

2. Otherwise, decompose F into F_0 (with weight α_0) and F' using the above subroutine.

3. Recursively call this algorithm with $(\mathcal{P}, \mathcal{R}, h)$ and F' as input, decomposing F' into solutions $\{F_1, \ldots, F_z\}$ with weights α.

4. Define $\beta = (1 - \alpha_0)\alpha$. Return the solutions $\{F_0, F_1, \ldots, F_z\}$ with weights $(\alpha_0, \beta_1, \ldots, \beta_z)$.

We now prove the correctness of this algorithm.

Theorem 5. *The decomposition algorithm correctly outputs integral solutions $\{F_1, \ldots, F_z\}$ to $(\mathcal{P}, \mathcal{R}, h)$ and α lying on the z-dimensional simplex, such that $F = \sum_{i=1}^{z} \alpha_i F_i$.*

Proof. We prove this statement by induction. If the algorithm returns in step 1, the theorem's statement holds. Now, assume that the theorem's statement holds for the decomposition returned by the recursive call to the algorithm in step 3, so that the following all hold: $\{F_1, \ldots, F_z\}$ are integral solutions to $(\mathcal{P}, \mathcal{R}, h)$, α lies on the z-dimensional simplex, and $F' = \sum_{i=1}^{z} \alpha_i F_i$. By Lemma 3, F_0 is an integral solution to $(\mathcal{P}, \mathcal{R}, h)$, so the $z + 1$ solutions returned in step 4 are integral solutions to $(\mathcal{P}, \mathcal{R}, h)$. Since $\alpha_0 \in [0, 1]$, $\beta \in [0, 1]^z$, and $\alpha_0 + \sum_{i=1}^{z} \beta_z = 1$, the weights returned in step 4 lie on the $z + 1$ dimensional simplex. Finally, by Lemma 3,

$$F = \alpha_0 F_0 + (1 - \alpha_0) F'$$

$$= \alpha_0 F_0 + (1 - \alpha_0) \sum_{i=1}^{z} \alpha_i F_i$$

$$= \alpha_0 F_0 + \sum_{i=1}^{z} \beta_i F_i.$$

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

This decomposition algorithm can be used as part of the algorithm that solves the Pairwise-Constrained Problem, substituting for the sampling algorithm described in Section 4.2. It finds the full decomposition of the fractional assignment matrix into deterministic assignments rather than sampling a deterministic assignment. The capacity function h used as the original input to the algorithm is defined as $h(r) = k, \forall r \in \mathcal{R}$ and $h(p) = \ell, \forall p \in \mathcal{P}$, and the input solution F is exactly the fractional assignment matrix found as the solution to $\mathcal{LP}1$. The output integral solutions represent deterministic assignments, and the corresponding weights represent the probability with which each assignment should be chosen.

Time complexity: Since F' has at least one fewer fractional entry than F, the recursive procedure has depth $O(dn)$ and therefore makes $O(dn)$ calls to the decomposition subroutine. In each call, the bottleneck is finding a maximum matching on E subject to capacities h. This can be solved as a max-flow problem on a graph with $O(d + n)$ vertices and $O(dn)$ edges [59]. Using Dinic's algorithm [60], the computation of each matching takes $O(dn(d + n)^2)$ time, giving an overall time complexity of $O(d^2 n^2 (d + n)^2)$.

E Proof of Theorem 2 and a Corollary

We first define a decision variant of the Triplet-Constrained Problem, called "Arbitrary-Constraint Feasibility." An instance of this problem is defined by the paper and reviewer loads ℓ and k, and a 3-dimensional tensor $T \in [0, 1]^{n \times n \times d}$. For all $i, j \in \mathcal{R}, i \neq j$ and for all $p \in \mathcal{P}$, T_{ijp} denotes the maximum probability that both reviewers i and j are assigned to paper p. The question is: does there exist a randomized assignment that obeys the

constraints given by T? We next show that Arbitrary-Constraint Feasibility is NP-hard by a reduction from 3-Dimensional Matching.

An instance of 3-Dimensional Matching consists of three sets X, Y, Z of size s, and a collection of tuples in $X \times Y \times Z$. It asks whether there exists a selection of s tuples that includes each element of X, Y, and Z exactly once. This problem is known to be NP-complete [61].

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

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

If the 3-Dimensional Matching instance is a yes (that is, there exists a valid selection of s tuples), then consider the paper assignment that assigns the corresponding reviewers and paper within each triple in the matching. Each paper has exactly 2 reviewers and each reviewer has exactly 1 paper, so this is a deterministic assignment. Since it includes only the triples in the matching instance, it obeys the probability constraints of T, so the Arbitrary-Constraint Feasibility instance is a yes.

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

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

Theorem 2 implies a more fundamental result about the feasible region of implementable reviewer-reviewer-paper probability tensors, that is, the tensors $G \in [0,1]^{n \times n \times d}$ where entry G_{ijp} represents the marginal probability that both reviewers i and j are assigned to paper p under some randomized assignment. We can represent any deterministic assignment by a 3-dimensional tensor $M \in \{0,1\}^{n \times n \times d}$ where $M_{ijp} = 1$ if and only if both reviewers i and j are assigned to paper p. Just as in the earlier case of fractional assignment matrices, the set of implementable probability tensors is a polytope with deterministic assignment

tensors at the vertices (since any implementable probability tensor is a convex combination of deterministic assignment tensors). For fractional reviewer-paper assignment matrices, this polytope was defined by a small number ($O(dn)$) of linear inequalities, despite the fact that it has a large number of vertices (factorial in d and n). However, this is no longer the case for reviewer-reviewer-paper probabilities.

Corollary 2. *The polytope of implementable reviewer-reviewer-paper probabilities is not expressible in a polynomial (in n and d) number of linear inequality constraints (assuming $P \neq NP$).*

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

F Sampling Algorithm for the Partition-Constrained Problem and Related Proofs

In Section 5.1, we briefly sketched the sampling algorithm that realizes Lemma 1 and Corollary 1. Here, we first present this algorithm in detail. We then present proofs of the guarantees.

F.1 Sampling Algorithm

The algorithm realizing Lemma 1 and Corollary 1 is obtained by changing three lines in Algorithm 1 (described in Appendix D), as follows:

- Line 6 is replaced with the subroutine in Algorithm 2.

- Line 9 is changed to:

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

- Line 16 is changed to:

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

The primary modification we make to Algorithm 1 is replacing Line 6 with the subroutine in Algorithm 2. In each iteration, when we look for an undirected cycle of fractional edges in the graph, we now choose the cycle carefully rather than arbitrarily. We find a cycle by starting from an arbitrary fractional edge in the graph and walk along adjacent fractional edges (ignoring direction) until we repeat a previously-visited vertex. As we do this, whenever we take a fractional edge from a reviewer in subset I into paper p, there are two cases.

- Case 1: If there exists a different fractional edge from paper p to subset I (Line 8 in Algorithm 2), we take this edge next. Note that if the total load from subset I on paper p is integral, such an edge must exist.

Algorithm 2 Loop-finding subroutine (replacing Line 6 in Algorithm 1).

1: Construct the set of undirected edges $E_U \leftarrow E \cup \{(v, u) \mid (u, v) \in E\}$
2: Construct the undirected flow function $f_U : E_U \rightarrow \mathbb{R}$ as $f_U((u, v)) \leftarrow$
$\begin{cases} f((u, v)) & \text{if } (u, v) \in E \\ f((v, u)) & \text{otherwise} \end{cases}$
3: Find arbitrary edge $(u, v) \in E$ such that $f((u, v)) \notin \mathbb{Z}$
4: $C \leftarrow \{(u, v)\}$
5: $D_1 \leftarrow \{\}, D_2 \leftarrow \{\}$
6: **while** v has not previously been visited **do**
7:     Visit v
8:     **if** $u \in \mathcal{R}$ and $v \in \mathcal{P}$ **then**
9:         Set $I \in \{I_1, \ldots, I_m\}$ such that $u \in I$
10:         **if** $\exists w \in I \setminus \{u\}$ such that $(v, w) \in E_U$ and $f_U((v, w)) \notin \mathbb{Z}$ **then**
11:             Find such a w
12:         **else**
13:             For some $J \in \{I_1, \ldots, I_m\} \setminus \{I\}$ such that $\sum_{r \in J} f((r, v)) \notin \mathbb{Z}$, find $w \in J$ such that $(v, w) \in E_U$ and $f_U((v, w)) \notin \mathbb{Z}$
14:             $D_1 \leftarrow D_1 \cup \{\sum_{r \in I} f((r, v))\}$ (corresponding to (u, v))
15:             $D_2 \leftarrow D_2 \cup \{\sum_{r \in J} f((r, v))\}$ (corresponding to (v, w))
16:         **end if**
17:     **else**
18:         Find $w \in V \setminus \{u\}$ such that $(v, w) \in E_U$ and $f_U((v, w)) \notin \mathbb{Z}$
19:     **end if**
20:     $C \leftarrow C \cup \{(v, w)\}$
21:     $u \leftarrow v$
22:     $v \leftarrow w$
23: **end while**
24: Set e_1 as the first edge in C leaving v
25: Set e_{-1} as the last edge in C (entering v)
26: Remove edges preceding e_1 from C, and remove the corresponding elements from D_1 and D_2
27: **if** $v \in \mathcal{P}$ and $\exists I \in \{I_1, \ldots, I_m\}$ such that $e_1 \in \{v\} \times I$ and $e_{-1} \in I \times \{v\}$ **then**
28:     Remove the elements corresponding to e_1 and e_{-1} from D_1 and D_2
29: **end if**
30: **if** $e_1 \notin E$ **then**
31:     Swap D_1 and D_2
32: **end if**
33: Replace each edge in C from E_U with the corresponding edge from E

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

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

subset I or J on paper p is integral, for any I, J, p found in Case 2. The total loads from each subset on each paper found in Case 2 are saved in either set D_1 or set D_2 depending on the direction of the corresponding edges in the cycle, and each subset-paper pair with an edge corresponding to an element of D_1 or D_2 has only that one edge in the cycle. If the total (fractional) load from subset I on paper p is t, then only $\lceil t \rceil - t$ additional flow can be added to any edge from subset I to paper p before the load becomes integral; similarly, only $t - \lfloor t \rfloor$ flow can be removed from any edge before the load becomes integral. This leads to the stated changes to Lines 9 and 16 in Algorithm 1.

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

F.2   Proof of Lemma 1 and Corollary 1

Proof of Lemma 1:   We first prove part (i) of the lemma. Consider any subset I and any paper p, and recall that in Section 5.1 we showed that the algorithm presented there has the property that the total load on each paper from each subset is preserved exactly if originally integral and rounded in either direction if originally fractional. If the total load from subset I on paper p is less than or equal to 1 originally (i.e., $\sum_{r \in I} F_{rp} \leq 1$), then this algorithm will only ever sample assignments with either 0 or 1 reviewers, so it never samples a integral assignment that assigns two reviewers from subset I to paper p.

We now prove part (ii) of the lemma. Suppose that the total load from subset I on paper p is originally strictly greater than 1 (i.e., $\sum_{r \in I} F_{rp} > 1$). Let X denote a random variable that represents the number of reviewers from subset I on paper p, that is, $X = \sum_{r \in I} M_{rp}$. Hence, we have $\mathbb{E}[X] = \sum_{r \in I} F_{rp} > 1$. Suppose that we implement the marginal probabilities F as a distribution over deterministic assignments that places zero mass on any deterministic assignment where $X \geq 2$. Since X is integral in any deterministic assignment, all of the mass must be placed on deterministic assignments where $X \leq 1$. Since $\mathbb{E}[X] > 1$, this is impossible. Therefore, F cannot be implemented without having some probability of placing two reviewers from subset I on paper p, so the expected number of pairs of reviewers from subset I assigned to paper p must be non-zero for any sampling algorithm.

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

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

First, assume that $\mathbb{E}[X] = \mu$ is integral. $\mathbb{E}[X] = \sum_{r \in I} F_{rp}$, so μ is equal to the total load from subset I on paper p. From Section 5.1, our sampling algorithm preserves exactly the loads from any subset on any paper that are originally integral, meaning that it will

always assign exactly μ reviewers from subset I to paper p. In other words, our sampling algorithm always gives the distribution of X where $f(\mu) = 1$ and $f(i) = 0$ for $i \neq \mu$. Since all distributions of X have the same expectation, $\sum_{i=0}^{|I|} f'(i)i = \mu$; we also know that $f'(i) > 0$ for some $i \neq \mu$. For this distribution, we have that

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

Now, suppose that $\mathbb{E}[X] = \mu$ is not integral. From Section 5.1, our sampling algorithm rounds to a neighboring integer the loads from any subset on any paper that are originally not integral, meaning that it will always assign exactly $\lceil\mu\rceil$ or $\lfloor\mu\rfloor$ reviewers from subset I to paper p. In other words, our sampling algorithm only places probability mass on outcomes $X = \lceil\mu\rceil$ or $X = \lfloor\mu\rfloor$, so $f(i) = 0$ for $i \notin \{\lceil\mu\rceil, \lfloor\mu\rfloor\}$. There is only one way to do this so that $\mathbb{E}[X] = \mu$; exactly $f(\lceil\mu\rceil) = \mu - \lfloor\mu\rfloor$ and $f(\lfloor\mu\rfloor) = \lceil\mu\rceil - \mu$. Then under this distribution, via some algebraic simplifications,

$$\mathbb{E}_f[X^2] = f(\lceil\mu\rceil)\lceil\mu\rceil^2 + f(\lfloor\mu\rfloor)\lfloor\mu\rfloor^2$$
$$= -\lceil\mu\rceil^2 + \lceil\mu\rceil - \mu + 2\lceil\mu\rceil\mu. \tag{19}$$

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

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

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

$$= \lceil\mu\rceil^2 + 2\lceil\mu\rceil(\mu - \lceil\mu\rceil) + \sum_{\Delta=-\lceil\mu\rceil}^{|I|-\lceil\mu\rceil} \left(f'(\lceil\mu\rceil+\Delta)\Delta^2\right). \tag{20}$$

We want to show that $\mathbb{E}_{f'}[X^2] > \mathbb{E}_f[X^2]$. From (19) and (20), it remains to show that

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

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

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

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

G   Synthetic Simulations

We now present experimental results on synthetic simulations. All results are averaged over 10 trials with error bars plotted representing the standard error of the mean, although error

(a) Pairwise-Constrained Problem

(b) Partition-Constrained Problem

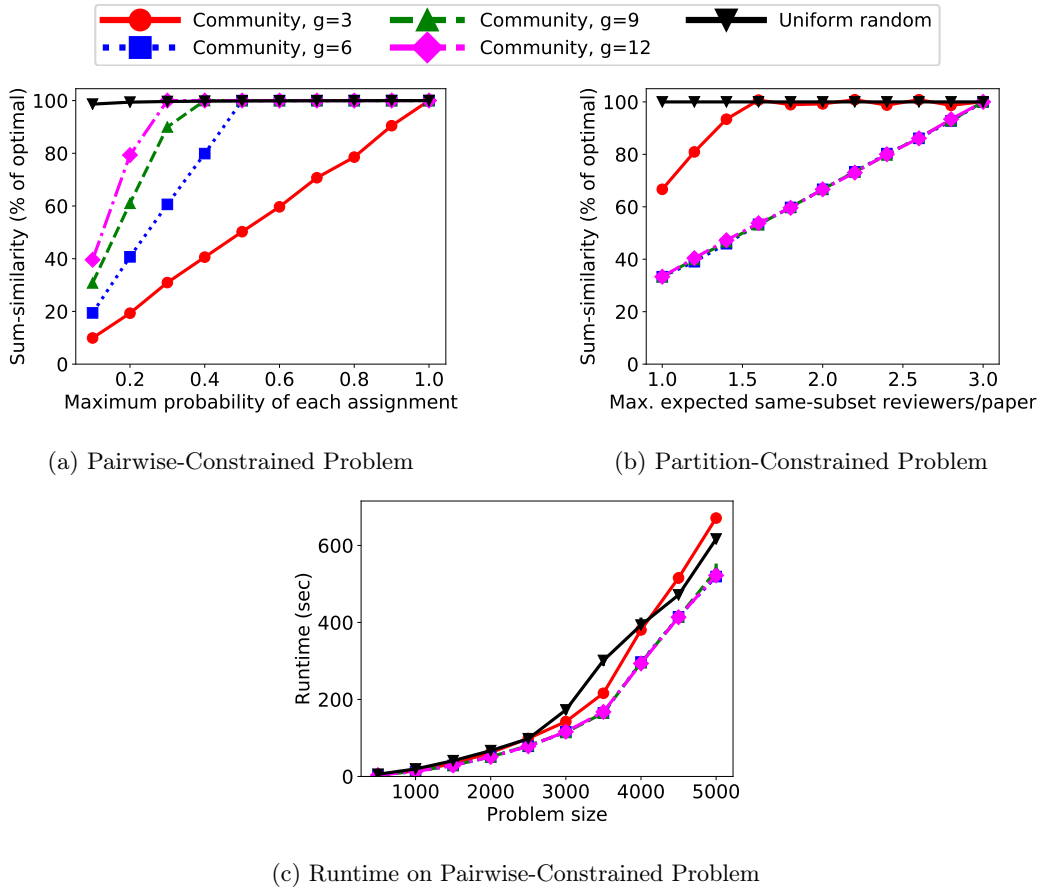(c) Runtime on Pairwise-Constrained Problem

Figure 3: Experimental results on synthetic simulations.

bars are sometimes not visible since the variance is low. All experiments were run on a computer with 8 cores and 16 GB of RAM, running Ubuntu 18.04 and solving the LPs with Gurobi 9.0.2 [49].

We consider two different simulations. First, we consider a simulated "community model" as used in past work [23]. In this model, $n = d = 360$ and $k = \ell = 3$; it is further parameterized by a group size g. For all $i \in \{0, g, 2g, \ldots, n\}$, reviewers i through $i + g - 1$ have similarity 1 with papers i through $i + g - 1$ and similarity 0 with all other papers. We consider four different group sizes g: 3, 6, 9, 12. We also consider a uniform random simulation, where each entry of the similarity matrix is independently and uniformly drawn from $[0, 1)$, fixing $n = d = 1000$ and $k = \ell = 3$.

In Figure 3a, we examine the performance of our algorithm for the Pairwise-Constrained Problem. For each simulation, we set all entries of Q to a constant q_0 and observe the sum-similarity as we vary q_0 (on the x-axis). The objective value is reported here as a percentage of the optimal unconstrained solution's objective, as was done in Section 6. For the community models, the group size makes a large difference as to what an acceptable value of q_0 is. For example, with group size 6 and $q_0 = 0.5$, our algorithm will always assign all good reviewers to all papers; however, for any lower value of q_0 it can no longer do this and so the objective deteriorates rapidly. Note that since our algorithm is optimal, this deterioration is due to the problem being overconstrained for low values of q_0 and not due to an issue with the algorithm. For the uniform random simulation, our algorithm performs

32

very well, since there are likely many reviewers with high similarity for each paper.

We also examine the performance of our algorithm for the Partition-Constrained Problem in Figure 3b. For each simulation, we fix $q_0 = 0.5$ and gradually loosen Constraint (6) in $\mathcal{LP}2$ by increasing the constant from 1 to 3 in increments of 0.2, shown on the x-axis. We plot the sum-similarity objective of the resulting assignment, reported as a percentage of the optimal non-partition-constrained solution's objective (that is, the solution to the Pairwise-Constrained Problem with $q_0 = 0.5$). For the community model simulations, we assign all reviewers in each group to the same subset of the partition. Since all of the reviewers who can review each paper well are in the same subset, this presents a highly constrained problem (which our algorithm is solving optimally). As expected, our algorithm trades off the number of same-subset reviewer pairs assigned to the same paper and the sum-similarity objective rather poorly (as would any other algorithm). Since $q_0 = 0.5$, there is no difference between the cases with group size 6 or greater. For the uniform random simulation, we assign random subsets of size 100. Since there are likely many reviewers with high similarity for each paper in different subsets, our algorithm again performs very well.

In Figure 3c, we show the runtime of our algorithm for the Pairwise-Constrained Problem on the various simulations, fixing $q_0 = 0.5$ and varying $n = d$ on the x-axis. The runtime of our algorithm is similar across the different simulations. Our algorithm solves the uniform random simulation case with $n = d = 5000$ in just over 10 minutes.

H Bidding Model and Additional Manipulation Experiment Results

In this section, we describe the methods used to add synthetic bids to the ICLR 2018 dataset for the experiment described in Section 6.2. We then present results from an additional experiment.

We first describe the bidding itself. Each reviewer r chooses a bid $b_{rp} \in \{-1, 0, 1\}$ for each paper p, indicating "not interested," "neutral," or "interested" respectively. Based on the similarity function used in NeurIPS 2016 [27], we compute the final similarity between reviewer r and paper p as $S'_{rp} = \gamma^{b_{rp}} S_{rp}$, where S_{rp} is the text similarity from the ICLR dataset and γ is a fixed scale parameter. For the results presented in Section 6.2, $\gamma = 2$.

We now describe the model used to generate bids for the honest (non-malicious) reviewers. We divide the reviewers uniformly at random into three groups. The first group contains 20% of the reviewers, who all bid 0 on all papers. The second group contains 50% of the reviewers, who bid non-zero on a low number of papers. These reviewers consider each paper within the 10% of papers that have highest text similarity with them, and independently choose to bid non-zero on each one with probability 0.016. If a paper is selected to bid non-zero, the bid is chosen from $\{-1, 1\}$ with uniform probability. The third group contains 30% of the reviewers, who bid non-zero on a high number of papers. They follow the same bidding procedure as the second group, but bid with probability 0.24.

Finally, we present some additional results in Figure 4. In this experiment, we set scale parameter $\gamma = 4$. The results here are qualitatively similar to the results in Section 6.2 (which set $\gamma = 2$), except that the manipulation is even more powerful here. However, our randomized assignment algorithm still limits the success rate of the manipulation to the desired level of 0.5.
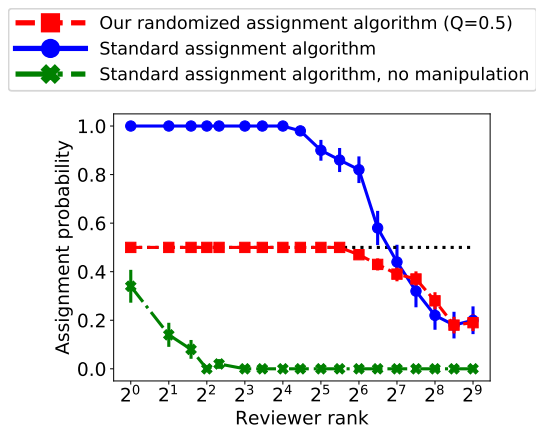
Figure 4: Additional results for the effectiveness of bidding manipulation on ICLR, $\gamma = 4$.