Strategyproofing Peer Assessment via Partitioning: The Price in Terms of Evaluators' Expertise

Komal Dhull, Steven Jecmen, Pravesh Kothari, Nihar B. Shah
Carnegie Mellon University
dhullkom@gmail.com, sjecmen@cs.cmu.edu, praveshk@andrew.cmu.edu, nihars@cs.cmu.edu

Abstract

Strategic behavior is a fundamental problem in a variety of real-world applications that require some form of peer assessment, such as peer grading of homeworks, grant proposal review, conference peer review of scientific papers, and peer assessment of employees in organizations. Since an individual's own work is in competition with the submissions they are evaluating, they may provide dishonest evaluations to increase the relative standing of their own submission. This issue is typically addressed by partitioning the individuals and assigning them to evaluate the work of only those from different subsets. Although this method ensures strategyproofness, each submission may require a different type of expertise for effective evaluation. In this paper, we focus on finding an assignment of evaluators to submissions that maximizes assigned evaluators' expertise subject to the constraint of strategyproofness. We analyze the price of strategyproofness: that is, the amount of compromise on the assigned evaluators' expertise required in order to get strategyproofness. We establish several polynomial-time algorithms for strategyproof assignment along with assignment-quality guarantees. Finally, we evaluate the methods on a dataset from conference peer review.

1 Introduction

Many applications require evaluation of certain submissions. When the number of submissions is large enough to make independent expert evaluations of all of them infeasible, the individuals who submitted are each asked to evaluate submissions made by their peers. In education, peer grading of homeworks has become increasingly prevalent in Massive Open Online Courses (MOOCs) (Shah et al. 2013; Díez Peláez et al. 2013; Piech et al. 2013) and conventional classrooms. In scientific research, peer review is used for grant proposals and conference paper submissions (Shah et al. 2017; Tomkins, Zhang, and Heavlin 2017; Shah 2021). In the workplace, peer evaluation is frequently used to assess employee performance and determine employee promotions and bonuses (Wexley and Klimoski 1984; Fiore and Souza 2021).

In many of these applications, peer assessment is *competitive*, meaning that the eventual outcome of a submission is impacted by the evaluations of other submissions. Examples include a class graded on a curve such that only a certain percentage receives an 'A' grade, a conference that intends to accept some fixed fraction of the papers, an agency awarding grants under a certain budget, or a company with a limited number of promotions on offer.

A key challenge in competitive peer assessment is that agents behave *strategically*: an agent may give low scores to the submissions they evaluate, in the hope that by hurting the chances of those submissions, they increase the relative chance of a good outcome for their own submission. A controlled experiment (Balietti, Goldstone, and Helbing 2016) found that people indeed behave in such a strategic manner in competitive peer assessment. Furthermore, the work (Thurner and Hanel 2011) shows that even a small fraction of agents behaving strategically in scientific peer review can significantly lower the average quality of the accepted papers. It is thus vital to ensure the fairness and integrity of the process by developing mechanisms to prevent such strategic behavior. In fact, the NSF briefly experimented with a method (introduced by (Merrifield and Saari 2009)) that attempts to prevent strategic behavior in the peer review of research proposals (Naghizadeh and Liu 2013), but this method does not come with theoretical guarantees.

By far the most well-studied way of ensuring strategyproofness is the partitioning method introduced in (Alon et al. 2011) and studied further in (Holzman and Moulin 2013; Bousquet, Norin, and Vetta 2014; Fischer and Klimm 2015; Aziz et al. 2016, 2019; Mattei, Turrini, and Zhydkov 2020; Kahng et al. 2018; Xu et al. 2019). Under the partitioning method, submissions are partitioned into some number of subsets, and no agent is assigned a submission from the same subset as their own. The individual agent evaluations are then aggregated separately for each subset, so that any agent's evaluations cannot influence the final outcome for their own submission.

Apart from strategyproofness, another key aspect in assigning evaluators to submissions is matching based on expertise. For instance, in peer review of papers or proposals, not all agents have expertise for all papers or proposals. Similarly, in peer assessment within an organization, the peer assessors for any employee must be chosen to have a suitable understanding of that employee's work. In peer grading of essays or projects, the assessors must have the relevant background to do a suitable evaluation. Since the goal of peer assessment is to evaluate each submission as competently as possible, it is important to ensure that each submission is assigned evaluators with suitable expertise, or in other words, to maximize the quality of the assignment of evaluators to submissions.

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

Our work contributes to the body of literature on analyzing the price of strategyproofness in various settings (Procaccia and Tennenholtz 2013; Dughmi and Ghosh 2010; Koutsoupias 2014; Ashlagi et al. 2015; Kahng et al. 2018). This includes a line of work on impartial peer nomination/selection (Alon et al. 2011; Bousquet, Norin, and Vetta 2014; Holzman and Moulin 2013; Aziz et al. 2016; Kurokawa et al. 2015; Fischer and Klimm 2015; Aziz et al. 2019; Mattei, Turrini, and Zhydkov 2020), which focuses on selecting the best k submissions in a strategyproof manner given an profile of evaluations. In contrast, we optimize the *assignment* of evaluators to submissions subject to a strategyproofness constraint and characterize the price of strategyproofness in terms of the assigned evaluators' expertise. Further, our setting generalizes the standard peer selection setting, since evaluations may be used for various relative grading schemes other than best-k selection. The prior work closest to ours is (Xu et al. 2019), which considers the partitioning mechanism specifically for conference peer review. They provide an algorithm that utilizes partitioning and conduct empirical analysis on its quality. However, their algorithm is designed to guarantee that the output ranking of submissions satisfies an efficiency property and does not focus on optimizing the evaluator assignment.

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

1. We establish fundamental limits on the amount of compromise that must be made on the assignment quality in order to impose strategyproofness via partitioning.
2. We present polynomial-time computable algorithms that are optimal in the worst case.
3. We show that the problem of instance-wise optimal strategyproof assignment via partitioning is NP-hard.
4. We conduct experimental evaluations on data from the peer-review process of the ICLR 2018 conference, where we find that our algorithms achieve high-expertise assignments while producing fair partitions of papers.

We accomplish these goals using various techniques: applying combinatorial methods, drawing a connection to equitable graph coloring, and formulating our problem as a max-cut problem.

Apart from considering strategyproofness and assignment quality together, we note two points of contrast of our work as compared to the literature. First, previous works on strategyproof partitioning consider a uniform random partition in order to ensure fairness: that is, to ensure that no partition contains disproportionately strong or disproportionately weak submissions. In our work, we analyze the random partition approach and use it as a baseline for the rest of our results. Moreover, we conduct experiments using data from the ICLR conference, which reveal that the non-random partition output by our algorithms does not result in any substantial unfairness. Second, the work (Xu et al. 2019), which deals with both assignment quality and strategyproofing, considers arbitrary authorships where each submission may have multiple authors and each agent may have authored multiple submissions. In contrast, our theoretical analysis restricts attention to each agent having authored one submission and each submission being authored by one agent. Such one-to-one authorship occurs often in peer grading, peer assessment of employees, or peer review of certain proposals, and is equivalent to common settings in the strategyproofing literature (Alon et al. 2011; Bousquet, Norin, and Vetta 2014; Holzman and Moulin 2013; Aziz et al. 2016; Kurokawa et al. 2015; Fischer and Klimm 2015; Aziz et al. 2019; Mattei, Turrini, and Zhydkov 2020; Kahng et al. 2018). In Section 5, we further provide an extension and empirical evaluation that handles arbitrary authorships.

The full version of the paper can be found online,[1] as can all of the code for our algorithms and our empirical results.[2]

2 Background and Problem Formulation

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

Preliminaries

Let $\mathcal{A} = \{a_1, \dots, a_n\}$ be the set of agents and let $\mathcal{P} = \{p_1, \dots, p_n\}$ be the set of submissions from the agents. We assume that each agent a_i ($i \in [n]$) authors exactly one submission p_i. (This is equivalent to common settings in the strategyproofing literature (Holzman and Moulin 2013; Bousquet, Norin, and Vetta 2014; Fischer and Klimm 2015; Aziz et al. 2016; Kahng et al. 2018). Furthermore, we handle arbitrary authorships in Section 5.)

A key focus of our work is the assignment of agents to submissions for review. Constructing a high-quality assignment for peer assessment (in the absence of strategyproofing requirements) is a well-studied problem, and is conducted in two phases. The first phase involves computing a "similarity" between every agent-submission pair, a number between 0 and 1 where a higher value indicates a better match in terms of expertise. Similarities are computed in various ways (Charlin and Zemel 2013; Mimno and McCallum 2007; Fiez, Shah, and Ratliff 2020; Meir et al. 2020).

[1] https://arxiv.org/abs/2201.10631

[2] https://github.com/sjecmen/optimal_strategyproof_assignment

Our work is agnostic to the method used to compute similarity scores. We assume we are given a matrix $S \in [0,1]^{n \times n}$ of 'similarity scores' for each agent-submission pair that capture the expertise of each agent to evaluate each submission. For any $i \in [n], j \in [n]$, the $(i,j)^{\text{th}}$ entry of matrix S, denoted by $s_{i,j}$, represents the similarity between agent a_i and submission p_j, where a higher value means that one expects a better quality of evaluation.

Assignments

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

The assignment is chosen by maximizing a specified objective subject to the load constraints. By far the most common choice of objective is to maximize the sum of the assigned similarities (Goldsmith and Sloan 2007; Taylor 2008; Tang, Tang, and Tan 2010; Charlin and Zemel 2013; Charlin, Zemel, and Boutilier 2011; Li and Hou 2016), and this approach is widely used in practice: for instance, in IJCAI, NeurIPS, AAAI and other conferences. Formally, for any assignment $\mathcal{M} \subseteq \mathcal{A} \times \mathcal{P}$, the total similarity is given by $\sum_{(a_i,p_j) \in \mathcal{M}} s_{i,j}$. Fixing some k, define $\mathcal{M}^*_S$ as the maximum-similarity assignment

$$\mathcal{M}^*_S = \underset{\mathcal{M} \subseteq \mathcal{A} \times \mathcal{P}}{\text{argmax}} \sum_{(a_i,p_j) \in \mathcal{M}} s_{i,j} \tag{1a}$$

$$\text{subject to} \sum_{a_i \in \mathcal{A}} \mathbb{I}[(a_i,p_j) \in \mathcal{M}] = k \quad \forall p_j \in \mathcal{P} \tag{1b}$$

$$\sum_{p_j \in \mathcal{P}} \mathbb{I}[(a_i,p_j) \in \mathcal{M}] = k \quad \forall a_i \in \mathcal{A} \tag{1c}$$

$$(a_i,p_i) \notin \mathcal{M} \qquad\qquad \forall a_i \in \mathcal{A}. \tag{1d}$$

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

While we consider the aforementioned popular objective in most of our analysis, we note that another objective that is sometimes used is the leximin or max-min fairness of the assignment (Garg et al. 2010; Stelmakh, Shah, and Singh 2019b), which we examine in Section 3.

Strategyproofness via Partitioning

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

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

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

Definition 1. *An assignment $\mathcal{M}$ is* **strategyproof-via-partitioning** *if there exists a partition of $\mathcal{A}$ into two subsets $\mathcal{A}_1, \mathcal{A}_2$ such that*

$$(a_i,p_j) \notin \mathcal{M} \qquad \forall a_i, a_j \in \mathcal{A}_t; \forall t \in \{1,2\} \tag{2a}$$

$$\mathcal{A}_1 \cup \mathcal{A}_2 = \mathcal{A}; \quad \mathcal{A}_1 \cap \mathcal{A}_2 = \emptyset. \tag{2b}$$

In Section 3, we extend this definition to allow for partitioning into more than two subsets. Our goal is to find a maximum-similarity strategyproof-via-partitioning assignment

$$\underset{\mathcal{A}_1,\mathcal{A}_2 \subseteq \mathcal{A}; \mathcal{M} \subseteq \mathcal{A} \times \mathcal{P}}{\text{argmax}} \sum_{(a_i,p_j) \in \mathcal{M}} s_{i,j}$$

$$\text{subject to (1b)} - \text{(1d), (2a), (2b).}$$

If an assignment satisfies (2a) for some partition, we say that assignment "respects" the partition; we say that a pair (a_i, p_j) respects the partition if a_i and a_j are in different subsets. Note that the load constraints imply $|\mathcal{A}_1| = |\mathcal{A}_2|$ for any feasible solution, so we assume that n is even in all of our results; we also assume that $k \leq \frac{n}{2}$ for feasibility.

Given a partition $(\mathcal{A}_1, \mathcal{A}_2)$, finding the maximum-similarity assignment can be done via standard methods by additionally disallowing any pairs violating constraint (2a). Thus, the primary question we consider in this paper is how to optimally choose the partition in order to maximize the similarity of the resulting assignment.

Evaluation Metric

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

$$\min_{S : \mathsf{Opt}_S > 0} \frac{\sum_{(a_i,p_j) \in \mathcal{M}_S} s_{i,j}}{\mathsf{Opt}_S}.$$

Algorithm 1: Random Partition

```
Input: $\mathcal{A}, \mathcal{P}, S, k$
Sample $\mathcal{A}_1$ uniformly at random from $\{\mathcal{A}' : \mathcal{A}' \subseteq \mathcal{A}, |\mathcal{A}'| = |\mathcal{A}|/2\}$
$\mathcal{A}_2 \leftarrow \mathcal{A} \setminus \mathcal{A}_1$
$\mathcal{M} \leftarrow$ max-similarity assignment with loads $k$ respecting $(\mathcal{A}_1, \mathcal{A}_2)$
return assignment $\mathcal{M}$ and partition $(\mathcal{A}_1, \mathcal{A}_2)$
```

3 Theoretical Results

In this section, we present our main theoretical results.

Baseline: Random Partitioning

We begin with a result that provides a simple baseline for comparison: Algorithm 1 chooses a partition uniformly at random. This is the approach taken by most prior literature on partitioning-based mechanisms. It is easy to show that such a uniformly random partition can attain at least half of the optimal similarity.

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

Proof. Since it is feasible to assign all pairs in $\mathcal{M}_S^*$ that respect the partition, Algorithm 1 achieves expected similarity

$$\begin{aligned}
\mathbb{E}_{\mathcal{M}}\left[\sum_{(a_i,p_j)\in\mathcal{M}} s_{i,j}\right] &\geq \sum_{(a_i,p_j)\in\mathcal{M}_S^*} s_{i,j}(\mathbb{P}[a_i \in \mathcal{A}_1, a_j \in \mathcal{A}_2] \\
&\qquad + \mathbb{P}[a_j \in \mathcal{A}_1, a_i \in \mathcal{A}_2]) \\
&= \sum_{(a_i,p_j)\in\mathcal{M}_S^*} s_{i,j}\left(\frac{n}{n-1}\right)\frac{1}{2} \\
&\geq \frac{1}{2}\mathsf{Opt}_S.
\end{aligned}$$

□

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

Worst-Case Upper Bound

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

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

Algorithm 2: Cycle-Breaking Algorithm

```
Input: Agents $\mathcal{A}$, papers $\mathcal{P}$, similarities $S$, load $k$
$\widetilde{\mathcal{M}}_S^* \leftarrow$ max-similarity assignment with loads 1
$\mathcal{A}_1 \leftarrow \emptyset$; $\mathcal{A}_2 \leftarrow \emptyset$
for cycle $\gamma$ of length $\ell$ in $\widetilde{\mathcal{M}}_S^*$ do
    $y \leftarrow \min_{i\in[\ell]} s_{\gamma_i,\gamma_{i+1}}$
    $A \leftarrow \emptyset$; $B \leftarrow \emptyset$
    for $i \in [\ell]$ do
        $j \leftarrow y + i \mod \ell$
        if $i$ odd then
            $A \leftarrow A \cup \{a_{\gamma_j}\}$
        else
            $B \leftarrow B \cup \{a_{\gamma_j}\}$
        end if
    end for
    if $|\mathcal{A}_1| \leq |\mathcal{A}_2|$ then
        $\mathcal{A}_1 \leftarrow \mathcal{A}_1 \cup A$; $\mathcal{A}_2 \leftarrow \mathcal{A}_2 \cup B$
    else
        $\mathcal{A}_1 \leftarrow \mathcal{A}_1 \cup B$; $\mathcal{A}_2 \leftarrow \mathcal{A}_2 \cup A$
    end if
end for
$\mathcal{M} \leftarrow$ max-similarity assignment with loads $k$ respecting $(\mathcal{A}_1, \mathcal{A}_2)$
return assignment $\mathcal{M}$ and partition $(\mathcal{A}_1, \mathcal{A}_2)$
```

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

Cycle-Breaking Algorithm

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

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

Algorithm 2 works by splitting each cycle in the optimal $k = 1$ assignment across the partition in the way that maximizes similarity. The following theorem shows a lower bound on the similarity of the strategyproof-via-partitioning assignment produced by this algorithm when $k = 1$.

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

Proof. $(\mathcal{A}_1, \mathcal{A}_2)$ is a partition of $\mathcal{A}$ since each agent is included in exactly one cycle in $\widetilde{\mathcal{M}}_S^*$. Further, $|\mathcal{A}_1| = |\mathcal{A}_2|$

```
Algorithm 3: Coloring Algorithm
Input: Agents $\mathcal{A}$, papers $\mathcal{P}$, similarities $S$, load $k$
 1: $\mathcal{M}^*_S \leftarrow$ max-similarity assignment with loads $k$
 2: $G_{\mathcal{M}^*} \leftarrow$ directed graph representing $\mathcal{M}^*_S$
 3: $f \leftarrow$ equitable $(2k+2)$-coloring of $G_{\mathcal{M}^*}$
 4: for $T \in \{T : T \subseteq [2k+2], |T| = k+1\}$ do
 5:     $\mathcal{A}_T \leftarrow \{a_i : v_i \in V, f(v_i) \in T\}$
 6:     $\mathcal{A}'_T \leftarrow \{a_i : v_i \in V, f(v_i) \notin T\}$
 7:     $x_T \leftarrow \sum_{a_i \in \mathcal{A}_T, a_j \in \mathcal{A}'_T} s_{i,j} \mathbb{I}[(a_i, p_j) \in \mathcal{M}^*_S]$
 8: end for
 9: $T^* = \operatorname{argmax}_T x_T$
10: $\mathcal{A}_1 \leftarrow \mathcal{A}_{T^*}$; $\mathcal{A}_2 \leftarrow \mathcal{A}'_{T^*}$
11: $\mathcal{M} \leftarrow$ max-similarity assignment with loads $k$ respecting $(\mathcal{A}_1, \mathcal{A}_2)$
12: return assignment $\mathcal{M}$ and partition $(\mathcal{A}_1, \mathcal{A}_2)$
```

since agents are added to the partition to keep it as balanced as possible and we assume n is even.

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

The partitioning step can be done in $O(n)$ time, since each agent is considered once, and finding the two maximum-similarity matchings can be done with high probability in $\widetilde{O}(n^3)$ time (van den Brand et al. 2021). □

Coloring Algorithm

In this section, we present another algorithm for strategyproof peer assessment, which meets the upper bound of Theorem 1 for any k. The algorithm begins by constructing a directed graph $G_{\mathcal{M}^*}$ representing the optimal assignment $\mathcal{M}^*$. This graph contains one vertex v_i for all $i \in [n]$, and an edge (v_i, v_j) if $(a_i, p_j) \in \mathcal{M}^*$. We then find an equitable coloring of this graph, which is defined as follows.

Definition 2. *For any $\alpha \in \mathbb{Z}_+$, an **equitable α-coloring** of a directed graph $G = (V, E)$ is a function $f : V \to [\alpha]$ such that $f(v_i) \neq f(v_j) \quad \forall (v_i, v_j) \in E$ and $|\{v : f(v) = x\}| - |\{v : f(v) = y\}| \leq 1 \quad \forall x, y \in [\alpha]$.*

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

Theorem 3. *(Hajnal and Szemerédi 1970; Kierstead et al. 2010) A graph $G = (V, E)$ with maximum degree at most Δ has an equitable $\Delta + 1$-coloring that can be found in $O(\Delta|V|^2)$ time.*

Algorithm 3 uses this result as a subroutine to find an equitable $(2k+2)$-coloring of $G_{\mathcal{M}^*}$. It then partitions the colors in the way that maximizes the total similarity of pairs in $\mathcal{M}^*$ split by the partition. The following result proves that this algorithm is worst-case optimal.

Theorem 4. *For any k and any S, if n is divisible by $2k+2$, Algorithm 3 finds a strategyproof-via-partitioning assignment with similarity at least $\frac{k+1}{2k+1}\mathsf{Opt}_S$ in polynomial time.*

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

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

Assuming k is constant, the time complexity of the partitioning step is dominated by the $O(n^2)$ time taken to find the equitable coloring. Finding the two maximum-similarity matchings can be done with high probability in $\widetilde{O}(n^3)$ time (van den Brand et al. 2021). □

The assumption that n is divisible by $2k+2$ is needed to guarantee that the partition is balanced. However, for arbitrary n, the subsets of the partition differ in size by only $k+1$ agents at most. If there are a small number of "reserve" agents who did not submit any work and are not used in $\mathcal{M}^*$, these agents can provide any evaluations needed for a feasible assignment. Since k is a small constant (often ≤ 3), having access to enough reserve agents is likely not an issue in practice. For example, in a scientific peer review setting, many extra non-author reviewers are available; in a classroom setting, an instructor could grade the extra submissions.

Hardness

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

Theorem 5. *For any $k \geq 2$, it is NP-hard to find the optimal strategyproof-via-partitioning assignment, even when similarities are binary (that is, when $S \in \{0,1\}^{n \times n}$).*

Proof Sketch. The proof is by reduction from the "Simple Max Cut on Cubic Graphs" problem (Yannakakis 1978). We construct an instance of the strategyproof-via-partitioning assignment problem where each agent corresponds to a vertex. For some orientation of the input graph, we set $s_{ij} = 1$ for each directed edge (v_i, v_j), and set similarities to zero elsewhere. These edges could all be assigned by $\mathcal{M}^*$ when $k \geq 2$, but the optimal strategyproof-via-partitioning assignment is limited to the max-cut value in the original graph. □

The complete proof is provided in Appendix A.

Algorithm 4: Multi-Partition Algorithm

```
Input: Agents $\mathcal{A}$, papers $\mathcal{P}$, similarities $S$, load $k$
1: $\mathcal{M}_S^* \leftarrow$ max-similarity assignment with loads $k$
2: $G_{\mathcal{M}^*} \leftarrow$ directed graph representing $\mathcal{M}_S^*$
3: $f \leftarrow$ equitable $(2k+1)$-coloring of $G_{\mathcal{M}^*}$
4: return assignment $\mathcal{M}_S^*$ and partition with $2k+1$ subsets
   $(\{a_j : v_j \in V, f(v_j) = i\}_{i \in [2k+1]})$
```

Partitions With More Than Two Subsets

We now relax the definition of "strategyproof-via-partitioning" given in Definition 1. Rather than requiring that agents be partitioned into two subsets, we allow them to be partitioned into any constant (i.e., not depending on n) number of subsets. This slight relaxation of our problem formulation allows us to obtain a strategyproof-via-partitioning assignment that achieves total similarity Opt_S for any S.

Theorem 6. *For any $k \geq 1$ and any S, Algorithm 4 finds a partition of agents into $2k+1$ subsets, where each subset contains either $\lfloor \frac{n}{2k+1} \rfloor$ or $\lceil \frac{n}{2k+1} \rceil$ agents, and a strategyproof-via-partitioning assignment respecting this partition in polynomial time. This assignment has total similarity Opt_S.*

Proof. Each vertex in $G_{\mathcal{M}^*}$ has in-degree and out-degree k, so the maximum (total) degree is at most $2k$. Therefore, by Theorem 3 we can find an equitable $(2k+1)$-coloring of $G_{\mathcal{M}^*}$ in $O(n^2)$ time. By Definition 2, the entirety of $\mathcal{M}_S^*$ respects the partition induced by the coloring and so is strategyproof-via-partitioning with respect to this partition. Also by Definition 2, all color classes differ in size by at most 1. □

Algorithm 4 constructs a directed graph representing $\mathcal{M}^*$ as described in Section 3. It then finds an equitable $(2k+1)$-coloring using Theorem 3 and uses this coloring as the partition.

Although we can recover the entire optimal similarity with this method, increasing the number of subsets comes at the cost of reliability in determining the post-evaluation outcomes, since all relative outcomes must be chosen independently in each subset. In Section 4, we experimentally examine this cost.

Fairness Objective

So far we have analyzed the objective of maximizing total similarity (1) due to its widespread use. However, this objective has been found to result in imbalanced or unfair assignments (Stelmakh, Shah, and Singh 2019b). An alternative proposed in the literature is to optimize the max-min fairness objective, which maximizes the total similarity assigned to the submission with minimum assigned similarity (Garg et al. 2010; Stelmakh, Shah, and Singh 2019b; Kobren, Saha, and McCallum 2019). Formally, the problem of finding the optimal strategyproof-via-partitioning assignment under this objective is:

$$\underset{\mathcal{A}_1, \mathcal{A}_2 \subseteq \mathcal{A}; \mathcal{M} \subseteq \mathcal{A} \times \mathcal{P}}{\operatorname{argmax}} \min_{p_j \in \mathcal{P}} \sum_{a_i \in \mathcal{A}} s_{i,j} \mathbb{I}[(a_i, p_j) \in \mathcal{M}]$$

$$\text{subject to (1b)} - \text{(1d), (2a), (2b).}$$

Assignment algorithms optimizing this objective have been used in venues such as ICML 2020 and implemented in conference management platforms such as OpenReview.net.

In this section, we analyze the price of strategyproofing under this max-min objective. The following result shows that unfortunately, we cannot hope to do well on this objective in the worst-case.

Theorem 7. *For any k and any $n \geq 6$, there exist similarities S on n agents such that the optimal non-strategyproof assignment has max-min objective value strictly greater than 0 while no strategyproof-via-partitioning assignment has a max-min objective value greater than 0.*

Proof. Split the agents into two groups such that both groups have an odd number of agents at least 3; this is possible since we assume n is even. Within each group $\{a_{\gamma_1}, \ldots, a_{\gamma_\ell}\}$ of size ℓ, set similarities $s_{\gamma_i, \gamma_{i+1}} = 1$ for all $i \in [\ell - 1]$ and $s_{\gamma_\ell, \gamma_1} = 1$. Set similarities to 0 elsewhere. On these similarities, the optimal non-strategyproof assignment can assign every similarity-1 pair for a max-min fairness of 1. However, since the number of reviewers in each group is odd, any partition of $\mathcal{A}$ into two subsets must place two agents $a_{\gamma_i}, a_{\gamma_{i+1}}$ (or $a_{\gamma_\ell}, a_{\gamma_1}$) from each group in the same subset. Therefore, some submission $p_{\gamma_{i+1}}$ from each group will have an assigned similarity of 0. □

4 Experimental Results

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

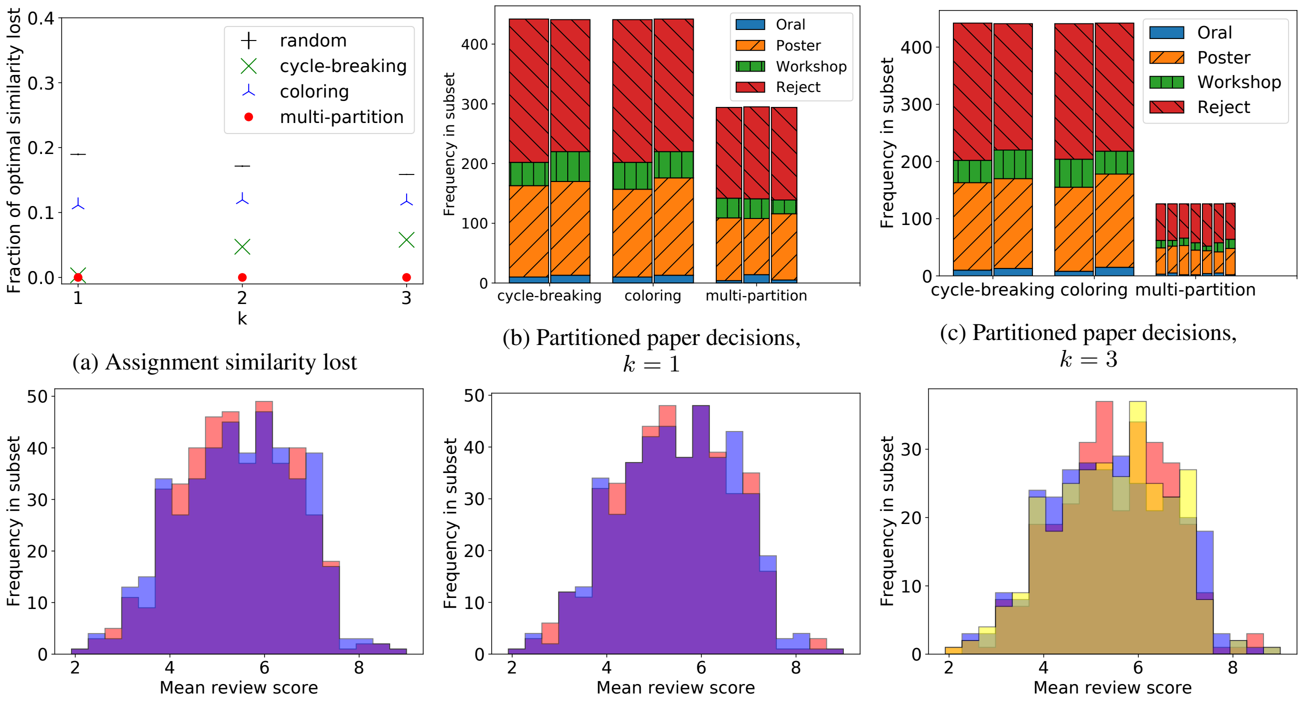
Setup

We evaluate our algorithms on data from the peer-review process at ICLR 2018. We use similarities recreated in (Xu et al. 2019). To evaluate the partition quality, we also use the actual review scores and the accept/reject decisions at the ICLR 2018 conference (He 2020).

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

We evaluate four partitioning algorithms: random partitioning (Algorithm 1), the cycle-breaking algorithm (Algorithm 2), the coloring algorithm (Algorithm 3), and the multi-partition algorithm (Algorithm 4). Since each paper received 3 reviews at ICLR 2018, we test values of $k \in \{1, 2, 3\}$.

Additional experimental results are available in Appendix B.



(a) Assignment similarity lost

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

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

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

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

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

Figure 1: Experimental results on data from ICLR 2018.

Assignment Similarity

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

Partition Quality

We next examine whether the partitions produced by these algorithms place similar-quality papers into each subset, since under the partition-based method, the final accept/reject decisions for papers are performed independently in each subset. In Figures 1b and 1c, we display the number of papers receiving each decision (oral presentation, poster presentation, invitation to workshop track, or rejection) in each subset of the partitions. For each algorithm, each bar displays the decisions for the papers in one subset of the partition. Across all algorithms and values of k, the partitions constructed have very similar numbers of papers receiving each decision in each subset. Since a very small number of papers (23 out of 883) are accepted for oral presentation overall, the relative difference in the number of oral papers between subsets is sometimes large; however, the absolute difference in the number of oral papers remains small.

Further, in Figures 1d, 1e, and 1f, we show the mean review scores given to each paper for the case of $k = 1$. In Figures 1d and 1e, the red and blue histograms correspond to the scores given to the papers in the two subsets of the algorithm's partition, with the purple section indicating their overlap; in Figure 1f, the third subset is additionally indicated in yellow. For all algorithms, the distributions of scores appear very similar across subsets of the partition. Formally, we test the difference between the score distributions of different subsets via the two-sample Kolmogorov-Smirnov test, a non-parametric test of the null hypothesis that the two samples came from the same distribution. Each sample is the set of scores given to the papers in one subset of the partition. We report the results of the test in Table 1, which contains the p-values of the test along with the effect size D, defined as the maximum difference between the empirical cdfs of

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

Table 1: Results of the Kolmogorov-Smirnov test of whether the review scores in the two partitioned subsets are drawn from the same distribution.

the two samples. For the multi-partition algorithm, we test each pair of subsets and we report results for the pair with highest D. In all cases, the p-values are high, meaning that the test cannot reject the hypothesis that the subsets were drawn from the same distribution.

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

5 Heuristic Algorithm for Arbitrary Authorship

In this section, we propose an algorithm for strategyproof-via-partitioning assignment that can accommodate arbitrary authorship of submissions, as opposed to the one-to-one authorship that we assume in our problem formulation (Section 2). This algorithm is closely based on the cycle-breaking algorithm (Algorithm 2) from Section 3. We do not have any theoretical guarantees for this algorithm, but we provide evaluations on the ICLR 2018 dataset introduced in Section 4.

Algorithm

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

Algorithm 5 works by taking a problem instance with arbitrary authorship, using it to construct a (fake) problem instance with one-to-one authorship, and running Algorithm 2 on this fake instance to find a partition. Each agent in the fake instance corresponds to a connected component of the authorship graph $\mathcal{U}$. Similarities between fake agents are set equal to the total similarity of pairs in the optimal non-strategyproof assignment that are split between the respective components. After construction, we pass this fake instance into Algorithm 2.

We slightly modify Algorithm 2 to encourage more balanced partitions in this setting before calling it in Line 7. In Lines 14-18 of Algorithm 2, we take the larger of A and B and add it to the smaller of $\mathcal{A}_1$ and $\mathcal{A}_2$ as measured by the total number of papers in the connected components represented within each set. In addition, we iterate through vertices (when finding cycles) in the order of largest connected component to smallest, where size is again determined by the number of papers in each component.

Experimental Results

We test Algorithm 5 on the ICLR 2018 dataset, using the full authorship graph from the conference. Following the suggestion in (Xu et al. 2019), we also try running Algorithm 5 after removing reviewers with a large number of authored papers; this breaks up large connected components in the authorship graph, thus allowing more flexibility in choosing a partition. Specifically, we remove the 53 reviewers with more than 3 papers authored (2.2% of reviewers) from the reviewer pool. As a baseline for comparison, we also test 100 trials of random partitioning, which chooses half of the connected components at random for each subset. We set loads of $k_p = 3$ and $k_a = 6$, since these are standard conference loads (Xu et al. 2019).

First, we see in Figure 2a that Algorithm 5 outperforms random partitioning in terms of similarity. Our algorithm loses 11.7% of the non-strategyproof optimal similarity, whereas the random partitioning loses 16.8% of optimal on average. When we remove high-authorship reviewers before running Algorithm 5, it only loses 8.9% of the optimal similarity (which is still allowed to use all reviewers).

Finally, we examine the partition quality in a similar manner as in Section 4. In Figure 2b, we plot the proportion of papers within each subset of the partitions produced by Algorithm 5 that received each decision. We see that the subsets have similar proportions of papers receiving each decision, regardless of whether we remove high-authorship reviewers. However, removing these reviewers results in a significantly more balanced partition: the number of papers differs between subsets by 109 when high-authorship reviewers are not removed and by only 1 when they are. In Figure 2c, we see that the two subsets also have similar distributions of paper scores when high-authorship reviewers are removed.

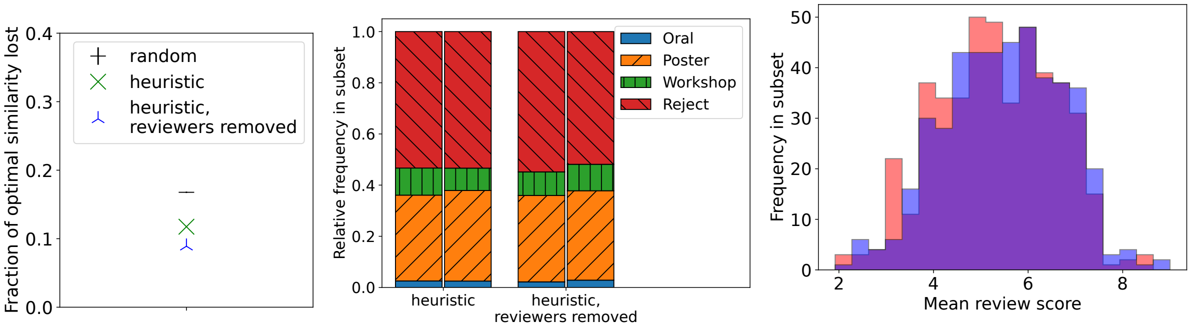
Our results are highly comparable to those of (Xu et al. 2019), who provide a partitioning algorithm that simply returns an arbitrary feasible partition of the connected components of the authorship graph. The authors report that this algorithm loses only 11.4% of the optimal similarity on the ICLR 2018 data with the same loads, a similar performance to our algorithm's despite the fact that our algorithm more carefully chooses the partition. This phenomenon may be related to the results of (Jecmen et al. 2022), who find that randomly splitting reviewers into two "phases" of reviewing does not significantly degrade assignment quality on real conference datasets.

6 Discussion

We jointly considered two key aspects of the peer-assessment process—strategyproofing and assignment

Algorithm 5: Heuristic Algorithm for Arbitrary Authorship

```
Input: agents $\mathcal{A}$, papers $\mathcal{P}$, similarities $S$, authorship graph $\mathcal{U}$, paper load $k_p$, maximum agent load $k_a$
$\overline{\mathcal{M}}^* \leftarrow$ max-similarity assignment with loads $(k_a, k_p)$
$\{V_1, \dots, V_N\} \leftarrow$ vertices of the connected components of $\mathcal{U}$
$\mathcal{A}' \leftarrow \{a'_i : i \in [N]\}$; $\mathcal{P}' \leftarrow \{p'_i : i \in [N]\}$
for $i, j \in [N]$ do
    $s'_{ij} \leftarrow \sum_{a_a \in V_i, p_b \in V_j} s_{ab} \mathbb{I}[(a_a, p_b) \in \overline{\mathcal{M}}^*] + \sum_{a_a \in V_j, p_b \in V_i} s_{ab} \mathbb{I}[(a_a, p_b) \in \overline{\mathcal{M}}^*]$
end for
$\mathcal{M}', (\mathcal{A}'_1, \mathcal{A}'_2) \leftarrow$ output of Algorithm 2 on input $(\mathcal{A}', \mathcal{P}', S', k' = 1)$
$\mathcal{T}_1 \leftarrow \bigcup_{i : a'_i \in \mathcal{A}'_1} V_i$; $\mathcal{T}_2 \leftarrow \bigcup_{i : a'_i \in \mathcal{A}'_2} V_i$
$\mathcal{M} \leftarrow$ max-similarity assignment with loads $(k_a, k_p)$ respecting $(\mathcal{T}_1, \mathcal{T}_2)$
return assignment $\mathcal{M}$ and partition $(\mathcal{T}_1, \mathcal{T}_2)$
```



(a) Assignment similarity lost

(b) Partitioned paper decisions

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

Figure 2: Experimental results using Algorithm 5 on the authorship from ICLR 2018.

quality—and derived fundamental limits as well as designed computationally-efficient algorithms that achieve these limits. Our theoretical and empirical contributions lead to several directions of future work.

A first key direction of future work is to extend these theoretical results to arbitrary authorship graphs, as in conference peer review. We present a heuristic algorithm with an empirical evaluation in Section 5, but the problem of establishing fundamental limits and optimal algorithms is open. Second, most of our work considered worst-case guarantees, while showing that it is NP-hard to attain instance-wise optimality. However, our experimental results showed that our algorithms perform much better than worst-case on real-world instances. This suggests a theoretically interesting and practically useful direction of future work: designing algorithms with approximately-optimal instance-wise guarantees. Third, in contrast to past work, our partitions are non-random. Building on our experimental results revealing that these non-random partitions still result in subsets with roughly equal submission strengths, future work could dig deeper into this phenomenon both theoretically and empirically. Fourth, recent work (Mattei, Turrini, and Zhydkov 2020) provides a strategyproof algorithm with theoretical guarantees that does not rely on partitioning. Even though partitioning is by far the dominant way of strategyproofing, it is of interest to extend our results to such strategyproofing methods that may not employ partitioning. Finally, there are various other types of strategic or dishonest behavior in peer assessment (Stelmakh, Shah, and Singh 2021; Hvistendahl 2013; Ferguson, Marcus, and Oransky 2014; Fanelli 2009; Resnik, Gutierrez-Ford, and Peddada 2008; Vijaykumar 2020; Littman 2021; Jecmen et al. 2020; Wu et al. 2021) and the design of computational methods to mitigate such behavior is vital. More generally, peer assessment is an important application with a broad set of challenges including subjectivity (Lee 2015; Noothigattu, Shah, and Procaccia 2021), miscalibration (Roos, Rothe, and Scheuermann 2011; Wang and Shah 2019), biases (Tomkins, Zhang, and Heavlin 2017; Stelmakh, Shah, and Singh 2019a; Manzoor and Shah 2021), and others (Meir et al. 2020; Fiez, Shah, and Ratliff 2020; Stelmakh et al. 2021; Wang et al. 2021; Shah 2021).

Acknowledgments

This work was supported by NSF CAREER award 1942124 and a Google Research Scholar Award. We thank Ariel Procaccia for helpful discussions.

References

Alon, N.; Fischer, F.; Procaccia, A.; and Tennenholtz, M. 2011. Sum of us: Strategyproof selection from the selectors. In *Conference on Theoretical Aspects of Rationality and Knowledge*, 101–110. ACM.

Ashlagi, I.; Fischer, F.; Kash, I. A.; and Procaccia, A. D. 2015. Mix and match: A strategyproof mechanism for multi-hospital kidney exchange. *Games and Economic Behavior*, 91: 284–296.

Aziz, H.; Lev, O.; Mattei, N.; Rosenschein, J. S.; and Walsh, T. 2016. Strategyproof Peer Selection: Mechanisms, Analyses, and Experiments. In *AAAI*, 397–403.

Aziz, H.; Lev, O.; Mattei, N.; Rosenschein, J. S.; and Walsh, T. 2019. Strategyproof peer selection using randomization, partitioning, and apportionment. *Artificial Intelligence*.

Balietti, S.; Goldstone, R. L.; and Helbing, D. 2016. Peer review and competition in the Art Exhibition Game. *Proceedings of the National Academy of Sciences*, 113(30): 8414–8419.

Bousquet, N.; Norin, S.; and Vetta, A. 2014. A near-optimal mechanism for impartial selection. In *International Conference on Web and Internet Economics*, 133–146. Springer.

Charlin, L.; Zemel, R.; and Boutilier, C. 2011. A framework for optimizing paper matching. In *UAI*, 86–95.

Charlin, L.; and Zemel, R. S. 2013. The Toronto Paper Matching System: An automated paper-reviewer assignment system. In *ICML Workshop on Peer Reviewing and Publishing Models*.

Díez Peláez, J.; Luaces Rodríguez, Ó.; Alonso Betanzos, A.; Troncoso, A.; and Bahamonde Rionda, A. 2013. Peer assessment in MOOCs using preference learning via matrix factorization. In *NIPS Workshop on Data Driven Education*.

Dughmi, S.; and Ghosh, A. 2010. Truthful assignment without money. In *Proceedings of the 11th ACM conference on Electronic commerce*, 325–334.

Fanelli, D. 2009. How many scientists fabricate and falsify research? A systematic review and meta-analysis of survey data. *PLOS One*, 4(5): e5738.

Ferguson, C.; Marcus, A.; and Oransky, I. 2014. Publishing: The peer-review scam. *Nature News*, 515(7528): 480.

Fiez, T.; Shah, N.; and Ratliff, L. 2020. A SUPER* algorithm to optimize paper bidding in peer review. In *UAI*, 580–589.

Fiore, A. D.; and Souza, M. 2021. Are Peer Reviews the Future of Performance Evaluations? *Harvard Business Review*.

Fischer, F.; and Klimm, M. 2015. Optimal impartial selection. *SIAM Journal on Computing*, 44(5): 1263–1285.

Garg, N.; Kavitha, T.; Kumar, A.; Mehlhorn, K.; and Mestre, J. 2010. Assigning papers to referees. *Algorithmica*, 58(1): 119–136.

Goldsmith, J.; and Sloan, R. H. 2007. The AI conference paper assignment problem. In *AAAI Workshop on Preference Handling for Artificial Intelligence*, 53–57.

Hajnal, A.; and Szemerédi, E. 1970. Proof of a conjecture of P. Erdos. *Combinatorial Theory and its Applications*, 2: 601–623.

He, H. 2020. OpenReview Explorer. https://github.com/Chillee/OpenReviewExplorer (accessed May 26, 2021).

Holzman, R.; and Moulin, H. 2013. Impartial nominations for a prize. *Econometrica*, 81(1): 173–196.

Hvistendahl, M. 2013. China's Publication Bazaar. *Science*, 342(6162): 1035–1039.

Jecmen, S.; Zhang, H.; Liu, R.; Fang, F.; Conitzer, V.; and Shah, N. B. 2022. Near-Optimal Reviewer Splitting in Two-Phase Paper Reviewing and Conference Experiment Design. In Faliszewski, P.; Mascardi, V.; Pelachaud, C.; and Taylor, M. E., eds., *21st International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2022, Auckland, New Zealand, May 9-13, 2022*, 1642–1644. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS).

Jecmen, S.; Zhang, H.; Liu, R.; Shah, N. B.; Conitzer, V.; and Fang, F. 2020. Mitigating Manipulation in Peer Review via Randomized Reviewer Assignments. In *NeurIPS*.

Kahng, A.; Kotturi, Y.; Kulkarni, C.; Kurokawa, D.; and Procaccia, A. D. 2018. Ranking wily people who rank each other. In *AAAI*.

Kierstead, H. A.; Kostochka, A. V.; Mydlarz, M.; and Szemerédi, E. 2010. A fast algorithm for equitable coloring. *Combinatorica*, 30(2): 217–224.

Kobren, A.; Saha, B.; and McCallum, A. 2019. Paper Matching with Local Fairness Constraints. In *ACM KDD*.

Koutsoupias, E. 2014. Scheduling without payments. *Theory of Computing Systems*, 54(3): 375–387.

Kurokawa, D.; Lev, O.; Morgenstern, J.; and Procaccia, A. D. 2015. Impartial Peer Review. In *IJCAI*, 582–588.

Lee, C. J. 2015. Commensuration bias in peer review. *Philosophy of Science*, 82(5): 1272–1283.

Li, B.; and Hou, Y. T. 2016. The new automated IEEE INFOCOM review assignment system. *IEEE Network*, 30(5): 18–24.

Littman, M. L. 2021. Collusion rings threaten the integrity of computer science research. *Communications of the ACM*, 64(6): 43–44.

Manzoor, E.; and Shah, N. B. 2021. Uncovering Latent Biases in Text: Method and Application to Peer Review. In *AAAI*.

Mattei, N.; Turrini, P.; and Zhydkov, S. 2020. PeerNomination: Relaxing Exactness for Increased Accuracy in Peer Selection. In *IJCAI*, 393–399.

Meir, R.; Lang, J.; Lesca, J.; Kaminsky, N.; and Mattei, N. 2020. A market-inspired bidding scheme for peer review paper assignment. In *Games, Agents, and Incentives Workshop at AAMAS*.

Merrifield, M. R.; and Saari, D. G. 2009. Telescope time without tears: A distributed approach to peer review. *Astronomy & Geophysics*, 50(4): 4–16.

Mimno, D.; and McCallum, A. 2007. Expertise Modeling for Matching Papers with Reviewers. In *ACM SIGKDD*, 500––509. ISBN 9781595936097.

Naghizadeh, P.; and Liu, M. 2013. Incentives, Quality, and Risks: A Look Into the NSF Proposal Review Pilot. *arXiv:1307.6528*.

Noothigattu, R.; Shah, N.; and Procaccia, A. 2021. Loss Functions, Axioms, and Peer Review. *Journal of Artificial Intelligence Research*.

Piech, C.; Huang, J.; Chen, Z.; Do, C.; Ng, A.; and Koller, D. 2013. Tuned models of peer assessment in MOOCs. *arXiv:1307.2579*.

Procaccia, A. D.; and Tennenholtz, M. 2013. Approximate mechanism design without money. *ACM Transactions on Economics and Computation (TEAC)*, 1(4): 1–26.

Resnik, D. B.; Gutierrez-Ford, C.; and Peddada, S. 2008. Perceptions of ethical problems with scientific journal peer review: An exploratory study. *Science and Engineering Ethics*, 14(3): 305–310.

Roos, M.; Rothe, J.; and Scheuermann, B. 2011. How to Calibrate the Scores of Biased Reviewers by Quadratic Programming. In *AAAI*.

Shah, N. B. 2021. An Overview of Challenges, Experiments, and Computational Solutions in Peer Review. Preprint http://bit.ly/PeerReviewOverview (accessed Jan 25, 2022); To appear in CACM.

Shah, N. B.; Bradley, J. K.; Parekh, A.; Wainwright, M.; and Ramchandran, K. 2013. A case for ordinal peer-evaluation in MOOCs. In *NIPS Workshop on Data Driven Education*.

Shah, N. B.; Tabibian, B.; Muandet, K.; Guyon, I.; and von Luxburg, U. 2017. Design and Analysis of the NIPS 2016 Review Process. *arXiv:1708.09794*.

Stelmakh, I.; Shah, N.; and Singh, A. 2019a. On Testing for Biases in Peer Review. In *NeurIPS*.

Stelmakh, I.; Shah, N.; and Singh, A. 2021. Catch Me if I Can: Detecting Strategic Behaviour in Peer Assessment. In *AAAI*.

Stelmakh, I.; Shah, N.; Singh, A.; and Daumé III, H. 2021. Prior and Prejudice: The Novice Reviewers' Bias against Resubmissions in Conference Peer Review. In *CSCW*.

Stelmakh, I.; Shah, N. B.; and Singh, A. 2019b. PeerReview4All: Fair and Accurate Reviewer Assignment in Peer Review. In *Algorithmic Learning Theory*.

Tang, W.; Tang, J.; and Tan, C. 2010. Expertise Matching via Constraint-Based Optimization. In *International Conference on Web Intelligence and Intelligent Agent Technology*, volume 1, 34–41.

Taylor, C. J. 2008. On the optimal assignment of conference papers to reviewers. Forthcoming.

Thurner, S.; and Hanel, R. 2011. Peer-review in a world with rational scientists: Toward selection of the average. *The European Physical Journal B*, 84(4): 707–711.

Tomkins, A.; Zhang, M.; and Heavlin, W. D. 2017. Reviewer bias in single-versus double-blind peer review. *Proceedings of the National Academy of Sciences*, 114(48): 12708–12713.

van den Brand, J.; Lee, Y. T.; Liu, Y. P.; Saranurak, T.; Sidford, A.; Song, Z.; and Wang, D. 2021. Minimum Cost Flows, MDPs, and ℓ_1-Regression in Nearly Linear Time for Dense Instances. *arXiv:2101.05719*.

Vijaykumar, T. N. 2020. Potential Organized Fraud in ACM/IEEE Computer Architecture Conferences. https://medium.com/@tnvijayk/potential-organized-fraud-in-acm-ieee-computer-architecture-conferences-ccd61169370d (accessed Jan 25, 2022).

Wang, J.; and Shah, N. B. 2019. Your 2 is My 1, Your 3 is My 9: Handling Arbitrary Miscalibrations in Ratings. In *AAMAS*.

Wang, J.; Stelmakh, I.; Wei, Y.; and Shah, N. 2021. Debiasing Evaluations that are Biased by Evaluations. In *AAAI*.

Wexley, K.; and Klimoski, R. 1984. Performance appraisal: An update. *Research in Personnel and Human Resources Management*, 2: 35–79.

Wu, R.; Guo, C.; Wu, F.; Kidambi, R.; van der Maaten, L.; and Weinberger, K. 2021. Making Paper Reviewing Robust to Bid Manipulation Attacks. *arXiv:2102.06020*.

Xu, Y.; Zhao, H.; Shi, X.; and Shah, N. B. 2019. On Strategyproof Conference Peer Review. In *IJCAI*.

Yannakakis, M. 1978. Node-and edge-deletion NP-complete problems. In *ACM Symposium on Theory of Computing*, 253–264.