Journal of Machine Learning Research 22 (2021) 1-66 Submitted 2/20; Revised 6/21; Published 7/21

PeerReview4All: Fair and Accurate Reviewer Assignment in Peer Review

Ivan Stelmakh STIV@CS.CMU.EDU
Nihar Shah NIHARS@CS.CMU.EDU
Aarti Singh AARTI@CS.CMU.EDU
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA 15213

Editor: Moritz Hardt

Abstract

We consider the problem of automated assignment of papers to reviewers in conference peer review, with a focus on fairness and statistical accuracy. Our fairness objective is to maximize the review quality of the most disadvantaged paper, in contrast to the commonly used objective of maximizing the total quality over all papers. We design an assignment algorithm based on an incremental max-flow procedure that we prove is near-optimally fair. Our statistical accuracy objective is to ensure correct recovery of the papers that should be accepted. We provide a sharp minimax analysis of the accuracy of the peer-review process for a popular objective-score model as well as for a novel subjective-score model that we propose in the paper. Our analysis proves that our proposed assignment algorithm also leads to a near-optimal statistical accuracy. Finally, we design a novel experiment that allows for an objective comparison of various assignment algorithms, and overcomes the inherent difficulty posed by the absence of a ground truth in experiments on peer-review. The results of this experiment as well as of other experiments on synthetic and real data corroborate the theoretical guarantees of our algorithm.

Keywords: fairness, accuracy, top k recovery, assignment problem, peer review

1. Introduction

Peer review is the backbone of academia. In order to provide high-quality peer reviews, it is of utmost importance to assign papers to the right reviewers (Thurner and Hanel, 2011; Black et al., 1998; Bianchi and Squazzoni, 2015). Even a small fraction of incorrect reviews can have significant adverse effects on the quality of the published scientific standard (Thurner and Hanel, 2011) and dominate the benefits yielded by the peer-review process that may have high standards otherwise (Squazzoni and Gandelli, 2012). Indeed, researchers unhappy with the peer review process are somewhat more likely to link their objections to the quality or choice of reviewers (Travis and Collins, 1991).

We focus on peer-review in conferences where a number of papers are submitted at once. These papers must simultaneously be assigned to multiple reviewers who have load constraints. The importance of the reviewer-assignment stage of the peer-review process cannot be overestimated; quoting Rodriguez et al. (2007):

> *"one of the first and potentially most important stage is the one that attempts to distribute submitted manuscripts to competent referees."*

Given the massive scale of many conferences such as NeurIPS and ICML, these reviewer assignments are largely performed in an automated manner. For instance, NeurIPS 2016 assigned 5 out of 6 reviewers per paper using an automated process (Shah et al., 2018). This problem of automated reviewer assignments forms the focus of this paper.

Various past studies show that small changes in peer review quality can have far reaching consequences (Thorngate and Chowdhury, 2014; Squazzoni and Gandelli, 2012) not just for the papers under consideration but more generally also for the career trajectories of the researchers. These long term effects arise due to the widespread prevalence of the Matthew effect ("rich get richer") in academia (Merton, 1968).

It is also known (Travis and Collins, 1991; Lamont, 2009) that works that are novel or not mainstream, particularly those interdisciplinary in nature, face significantly higher difficulty in gaining acceptance. A primary reason for this undesirable state of affairs is the absence of sufficiently many good "peers" to aptly review interdisciplinary research (Porter and Rossini, 1985).

These issues strongly motivate the dual goals of the reviewer assignment procedure we consider in this paper — fairness and accuracy. By fairness, we specifically consider the notion of max-min fairness which is studied in various branches of science and engineering (Rawls, 1971; Lenstra et al., 1990; Hahne, 1991; Lavi et al., 2003; Bonald et al., 2006; Asadpour and Saberi, 2010). In our context of reviewer assignments, max-min fairness posits maximizing the review-quality of the paper with the least qualified reviewers. The max-min fair assignment guarantees that no paper is discriminated against in favor of more lucky counterparts. That is, even the most ambivalent paper with a small number of reviewers being competent enough to evaluate its merits will receive as good treatment as possible. The max-min fair assignment also ensures that in *any other assignment* there exists at least one paper with the fate at least as bad as the fate of the most disadvantaged paper in the aforementioned fair assignment.

Alongside, we also consider the requirement of statistical accuracy. One of the main goals of the conference peer-review process is to select the set of "top" papers for acceptance. Two key challenges towards this goal are to handle the noise in the reviews and subjective opinions of the reviewers; we accommodate these aspects in terms of existing (Ge et al., 2013; McGlohon et al., 2010; Dai et al., 2012) and novel statistical models of reviewer behavior. Prior works on the reviewer assignment problem (Long et al., 2013; Garg et al., 2010; Karimzadehgan et al., 2008; Tang et al., 2010) offer a variety of algorithms that optimize the assignment for certain deterministic objectives, but do not study their assignments from the lens of statistical accuracy. In contrast, our goal is to design an assignment algorithm that can simultaneously achieve both the desired objectives of fairness and statistical accuracy.

We make several contributions towards this problem. We first present a novel algorithm, which we call PeerReview4All, for assigning reviewers to papers. Our algorithm is based on a construction of multiple candidate assignments, each of which is obtained via an incremental execution of max-flow algorithm on a carefully designed flow network. These assignments cater to different structural properties of the similarities and a judicious choice between them provides the algorithm appealing properties.

Our second contribution is an analysis of the fairness objective that our PeerReview4All algorithm can achieve. We show that our algorithm is optimal, up to a constant factor, in terms of the max-min fairness objective. Furthermore, our algorithm can adapt to the underlying structure of the given similarity data between reviewers and papers and in various cases yield better guarantees including the exact optimal solution in certain scenarios. Finally, after optimizing the outcome for the most worst-off paper and fixing the assignment for that paper, our algorithm aims at finding the most fair assignment for the next worst-off paper and proceeds in this manner until the assignment for each paper is fixed.

As a third contribution, we show that our PeerReview4All algorithm results in strong statistical guarantees in terms of correctly identifying the top papers that should be accepted. We consider a popular statistical model (Ge et al., 2013; McGlohon et al., 2010; Dai et al., 2012) which assumes existence of some true objective score for every paper. We provide a sharp analysis of the minimax risk in terms of "incorrect" accept/reject decisions, and show that our PeerReview4All algorithm leads to a near-optimal solution.

Fourth, noting that paper evaluations are typically subjective (Kerr et al., 1977; Mahoney, 1977; Ernst and Resch, 1994; Bakanic et al., 1987; Lamont, 2009), we propose a novel statistical model capturing subjective opinions of reviewers, which may be of independent interest. We provide a sharp minimax analysis under this subjective setting and prove that our assignment algorithm PeerReview4All is also near-optimal for this subjective-score setting.

Our fifth and final contribution comprises empirical evaluations. We designed and conducted an experiment on the Amazon Mechanical Turk crowdsourcing platform to objectively compare the performance of different reviewer-assignment algorithms. The design of the experiment is done carefully to circumvent the challenge posed by the absence of a ground truth in peer review settings, so that we can evaluate accuracy objectively. In addition to the MTurk experiment, we provide an extensive evaluation of our algorithm on synthetic data, provide an evaluation on a reconstructed similarity matrix from the ICLR 2018 conference, and report the results of the experiment on real conference data conducted by Kobren et al. (2019). The results of these experiments highlight the promise of PeerReview4All in practice, in addition to the theoretical benefits discussed elsewhere in the paper. The data set pertaining to the MTurk experiment, as well as the code for our PeerReview4All algorithm, are available on the first author's website.

The remainder of this paper is organized as follows. We discuss related literature in Section 2. In Section 3, we present the problem setting formally with a focus on the objective of fairness. In Section 4 we present our PeerReview4All algorithm. We establish deterministic approximation guarantees on the fairness of our PeerReview4All algorithm in Section 5. We analyze the accuracy of our PeerReview4All algorithm under an objective-score model in Section 6, and introduce and analyze a subjective score model in Section 7. We empirically evaluate the algorithm in Section 8 using synthetic and real-world experiments. We then provide the proofs of all the results in Section 9. We conclude the paper with a discussion in Section 10.

2. Related Literature

The reviewer assignment process consists of two steps. First, a "similarity" between every (paper, reviewer) pair that captures the competence of the reviewer for that paper is computed. These similarities are computed based on various factors such as the text of the submitted paper, previous papers authored by reviewers, reviewers' bids and other features. Second, given the notion of good assignment, specified by the program chairs, papers are allocated to reviewers, subject to constraints on paper/reviewer loads. This work focuses on the second step (assignment), assuming the first step of computing similarities as a black box. In this section, we give a brief overview of the past literature on both of the steps of the reviewer-assignment process.

Computing similarities. The problem of identifying similarities between papers and reviewers is well-studied in data mining community. For example, Mimno and McCallum (2007) introduce a novel topic model to predict reviewers' expertise. Liu et al. (2014) use the random walk with restarts model to incorporate both expertise of reviewers and their authority in the final similarities. Co-authorship graphs (Rodriguez and Bollen, 2008) and more general bibliographic graph-based data models (Tran et al., 2017) give appealing methods which do not require a set of reviewers to be pre-determined by conference chair. Instead, these methods recommend reviewers to be recruited, which might be particularly useful for journal editors.

One of the most widely used automated assignment algorithms today is the Toronto Paper Matching System or TPMS (Charlin and Zemel, 2013) which also computes estimations of similarities between submitted papers and available reviewers using techniques in natural language processing. These scores might be enhanced with reviewers' self-accessed expertise adaptively queried from them in an automatic manner.

Our work uses these similarities as an input for our assignment algorithm, and considers the computation of these similarity values as a given black box.

Cumulative goal functions. With the given similarities, much of past work on reviewer assignments develop algorithms to maximize the cumulative similarity, that is, the sum of the similarities across all assigned reviewers and all papers. Such an objective is pursued by the organizers of SIGKDD conference (Flach et al., 2010) and by the widely employed TPMS assignment algorithm (Charlin and Zemel, 2013). Various other popular conference management systems such as EasyChair (`easychair.org`) and HotCRP (`hotcrp.com`) and several other papers (see Long et al. 2013; Charlin et al. 2012; Goldsmith and Sloan 2007; Tang et al. 2010 and references therein) also aim to maximize various cumulative functionals in their automated reviewer assignment procedures. In what follows, we argue however that optimizing such cumulative objectives is not fair — in order to maximize them, these algorithms may discriminate against some subset of papers. Moreover, it is the non-mainstream submissions that are most likely to be discriminated against. With this motivation, we consider a notion of fairness instead.

Fairness. In order to ensure that no papers are discriminated against, we aim at finding a *fair assignment* — an assignment that ensures that the most disadvantaged paper gets as competent reviewers as possible. The issue of fairness is partially tackled by Hartvigsen et al. (1999), where they necessitate every paper to have at least one reviewer with expertise higher than certain threshold, and then maximize the value of that threshold. However,

this improvement only partially solves the issue of discrimination of some papers: having assigned one strong reviewer to each paper, the algorithm may still discriminate against some papers while assigning remaining reviewers. Given that nowadays large conferences such as NeurIPS and ICML assign 4-6 reviewers to each paper, a careful assessment of the paper by one strong reviewer might be lost in the noise induced by the remaining weak reviews. In the present study, we measure the quality of assignment with respect to any particular paper as sum similarity over reviewers assigned to that paper. Thus, the fairness of assignment is the minimum sum similarity across all papers; we call an assignment fair if it maximizes the fairness. We note that assignment computed by our PeerReview4All algorithm is guaranteed to have *at least as large* max-min fairness as that proposed by Hartvigsen et al. (1999).

Benferhat and Lang (2001) discuss different approaches to selection of the "optimal" reviewer assignment. Together with considering a cumulative objective, they also note that one may define the optimal assignment as an assignment that minimizes a disutility of the most disadvantaged reviewer (paper). This approach resembles the notion of max-min fairness we study in this paper, but Benferhat and Lang (2001) do not propose any algorithm for computing the fair assignment.

The notion of max-min fairness was formally studied in context of peer-review by Garg et al. (2010). While studying a similar objective, our work develops both conceptual and theoretical novelties which we highlight here. First, Garg et al. (2010) measure the fairness in terms of reviewers' bids — for every reviewer they compute a value of papers assigned to that reviewer based on her/his bids and maximize the minimum value across all reviewers. While satisfying reviewers is a useful practice, we consider fairness towards the papers in their review to be of utmost importance. During a bidding process reviewers have limited time resources and/or limited access to papers' content to evaluate their relevance, and hence reviewers' bids alone are not a good proxy towards the measure of fairness. In contrast, in this work we consider similarities — scores that are designed to represent a competence of reviewer in assessing a paper. Besides reviewers' bids, similarities are computed based on the full text of the submissions and papers authored by reviewer and can additionally incorporate various factors such as quality of previous reviews, experience of reviewer and other features that cannot be self-assessed by reviewers.

The assignment algorithm proposed in Garg et al. (2010) works in two steps. In the first step, the problem is set up as an integer programming problem and a linear programming relaxation is solved. The second step involves a carefully designed rounding procedure that returns a valid assignment. The algorithm is guaranteed to recover an assignment whose fairness is within a certain additive factor from the best possible assignment. However, the fairness guarantees provided in Garg et al. (2010) turn out to be vacuous for various similarity matrices. As we discuss later in the paper, this is a drawback of the algorithm itself and not an artifact of their guarantees. In contrast, we design an algorithm with multiplicative approximation factor that is guaranteed to always provide a non-trivial approximation which is at most constant factor away from the optimal.

Next, Garg et al. (2010) consider fairness of the assignment as an eventual metric of the assignment quality. However, we note that the main goal of the conference paper reviewing process is an accurate acceptance of the best papers. Thus, in the present work we both

theoretically and empirically study the impact of the fairness of the assignment on the quality of the acceptance procedure.

Finally, although Garg et al. (2010) present their algorithm for the case of discrete reviewer's bids, we note that this assumption can be relaxed to allow real-valued similarities with a continuous range as in our setting. In this paper we refer to the corresponding extension of their algorithm as the Integer Linear Programming Relaxation (ILPR) algorithm.

Fair division. A direction of research that is relevant to our work studies the problem of fair division where max-min fairness is extensively developed. The seminal work of Lenstra et al. (1990) provides a constant factor approximation to the minimum makespan scheduling problem where the goal is to assign a number of jobs to the unrelated parallel machines such that the maximal running time is minimized. Recently Asadpour and Saberi (2010); Bansal and Sviridenko (2006) proposed approximation algorithms for the problem of assigning a number of indivisible goods to several people such that the least happy person is as happy as possible. However, we note that techniques developed in these papers cannot be directly applied for reviewer assignments problem in peer review due to the various idiosyncratic constraints of this problem. In contrast to the classical formulation studied in these works, our problem setting requires each paper to be reviewed by a fixed number of reviewers and additionally has constraints on reviewers' loads. Such constraints allow us to achieve an approximation guarantee that is independent of the total number of papers and reviewers, and depends only on λ, the number of reviewers required per paper, as $\frac{1}{\lambda}$. In contrast, the approximation factor of Asadpour and Saberi (2010) gets worse at a rate of $\frac{1}{\sqrt{m}\log^3 m}$, where m is a number of persons (papers in our setting).

Statistical aspects. Different statistical aspects related to conference peer-review have been studied in the literature. McGlohon et al. (2010) and Dai et al. (2012) studied aggregation of consumers ratings to generate a ranking of restaurants or merchants. They come up with objective score model of reviewer which we also use in this work. Ge et al. (2013) also use a similar model of reviewer and propose a Bayesian approach to calibrating reviewer' scores, which allows incorporating different biases in context of conference peer-review. Sajjadi et al. (2016) empirically compare different methods of score aggregation for peer grading of homeworks. Peer grading is a related problem to conference peer review, with the key difference that the questions and answers ("papers") are more closed-ended and objective. They conclude that although more sophisticated methods are praised in the literature, the simple averaging algorithm demonstrates better performance in their experiment. Another interesting observation they make is an edge of cardinal grades over ordinal in their setup. In this work we also consider the conferences with cardinal grading scheme of submissions.

To the best of our knowledge, no prior works on conference peer-review has studied the entire pipeline — from assignment to acceptance — from a statistical point of view. In this work we take the first steps to close this gap and provide a strong minimax analysis of naïve yet interesting procedure of determining top k papers. Our findings suggest that higher fairness of the assignment leads to better quality of acceptance procedure. We consider both the objective score model (Ge et al., 2013; McGlohon et al., 2010; Dai et al., 2012) and a novel subjective-score model that we propose in the present paper.

Coverage and Diversity. For completeness, we also discuss several related works that study reviewer assignment problem.

Li et al. (2015) consider a problem of bias in reviewers' scores. Specifically, they present a greedy assignment algorithm that tries to minimize the impact of the estimation bias on the mean of scores given to each submission. For this, the algorithm aims at heuristically ensuring the *diversity* of the assignment in terms of having different combinations of reviewers assigned to different papers.

Another way to ensure diversity of the assignment is proposed by Liu et al. (2014). Instead of designing the special assignment algorithm, they try to incentivize the diversity by special construction of similarities. Besides incorporating expertise and authority of reviewers in similarities, they add an additional term to the optimization problem which balances similarities by increasing scores for reviewers from different research areas.

Karimzadehgan et al. (2008) consider topic coverage as an objective and propose several approaches to maintain broad coverage, requiring reviewers assigned to paper being expert in different subtopics covered by the paper. They empirically verify that given a paper and a set of reviewers, their algorithms lead to better coverage of paper's topics as compared to baseline technique that assigns reviewers based on some measure of similarity between text of submission and papers authored by reviewers, but does not do topic matching.

A similar goal is formally studied by Long et al. (2013). They measure the coverage of the assignment in terms of the total number of distinct topics of papers covered by the assigned reviewers. They propose a constant factor approximation algorithm that benefits from a sub-modular nature of the objective. As we show in Appendix C, the techniques of Long et al. (2013) can be combined with our proposed algorithm to obtain an assignment which maintains not only fairness, but also a broad topic coverage.

Research on peer review. The explosion in the number of submissions in many conferences has spurred research in computer science on improving peer review. In addition to problems of fairness and accuracy of the reviewer-paper assignment process, there are a number of challenges in peer review which are addressed in the literature to various extents. These include problems of bias (Tomkins et al., 2017; Stelmakh et al., 2019a), miscalibration (Ge et al., 2013; Roos et al., 2011; Flach et al., 2010; Wang and Shah, 2019), subjectivity (Noothigattu et al., 2018), strategic behavior (Balietti et al., 2016; Xu et al., 2019a,b), and others (Lawrence and Cortes, 2014; Gao et al., 2019). Of particular interest is the work by Fiez et al. (2019) which optimizes the process by which reviewers can bid on which papers they prefer to review. In most automated reviewer-paper assignment systems, the bids and the text-matching similarities are then combined (Shah et al., 2018) to form the similarities used to compute the assignment. The bidding and the reviewer-paper assignments are executed separately in current systems, and given the intrinsic relations between the two, it is of interest to jointly design the two systems in the future.

3. Problem Setting

In this section we present the problem setting formally with a focus on the objective of fairness. (We introduce the statistical models we consider in Sections 6 and 7.)

3.1 Preliminaries and Notation

Given a collection of $m \geq 2$ papers, suppose that there exists a true, unknown total ranking of the papers. The goal of the program chair (PC) of the conference is to recover top k papers, for some pre-specified value $k < m$. In order to achieve this goal, the PC recruits $n \geq 2$ reviewers and asks each of them to read and evaluate some subset of the papers. Each reviewer can review a limited number of papers. We let μ denote the maximum number of papers that any reviewer is willing to review. Each paper must be reviewed by λ distinct reviewers. In order to ensure this setting is feasible, we assume that $n\mu \geq m\lambda$. In practice, λ is typically small (2 to 6) and hence should conceptually be thought of as a constant.

The PC has access to a similarity matrix $S = \{s_{ij}\} \in [0,1]^{n \times m}$, where s_{ij} denotes the similarity between any reviewer $i \in [n]$ and any paper $j \in [m]$.[1] These similarities are representative of the envisaged quality of the respective reviews: a higher similarity between any reviewer and paper is assumed to indicate a higher competence of that reviewer in reviewing that paper (this assumption is formalized later). We do not discuss the design of such similarities, but often they are provided by existing systems (Charlin and Zemel, 2013; Mimno and McCallum, 2007; Liu et al., 2014; Rodriguez and Bollen, 2008; Tran et al., 2017).

Our focus is on the assignment of papers to reviewers. We represent any assignment by a matrix $A \in \{0,1\}^{n \times m}$, whose $(i,j)^{\text{th}}$ entry is 1 if reviewer i is assigned paper j and 0 otherwise. We denote the set of reviewers who review paper j under an assignment A as $\mathcal{R}_A(j)$. We call an assignment *feasible* if it respects the (μ, λ) conditions on the reviewer and paper loads. We denote the set of all feasible assignments as $\mathcal{A}$:

$$\mathcal{A} := \Big\{ A \in \{0,1\}^{n \times m} \mid \sum_{i \in [n]} A_{ij} = \lambda \ \forall j \in [m], \sum_{j \in [m]} A_{ij} \leq \mu \ \forall i \in [n] \Big\}.$$

Our goal is to design a reviewer-assignment algorithm with a two-fold objective: (i) fairness to all papers, (ii) strong statistical guarantees in terms of recovering the top papers.

From a statistical perspective, we assume that when any reviewer i is asked to evaluate any paper j, then she/he returns score $y_{ij} \in \mathbb{R}$. The end goal of the PC is to accept or reject each paper. In this work we consider a simplified yet indicative setup. We assume that the PC wishes to accept the k "top" papers from the set of m submitted papers. We denote the "true" set of top k papers as $\mathcal{T}_k^*$. While the PC's decisions in practice would rely on several additional factors including the text comments by reviewers and the discussions between them, in order to quantify the quality of any assignment we assume that the top k papers are chosen through some estimator $\widehat{\theta}$ that operates on the scores provided by the reviewers. Such an estimator can be used in practice to serve as a guide to the program committee in order to help reduce their load. These acceptance decisions can be described by the chosen assignment and estimator $\left(A, \widehat{\theta}\right)$. We denote the set of accepted papers under an assignment A and estimator $\widehat{\theta}$ as $\mathcal{T}_k = \mathcal{T}_k\left(A, \widehat{\theta}\right)$. The PC then wishes to maximize the probability of recovering the set $\mathcal{T}_k^*$ of top k papers.

Although the goal of exact recovering of top k papers is appealing, given the large number of papers submitted to a conference such as ICML and NeurIPS, this goal might

1. Here, we adopt the standard notation $[\nu] = \{1, 2, \ldots, \nu\}$ for any positive integer ν.

be too optimistic. Another alternative is to recover top k papers allowing for a certain Hamming error tolerance $t \in \{0, \ldots, k-1\}$. For any two subsets $\mathcal{M}_1, \mathcal{M}_2$ of $[m]$, we define their Hamming distance to be the number of items that belong to exactly one of the two sets — that is

$$\mathcal{D}_H(\mathcal{M}_1, \mathcal{M}_2) = \text{card}\left(\{\mathcal{M}_1 \cup \mathcal{M}_2\} \setminus \{\mathcal{M}_1 \cap \mathcal{M}_2\}\right). \tag{1}$$

The goal of PC under this scenario is to choose a pair $\left(A, \widehat{\theta}\right)$ such that for the given error tolerance parameter t, the probability $\mathbb{P}\left\{\mathcal{D}_H\left(\mathcal{T}_k, \mathcal{T}_k^*\right) > 2t\right\}$ is minimized. We return to more details on the statistical aspects later in the paper.

3.2 Fairness Objective

An assignment objective that is popular in past papers (Charlin and Zemel, 2013; Charlin et al., 2012; Taylor, 2008) is to maximize the cumulative similarity over all papers. Formally, these works choose an assignment $A \in \mathcal{A}$ which maximizes the quantity

$$G^S(A) := \sum_{j=1}^{m} \sum_{i \in \mathcal{R}_A(j)} s_{ij}. \tag{2}$$

An assignment algorithm that optimizes this objective (2) is implemented in the widely used Toronto Paper Matching System (Charlin and Zemel, 2013). We will refer to the feasible assignment that maximizes the objective (2) as A^{TPMS} and denote the algorithm which computes A^{TPMS} as TPMS.

We argue that the objective (2) does not necessarily lead to a *fair* assignment. The optimal assignment can discriminate some papers in order to maximize the cumulative objective. To see this issue, consider the following example.

Consider a toy problem with $n = m = 3$ and $\mu = \lambda = 1$, with a similarity matrix shown in Table 1. In this example, paper c is easy to evaluate, having non-zero similarities with all the reviewers, while papers a and b are more specific and weak reviewer 2 has no expertise in reviewing them. Reviewer 1 is an expert and is able to assess all three papers. Maximizing total sum of similarities (2), the TPMS algorithm will assign reviewers 1, 2, and 3 to papers a, b, and c respectively. Observe that under this assignment, paper b is assigned a reviewer who has insufficient expertise to evaluate the paper. On the other hand, the alternative assignment which assigns reviewers 1, 2, and 3 to papers a, c, and b respectively ensures that every paper has a reviewer with similarity at least $1/5$. This "fair" assignment does not discriminate against papers a and b for improving the review quality of the already benefiting paper c.

With this motivation, we now formally describe the notion of fairness that we aim to optimize in this paper. Inspired by the notion of max-min fairness in a variety of other fields (Rawls, 1971; Lenstra et al., 1990; Hahne, 1991; Lavi et al., 2003; Bonald et al., 2006; Asadpour and Saberi, 2010), we aim to find a feasible assignment $A \in \mathcal{A}$ to maximize the following objective Γ^S for given similarity matrix S:

$$\Gamma^S(A) = \min_{j \in [m]} \sum_{i \in \mathcal{R}_A(j)} s_{ij}. \tag{3}$$

	Paper a	Paper b	Paper c
Reviewer 1	1	1	1
Reviewer 2	0	0	1/5
Reviewer 3	1/4	1/4	1/2

Table 1: Example similarity.

The assignment optimal for (3) maximizes the minimum sum similarity across all the papers. In other words, for *every other assignment* there exists some paper which has the same or lower sum similarity. Returning to our example, the objective (3) is maximized when reviewers 1, 2, and 3 are assigned to papers a, c, and b respectively.

Our reviewer assignment algorithm presented subsequently guarantees the aforementioned fair assignment. Importantly, while aiming at optimizing (3), our algorithm does even more — having the assignment for the worst-off paper fixed, it finds an assignment that satisfies the second worst-off paper, then the next one and so on until all papers are assigned.

It is important to note that similarities s_{ij} obtained by different techniques (Charlin and Zemel, 2013; Mimno and McCallum, 2007; Rodriguez and Bollen, 2008; Tran et al., 2017) all have different meanings. Therefore, the PC might be interested to consider a slightly more general formulation and aim to maximize

$$\Gamma_f^S(A) = \min_{j\in[m]} \sum_{i\in\mathcal{R}_A(j)} f(s_{ij}), \tag{4}$$

for some reasonable choice of monotonically increasing function $f : [0, 1] \to [0, \infty]$.[2] While the same effect might be achieved by redefining $s'_{ij} = f(s_{ij})$ for all $i \in [n],\ j \in [m]$, this formulation underscores the fact that assignment procedure is not tied to any particular method of obtaining similarities. Different choices of f represent the different views on the meaning of similarities. As a short example, let us consider $f(s_{ij}) = \mathbb{I}\{s_{ij} > \zeta\}$ for some $\zeta > 0$.[3] This choice stratifies reviewers for each paper into strong (similarity higher than ζ) and weak. The fair assignment would be such that the most disadvantaged paper is assigned to as many strong reviewers as possible. We discuss other variants of f later when we come to the statistical properties of our algorithm. In what follows we refer to the problem of finding reviewer assignment that maximizes the term (4) as the *fair assignment problem*.

Unfortunately, the assignment optimal for (4) is hard to compute for any reasonable choices of function f. Garg et al. (2010) showed that finding a fair assignment is an NP-hard problem even if $f(s) \in \{1, 2, 3\}$ and $\lambda = 2$.

With this motivation, in the next section we design a reviewer assignment algorithm that seeks to optimize the objective (4) and provide associated approximation guarantees. We will refer to a feasible assignment that exactly maximizes $\Gamma_f^S(A)$ as A_f^{HARD} and denote the algorithm that computes A_f^{HARD} as Hard. When the function f is clear from context, we drop the subscript f and denote the Hard assignment as A^{HARD} for brevity.

2. We allow $f(s_{ij}) = \infty$. When reviewer with similarity ∞ is assigned to paper, she/he is able to perfectly access the quality of the paper.
3. We use $\mathbb{I}$ to denote the indicator function, that is, $\mathbb{I}\{x\} = 1$ if x is true and $\mathbb{I}\{x\} = 0$ otherwise.

Finally we note that for our running example (Table 1 above), the ILPR algorithm (Garg et al., 2010), despite trying to optimize fairness of the assignment, also returns an unfair assignment A^{ILPR} which coincides with A^{TPMS}. The reason for this behavior lies in the inner-working of the ILPR algorithm: a linear programming relaxation splits reviewers 1 and 2 in two and makes them review both paper a and paper b. During the rounding stage, reviewer 1 is assigned to either paper a or paper b, ensuring that the remaining paper will be reviewed by reviewer 2. Given that reviewer 2 has zero similarity with both papers a and b, the fairness of the resulting assignment will be 0. Such an issue arises more generally in the ILPR algorithm and is discussed in more detail subsequently in Section 5.3 and in Appendix A.1.

4. Reviewer Assignment Algorithm

In this section we first describe our PeerReview4All algorithm followed by an illustrative example.

4.1 Algorithm

A high level idea of the algorithm is the following. For every integer $\kappa \in [\lambda]$, we try to assign each paper to κ reviewers with maximum possible similarities while respecting constraints on reviewer loads. We do so via a carefully designed "subroutine" that is explained below. Continuing for that value of κ, we complement this assignment with $(\lambda - \kappa)$ additional reviewers for each paper. Repeating the procedure for each value of $\kappa \in [\lambda]$, we obtain λ candidate assignments each with λ reviewers assigned to each paper, and then choose the one with the highest fairness. The assignment at this point ensures guarantees of worst-case fairness (4). We then also optimize for the second worst-off paper, then the third worst-off paper and so on in the following manner. In the assignment at this point, we find the most disadvantaged papers and permanently fix corresponding reviewers to these papers. Next, we repeat the procedure described above to find the most fair assignment among the remaining papers, and so on. By doing so, we ensure that our final assignment is not susceptible to bottlenecks which may be caused by irrelevant papers with small average similarities.

The higher-level idea behind the aforementioned subroutine to obtain the candidate assignment for any value of $\kappa \in [\lambda]$ is as follows. The subroutine constructs a layered flow network graph with one layer for reviewers and one layer for papers, that captures the similarities and the constraints on the paper/reviewer loads. Then the subroutine incrementally adds edges between (reviewer, paper) pairs in decreasing order of similarity and stops when the paper load constraints are met (each paper can be assigned to κ reviewers using only edges added at this point). This iterative procedure ensures that the papers are assigned reviewers with approximately the highest possible similarities.

We formally present our main algorithm as Algorithm 1 and the subroutine as Subroutine 1. In what follows, we walk the reader through the steps in the subroutine and the algorithm in more detail.

Subroutine. A key component of our algorithm is a construction of a flow network in a sequential manner in Subroutine 1. The subroutine takes as input, among other arguments, the set $\mathcal{M}$ of papers that are not yet assigned and the required number of reviewers per paper

Subroutine 1 PeerReview4All Subroutine

Input: $\kappa \in [\lambda]$: number of reviewers required per paper
$\mathcal{M}$: set of papers to be assigned
$S \in (\{-\infty\} \cup [0,1])^{n \times |\mathcal{M}|}$: similarity matrix
$(\mu^{(1)}, \dots, \mu^{(n)}) \in [\mu]^n$: reviewers' maximum loads

Output: Reviewer assignment A

Algorithm:

1. Initialize A to an empty assignment
2. Initialize the flow network:
   - **Layer 1:** one vertex (source)
   - **Layer 2:** one vertex for every reviewer $i \in [n]$, and directed edges of capacity $\mu^{(i)}$ and cost 0 from the source to every reviewer
   - **Layer 3:** one vertex for every paper $j \in \mathcal{M}$
   - **Layer 4:** one vertex (sink), and directed edges of capacity κ and cost 0 from each paper to the sink
3. Find (reviewer, paper) pair (i, j) such that the following two conditions are satisfied:
   - the corresponding vertices i and j are not connected in the flow network
   - the similarity s_{ij} is maximal among the pairs which are not connected (ties are broken arbitrarily)

   and call this pair (i', j')
4. Add a directed edge of capacity 1 and cost $s_{i'j'}$ between nodes i' and j'
5. Compute the max-flow from source to sink, if the size of the flow is strictly smaller than $|\mathcal{M}|\kappa$, then go to Step 3
6. If there are multiple possible max-flows, choose any one arbitrarily (or use any heuristic such as max-flow with max cost)
7. For every edge (i, j) between layers 2 (reviewers) and 3 (papers) which carries a unit of flow in the selected max-flow, assign reviewer i to paper j in the assignment A

$\kappa \leq \lambda$. The goal of the subroutine is to assign each paper in $\mathcal{M}$ with κ reviewers, respecting the reviewer load constraints, in a way that minimum similarity across all paper-reviewer pairs in resulting assignment is maximized.

The output of the subroutine is an assignment (represented by variable A) which is initially set as empty (Step 1). The subroutine begins (Step 2) with a construction of a directed acyclic graph (a "flow network") comprising 4 layers in the following order: a source, all reviewers, all papers in $\mathcal{M}$, and a sink. An edge may exist only between consecutive layers. The edges between the first two layers control the reviewers' workloads and edges between the last two layers represent the number of reviews required by the papers. Finally, costs of the all edges in this initial construction are set to 0. An example of the flow network constructed in Step 2 ($n = 4$ reviewers and $|\mathcal{M}| = 3$ papers) is given in Figure 1. Note that in subsequent steps, the edges are added only between the second and third layers. Thus, the maximum flow in the network is at most $|\mathcal{M}|\kappa$.

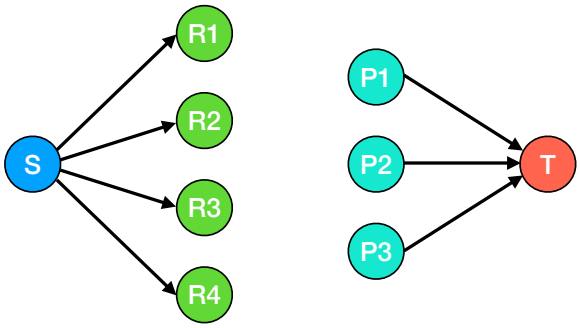


Figure 1: Example of the flow network constructed in Step 2 of Subroutine 1. All edges in the network have costs 0. Capacities of the edges are determined by the input passed to the subroutine.

The crux of the subroutine is to incrementally add edges one at a time between the layers, representing the reviewers and papers, in a carefully designed manner (Steps 3 and 4). The edges are added in order of decreasing similarities. These edges control a reviewer-paper relationship: they have a unit capacity to ensure that any reviewer can review any paper at most once and their costs are equal to the similarity between the corresponding (reviewer, paper) pair.

After adding each edge, the subroutine (Step 5) tests whether a max-flow of size $|\mathcal{M}|\kappa$ is feasible. Note that a feasible flow of size $|\mathcal{M}|\kappa$ corresponds to a feasible assignment: by construction of the flow network described earlier, we know that the reviewer and paper load constraints are satisfied. The capacity of each edge in our flow network is a non-negative integer, thereby guaranteeing that the max-flow is an integer, that it can be found in polynomial time, and that the flow in every edge is a non-negative integer under the max-flow. Once the max-flow of size $|\mathcal{M}|\kappa$ is reached, the subroutine stops adding edges. At this point, it is ensured that the value of the lowest similarity in the resulting assignment is maximized.

Finally, the subroutine assigns each paper to κ reviewers, using only the "high similarity" edges added to the network so far (Steps 6 and 7). The existence of the corresponding assignment is guaranteed by max-flow in the network being equal to $|\mathcal{M}|\kappa$. There may be more than one feasible assignments that attain the max-flow. While any of these assignments would suffice from the standpoint of optimizing the worst-case fairness objective (4), the PC may wish to make a specific choice for additional benefits and specify the heuristic to pick the max-flow in Step 6 of the subroutine. For example, if the max-flow with the maximum cost is selected, then the resulting assignment nicely combines fairness with the high average quality of the assignment. Another choice, discussed in Appendix C, helps with broad topic coverage of the assignment. Importantly, the approximation guarantees established in Theorem 1 and Corollary 2, as well as statistical guarantees from Sections 6 and 7 hold for any max-flow assignment chosen in Steps 6 and 7.

For comparison, we note that the TPMS algorithm can equivalently be interpreted in this framework as follows. The TPMS algorithm would first *connect all reviewers to all*

Algorithm 1 PeerReview4All Algorithm

Input: $\lambda \in [n]$: number of reviewers required per paper
$S \in [0,1]^{n \times m}$: similarity matrix
$\mu \in [m]$: reviewers' maximum load
f: transformation of similarities
Output: Reviewer assignment A_f^{PR4A}
Algorithm:

1. Initialize $\overline{\mu} = (\mu, \dots, \mu) \in [\mu]^n$
   A_f^{PR4A}, A_0 : empty assignments
   $\mathcal{M} = [m]$: set of papers to be assigned
2. For $\kappa = 1$ to λ
   (a) Set $\overline{\mu}^{\text{tmp}} = \overline{\mu}, S^{\text{tmp}} = S$
   (b) Assign κ reviewers to every paper using subroutine:
   $A_\kappa^1 = \text{Subroutine}(\kappa, \mathcal{M}, S^{\text{tmp}}, \overline{\mu}^{\text{tmp}})$
   (c) Decrease $\overline{\mu}^{\text{tmp}}$ for every reviewer by the number of papers she/he is assigned in A_κ^1, set corresponding similarities in S^{tmp} to $-\infty$
   (d) Run subroutine with adjusted $\overline{\mu}^{\text{tmp}}$ and S^{tmp} to assign remaining $\lambda - \kappa$ reviewers to every paper: $A_\kappa^2 = \text{Subroutine}(\lambda - \kappa, \mathcal{M}, S^{\text{tmp}}, \overline{\mu}^{\text{tmp}})$
   (e) Create assignment A_κ such that for every pair (i, j) of reviewer $i \in [n]$ and paper $j \in \mathcal{M}$, reviewer i is assigned to paper j if she/he is assigned to this paper in either A_κ^1 or A_κ^2
3. Choose $\widetilde{A} \in \arg\max_{\kappa \in [\lambda] \cup \{0\}} \Gamma_f^S(A_\kappa)$ with ties broken arbitrarily
4. For every paper $j \in \mathcal{J}^* := \arg\min_{\ell \in \mathcal{M}} \sum_{i \in \mathcal{R}_{\widetilde{A}}(\ell)} f(s_{i\ell})$, assign all reviewers $\mathcal{R}_{\widetilde{A}}(j)$ to paper j in A_f^{PR4A}
5. For every reviewer $i \in [n]$, decrease $\mu^{(i)}$ by the number of papers in $\mathcal{J}^*$ assigned to i
6. Delete columns corresponding to the papers $\mathcal{J}^*$ from S and $\widetilde{A}$, update $\mathcal{M} = \mathcal{M} \backslash \mathcal{J}^*$
7. Set $A_0 = \widetilde{A}$
8. If $\mathcal{M}$ is not empty, go to Step 2

papers in layers 2 and 3 of the flow graph. It will then compute a max-flow with max cost in this fully connected flow network and make reviewer-paper assignments corresponding to the edges with unit flow between layers 2 and 3. In contrast, our sequential construction of the flow graph prevents papers from being assigned to weak reviewers and is crucial towards ensuring the fairness objective.

Algorithm. The algorithm calls the subroutine iteratively and uses the outputs of these iterates in a carefully designed manner. Initially, all papers belong to a set $\mathcal{M}$ which represents papers that are not yet assigned. The algorithm repeats Steps 2 to 7 until all papers are assigned. In every iteration, for every value of $\kappa \in [\lambda]$, the algorithm first calls the subroutine to assign κ reviewers to each paper from $\mathcal{M}$ (Step 2b), and then adjusts reviewers' capacities and the similarity matrix (Step 2c) to prevent any reviewer

being assigned to the same paper twice. Next, the subroutine is called again (Step 2d) to assign another $(\lambda - \kappa)$ reviewers to each paper. As a result, after completion of Step 2, λ feasible candidate assignments $A_1, \ldots, A_\lambda$ are constructed. Each assignment $A_\kappa, \kappa \in [\lambda]$, is guaranteed (through the Step 2b) to maximize the minimum similarity across pairs (i, j) where $j \in \mathcal{M}$ and reviewer i is among κ strongest reviewers assigned to paper j in A_κ; and (through the Steps 2d and 2e) to have each paper assigned with exactly λ reviewers.

In Step 3, the algorithm chooses the assignment with the highest fairness (4) among the λ candidate assignments and the assignment A_0 from the previous iteration (empty in the first iteration). Note that since A_0 is also included in the maximizer, the fairness cannot decrease in subsequent iterations.

In the chosen assignment, the algorithm identifies the papers that are most disadvantaged, and fixes the assignment for these papers (Step 4). The assignment for these papers will not be changed in any subsequent step. The next steps (Steps 5 and 6) update the auxiliary variables to account for this assignment that is fixed — decreasing the corresponding reviewer capacities and removing these assigned papers from the set $\mathcal{M}$. Step 7 then keeps a track of the present assignment $\widetilde{A}$ for use in subsequent iterations, ensuring that fairness cannot decrease as the algorithm proceeds.

Remarks: We make a few additional remarks regarding the PeerReview4All algorithm.

1. Computational cost: A naïve implementation of the PeerReview4All algorithm has a computational complexity $\widetilde{\mathcal{O}}\left(\lambda(m+n)m^2n\right)$. We give more details on implementation and computational aspects in Appendix B.

2. Variable reviewer or paper loads: More generally, the PeerReview4All algorithm allows for specifying different loads for different reviewers and/or papers. For general paper loads, we consider $\kappa \leq \max_{j\in[m]} \lambda^{(j)}$ and define the capacity of edge between node corresponding to any paper j and sink as $\min\{\kappa, \lambda^{(j)}\}$.

3. Incorporating conflicts of interest: One can easily incorporate any conflict of interest between any reviewer and paper by setting the corresponding similarity to $-\infty$.

4. Topic coverage: The techniques developed in Long et al. (2013) can be employed to modify our algorithm in a way that it first ensures fairness and then, among all approximately fair assignments, picks one that approximately maximizes the number of distinct topics of papers covered. We discuss this modification in Appendix C.

4.2 Example

To provide additional intuition behind the design of the algorithm, we now present an example that we also use in the next section to explain our approximation guarantees.

Let for a moment assume that $f(s) = s$ and let ζ be a constant close to 1. Consider the following two scenarios:

(S1) The optimal assignment A^{HARD} is such that all the papers are assigned to reviewers with high similarity:

$$\min_{i \in \mathcal{R}_{A^{\text{HARD}}}(j)} s_{ij} > \zeta \qquad \forall j \in [m]. \tag{5}$$

(S2) The optimal assignment A^{HARD} is such that there are some "critical" papers which have $\eta < \lambda$ assigned reviewers with similarities higher than ζ and the remaining

assigned reviewers with small similarities. All other papers are assigned to λ reviewers with similarity higher than ζ.

Intuitively, the first scenario corresponds to an ideal situation since there exists an assignment such that each paper has λ competent reviewers (with similarity $\zeta \approx 1$). In contrast, in the second scenario, even in the fair assignment, some papers lack expert reviewers. Such a scenario may occur, for example, if some non-mainstream papers were submitted to a conference. This case entails identifying and treating these disadvantaged papers as well as possible. To be able to find the fair assignment in both scenarios, the assignment algorithm should distinguish between them and adapt its behavior to the structure of similarity matrix. Let us track the inner-workings of PeerReview4All algorithm to demonstrate this behaviour.

We note that by construction, the fairness of the resulting assignment A^{PR4A} is determined in the first iteration of Steps 2 to 7 of Algorithm 1, so we restrict our attention to $\mathcal{M} = [m]$. First, consider scenario (S1). The subroutine called with parameter $\kappa = \lambda$ will add edges to the flow network until the maximal flow of size $m\lambda$ is reached. Since the optimal assignment A^{HARD} is such that the lowest similarity is higher than ζ, the last edge added to the flow network will have similarity at least ζ, implying that the fairness of the candidate assignment A_λ, which is a lower bound for the fairness of resulting assignment, will be at least $\lambda\zeta$. Given that ζ is close to one, we conclude that in this case algorithm is able to recover an assignment which is at least very close to optimal.

Now, let us consider scenario (S2). In this scenario, the subroutine called with $\kappa = \lambda$ may return a poor assignment. Indeed, since there is a lack of competent reviewers for critical papers, there is no way to assign each paper with λ reviewers having a high minimum similarity in the assignment. However, the subroutine called with parameter $\kappa = \eta$ will find η strong reviewers for each paper (including the critical papers), thereby leading to a fairness $\Gamma^S\left(A^{\text{PR4A}}\right) \geq \eta\zeta$. The obtained lower bound guarantees that the assignment recovered by the PeerReview4All algorithm is also close to the optimal, because in the fair assignment A^{HARD} some papers have only η strong reviewers.

This example thus illustrates how the PeerReview4All algorithm can adapt to the structure of the similarity matrix in order to guarantee fairness, as well as other guarantees that are discussed subsequently in the paper.

5. Approximation Guarantees

In this section we provide guarantees on the fairness of the reviewer-assignment by our algorithm. We first establish guarantees on the max-min fairness objective introduced earlier (Section 5.1). We subsequently show that our algorithm optimizes not only the worst-off paper but recursively optimizes all papers (Section 5.2). We then conclude this section on deterministic approximation guarantees with a comparison to past literature (Section 5.3).

5.1 Max-min Fairness

We begin with some notation that will help state our main approximation guarantees. For each value of $\kappa \in [\lambda]$, consider the reviewer-assignment problem but where each paper requires κ (instead of λ) reviews (each reviewer still can review up to μ papers). Let us

denote the family of all feasible assignments for this problem as $\mathcal{A}_\kappa$. Now define the quantities

$$\begin{aligned} s^*_\kappa &:= \max_{A\in\mathcal{A}_\kappa} \min_{j\in[m]} \min_{i\in\mathcal{R}_A(j)} s_{ij}, \\ s^*_0 &:= \max_{i\in[n]} \max_{j\in[m]} s_{ij}, \quad \text{and} \\ s^*_\infty &:= \min_{i\in[n]} \min_{j\in[m]} s_{ij}. \end{aligned} \tag{6}$$

Intuitively, for *every* assignment from the family $\mathcal{A}_\kappa$, the quantity s^*_κ upper bounds the minimum similarity for any assigned (reviewer, paper) pair. It also means that the value s^*_κ is achievable by some assignment in $\mathcal{A}_\kappa$. The value s^*_0 captures the value of the largest entry in the similarity matrix S and gives a trivial upper bound $\Gamma^S_f(A) \le \lambda f(s^*_0)$ for every feasible assignment $A \in \mathcal{A}$. Likewise, the value s^*_∞ captures the smallest entry in the similarity matrix S and yields a lower bound $\Gamma^S_f(A) \ge \lambda f(s^*_\infty)$ for every feasible assignment $A \in \mathcal{A}$.

We are now ready to present the main result on the approximation guarantees for the PeerReview4All algorithm as compared to the optimal assignment A^{HARD}.

Theorem 1 *Consider any feasible values of (n, m, λ, μ), any monotonically increasing function $f : [0, 1] \to [0, \infty]$, and any similarity matrix S. The assignment A^{PR4A}_f given by the* PeerReview4All *algorithm guarantees the following lower bound on the fairness objective* (4)*:*

$$\frac{\Gamma^S_f\left(A^{PR4A}_f\right)}{\Gamma^S_f\left(A^{HARD}_f\right)} \ge \frac{\max_{\kappa\in[\lambda]} \left(\kappa f(s^*_\kappa) + (\lambda-\kappa) f(s^*_\infty)\right)}{\min_{\kappa\in[\lambda]} \left((\kappa-1) f(s^*_0) + (\lambda-\kappa+1)\, f(s^*_\kappa)\right)} \tag{7a}$$

$$\ge 1/\lambda. \tag{7b}$$

Remarks: 1. The numerator of (7a) is a lower bound on the fairness of the assignment returned by our algorithm. It is important to note that if $\lambda = 1$, that is, if we only need to assign one reviewer for each paper, then our PeerReview4All Algorithm finds exact solution for the problem, recovering the classical results of Garfinkel (1971) as a special case.

2. In practice, the number of reviewers λ required per paper is a small constant (typically set as 3), and in that case, our algorithm guarantees a constant factor approximation. Note that the fraction in the right hand side of (7a) can become $0/0$ or ∞/∞, and in both cases it should be read as 1.

Early stopping guarantees. Recall that Algorithm 1 iteratively repeats Steps 2 to 7. However, the proof of Theorem 1 implies that the first time Step 3 of the PeerReview4All algorithm is executed, the resulting intermediate assignment $\widetilde{A}$ achieves the fairness guarantees of the theorem. Thus, to save computational time, one can stop the algorithm after the first iteration of Steps 2 to 7 and the fairness guarantee from Theorem 1 will still hold. Moreover, results that we establish in Section 6 and Section 7 will hold for this intermediate assignment as well. However, in Section 5.2 we demonstrate how additional iterations of the algorithm promote fairness of the assignment beyond the most worst-off paper.

The bound (7a) can be significantly tighter than $1/\lambda$, as we illustrate in the following example.

Example 1 *Consider two scenarios (S1) and (S2) from Section 4.2, and consider $f(s) = s$. One can see that under scenario (S1), we have $s^*_\lambda \geq \zeta$. Setting $\kappa = \lambda$ in the numerator and $\kappa = 1$ in the denominator of the bound* (7a)*, and recalling that $\zeta \approx 1$, we obtain:*

$$\frac{\Gamma^S\left(A^{PR4A}\right)}{\Gamma^S\left(A^{HARD}\right)} \geq \frac{\zeta}{s^*_1} \approx 1,$$

where we have also used the fact that $s^*_1 \leq 1$. Let us now consider the second scenario (S2) in the example of Section 4.2. In this scenario, since each paper can be assigned to η strong reviewers with similarity higher than ζ, we have $s^*_\eta = \zeta \approx 1$. We then also have $s^*_0 \leq 1$. Moreover, there are some papers which have only η strong reviewers in optimal assignment A^{HARD}, and hence we have $s^*_{\eta+1} \ll s^*_0$. Setting $\kappa = \eta$ in the numerator and $\kappa = \eta + 1$ in the denominator of the bound (7a)*, some algebraic simplifications yield the bound*

$$\frac{\Gamma^S\left(A^{PR4A}\right)}{\Gamma^S\left(A^{HARD}\right)} \geq \frac{\eta s^*_\eta + (\lambda - \eta)s^*_\infty}{\eta s^*_0 + (\lambda - \eta)s^*_{\eta+1}} \geq \frac{s^*_\eta}{s^*_0} - \frac{(\lambda - \eta)}{\eta}\frac{s^*_{\eta+1}}{s^*_0} \approx 1.$$

We now briefly provide more intuition on the bound (7a) by interpreting it in terms of specific steps in the algorithm. Setting $f(s) = s$, let us consider the first iteration of the algorithm. Recalling the definition (6) of s^*_κ, the PeerReview4All subroutine called with parameter κ on Step 2b finds an assignment such that all the similarities are at least s^*_κ. This guarantee in turn implies that the fairness of the corresponding assignment A_κ is at least $\kappa s^*_\kappa + (\lambda - \kappa)s^*_\infty$, thereby giving rise to the numerator of (7a). The denominator is an upper bound of the fairness of the optimal assignment A^{HARD}. The expression for any value of κ is obtained by simply appealing to the definition of s^*_κ which is defined in terms of the optimal assignment. By definition (6) of s^*_κ, for every feasible assignment A exists at least one paper such that at most $\kappa - 1$ of the assigned reviewers are of similarity larger than s^*_κ. Thus, the fairness of the optimal assignment is upper-bounded by the sum similarity of the paper that has $\kappa - 1$ reviewers with similarity s^*_0 (the highest possible similarity), and $\lambda - \kappa + 1$ reviewers with similarity s^*_κ.

Finally, one may wonder whether optimizing the objective (2) as done by prior works (Charlin and Zemel, 2013; Charlin et al., 2012) can also guarantee fairness. It turns out that this is not the case (see the example in Table 1 for intuition), and optimizing the objective (2) is not a suitable proxy towards the fairness objective (4). In Appendix A.2 we show that in general the fairness objective value of the TPMS algorithm which optimizes (2) may be *arbitrarily bad* as compared to that attained by our PeerReview4All algorithm.

In Appendix A.3 we show that the analysis of the approximation factor of our algorithm is tight in a sense that there exists a similarity matrix for which the bound (7b) is met with equality. That said, the approximation factor of our PeerReview4All algorithm can be much better than $\frac{1}{\lambda}$ for various other similarity matrices, as demonstrated in examples (S1) and (S2).

5.2 Beyond Worst Case

The previous section established guarantees for the PeerReview4All algorithm on the fairness of the assignment in terms of the worst-off paper. In this section we formally show

that the algorithm does more: having the assignment for the worst-off paper fixed, the algorithm then satisfies the second worst-off paper, and so on.

As mentioned in the comment on the early stopping guarantees made after Theorem 1, max-min guarantees of the theorem are achieved after the first iteration of Steps 2 to 7. However, the algorithm does not terminate at this point. Instead, it finds the most disadvantaged papers in the selected assignment and fixes them in the final output A_f^{PR4A} (Step 4), attributing these papers to reviewers according to $\widetilde{A}$. Then it repeats the entire procedure (Steps 2 to 7) again to identify and fix the assignment for the most disadvantaged papers among the remaining papers and so on until the all papers are assigned in A_f^{PR4A}. We denote the total number of iterations of Steps 2 to 7 in Algorithm 1 as p ($\leq m$). For any iteration $r \in [p]$, we let $\mathcal{J}_r$ be the set of papers which the algorithm, in this iteration, fixes in the resulting assignment. We also let $\widetilde{A}_r, r \in [p]$, denote the assignment selected in Step 3 of the r^{th} iteration. Note that eventually all the papers are fixed in the final assignment A_f^{PR4A}, and hence we must have $\bigcup\limits_{r\in[p]} \mathcal{J}_r = [m]$.

Once papers are fixed in the final output A_f^{PR4A}, the assignment for these papers are not changed any more. Thus, at the end of each iteration $r \in [p]$ of Steps 2 to 7, the algorithm deletes (Step 6) the columns of similarity matrix that correspond to the papers fixed in this iteration. For example, at the end of the first iteration, columns which correspond to $\mathcal{J}_1$ are deleted from S. For each iteration $r \in [p]$, we let S_r denote the similarity matrix at the beginning of the iteration. Thus, we have $S_1 = S$, because at the beginning of the first iteration, no papers are fixed in the final assignment A_f^{PR4A}.

Moving forward, we are going to show that for every iteration $r \in [p]$, the sum similarity of the worst-off papers $\mathcal{J}_r$ (which coincides with the fairness of $\widetilde{A}_r$) is close to the best possible, given the assignment for the all papers fixed in the previous iterations. As in Theorem 1, we will compare the fairness $\Gamma_f^S\left(\widetilde{A}_r\right)$ with the fairness of the optimal assignment that Hard algorithm would return if called at the beginning of the r^{th} iteration. We stress that for every $r \in [p]$, the Hard algorithm assigns papers $\bigcup\limits_{l=r}^{p} \mathcal{J}_l$ and respects the constraints on reviewers' loads, adjusted for the assignment of papers $\bigcup\limits_{l=1}^{r-1} \mathcal{J}_l$ in A_f^{PR4A}. We denote the corresponding assignment as $A_f^{\text{HARD}}(\mathcal{J}_{\{r:p\}})$. Note that $A_f^{\text{HARD}}(\mathcal{J}_{\{1:p\}}) = A_f^{\text{HARD}}$. The following corollary summarizes the main result of this section:

Corollary 2 *For any integer $r \in [p]$, the assignment $\widetilde{A}_r$, selected by the* PeerReview4All *algorithm in Step 3 of the r^{th} iteration, guarantees the following lower bound on the fairness objective* (4)*:*

$$\frac{\Gamma_f^S\left(\widetilde{A}_r\right)}{\Gamma_f^S\left(A_f^{HARD}(\mathcal{J}_{\{r:p\}})\right)} \geq \frac{\max\limits_{\kappa\in[\lambda]}\left(\kappa f(s_\kappa^*) + (\lambda-\kappa) f(s_\infty^*)\right)}{\min\limits_{\kappa\in[\lambda]}\left((\kappa-1) f(s_0^*) + (\lambda-\kappa+1)\, f(s_\kappa^*)\right)} \geq 1/\lambda, \tag{8}$$

where values $s_\kappa^*, \kappa \in \{0,\ldots,\lambda\} \cup \{\infty\}$, are defined with respect to the similarity matrix S_r and constraints on reviewers' loads adjusted for the assignment of papers $\bigcup\limits_{l=1}^{r-1} \mathcal{J}_l$ in A_f^{PR4A}.

The corollary guarantees that each time the algorithm fixes the assignment for some papers $j \in \mathcal{M}$ in A_f^{PR4A}, the sum similarity for these papers (which is smallest among papers from $\mathcal{M}$) is close to the optimal fairness, where optimal fairness is conditioned on the previously assigned papers. In case $r = 1$, the bound (8) coincides with the bound (7) from Theorem 1. Hence, once the assignment for the most worst-off papers is fixed, the PeerReview4All algorithm adjusts maximum reviewers' loads and looks for the most fair assignnment of the remaining papers.

5.3 Comparison to Past Literature

In this section we discuss how the approximation results established in previous sections relate to the past literature.

First, we note that the assignment A_1, computed in Step 2 in the first iteration of Steps 2 to 7 of Algorithm 1, recovers the assignment of Hartvigsen et al. (1999), thus ensuring that our algorithm is *at least as fair* as theirs. Second, if the goal is to assign only one reviewer ($\lambda = 1$) to each of the papers, then our PeerReview4All algorithm finds the optimally fair assignment and recovers the classical result of Garfinkel (1971).

In the remainder of this section, we provide a comparison between the guarantees of the PeerReview4All algorithm established in Theorem 1 and the guarantees of the ILPR algorithm (Garg et al., 2010). Rewriting the results of Garg et al. (2010) in our notation, we have the bound:

$$\frac{\Gamma_f^S\left(A_f^{\text{ILPR}}\right)}{\Gamma_f^S\left(A_f^{\text{HARD}}\right)} \geq \frac{\Gamma_f^S\left(A_f^{\text{HARD}}\right) - (f(s_0^*) - f(s_\infty^*))}{\Gamma_f^S\left(A_f^{\text{HARD}}\right)} = 1 - \frac{f(s_0^*) - f(s_\infty^*)}{\Gamma_f^S\left(A_f^{\text{HARD}}\right)}. \tag{9}$$

Note that our bound (7) for our PeerReview4All algorithm is multiplicative and bound for the ILPR algorithm is additive which makes them incomparable in a sense that neither one dominates another. However, we stress the following differences. First, if we assume f to be upper-bounded by one, then assignment A^{ILPR} satisfies the bound

$$\Gamma_f^S\left(A_f^{\text{ILPR}}\right) \geq \Gamma_f^S\left(A_f^{\text{HARD}}\right) - 1. \tag{10}$$

This bound gives a nice additive approximation factor — for a large value of the optimal fairness $\Gamma_f^S\left(A_f^{\text{HARD}}\right)$, the constant additive factor is negligible. However, if the optimal fairness is small, which can happen if some papers do not have a sufficient number of high-expertise reviewers, then the lower bound on the fairness of the ILPR assignment (10) becomes negative, making the guarantees vacuous as any arbitrary assignment will achieve a non-negative fairness. Note that this issue is not an artifact of the analysis but is inherent in the ILPR algorithm itself, as we demonstrate in the example presented in Table 1 and in Appendix A.1. In contrast, our algorithm in the worst case has a multiplicative approximation factor $1/\lambda$ ensuring that it always returns a non-trivial assignment.

This discrepancy becomes more pronounced if the function f is allowed to be unbounded, and the similarities are significantly heterogeneous. Suppose there is some reviewer $i \in [n]$ and paper $j \in [m]$ such that $f(s_{ij}) \gg \Gamma_f^S\left(A^{\text{HARD}}\right)$. Then the bound (9) for the ILPR algorithm again becomes vacuous, while the bound (7) for the PeerReview4All algorithm continues to provide a non-trivial approximation guarantee.

Finally, we note that the bound (9) is also extended by Garg et al. (2010) to obtain guarantees on the fairness for the second worst-off paper and so on.

6. Objective-score Model

We now turn to establishing statistical guarantees for our PeerReview4All algorithm from Section 4. We begin by considering an "objective" score model which we borrow from past works.

6.1 Model Setup

The objective-score model assumes that each paper $j \in [m]$ has a true, unknown quality $\theta^*_j \in \mathbb{R}$ and each reviewer $i \in [n]$ assigned to paper j gives her/his estimate y_{ij} of θ^*_j. The eventual goal is to estimate top k papers according to true qualities $\theta^*_j, j \in [m]$. Following the line of works by Ge et al. (2013); McGlohon et al. (2010); Dai et al. (2012); Sajjadi et al. (2016), we assume the score y_{ij} given by any reviewer $i \in [n]$ to any paper $j \in [m]$ to be independently and normally distributed around the true paper qualities:

$$y_{ij} \sim \mathcal{N}\left(\theta^*_j, \sigma^2_{ij}\right). \tag{11}$$

Note that McGlohon et al. (2010); Dai et al. (2012) and Sajjadi et al. (2016) consider the restricted setting with $\sigma_{ij} = \sigma_i$ for all $(i, j) \in [n] \times [m]$, which implies that the variance of the reviewers' scores depends only on the reviewer, but not on the paper reviewed. We claim that this assumption is not appropriate for our peer-review problem: conferences today (such as ICML and NeurIPS) cover a wide spectrum of research areas and it is not reasonable to expect the reviewer to be equally competent in all of the areas.

In our analysis, we assume that the noise variances are some function of the underlying computed similarities.[4] We assume that for any $i \in [n]$ and $j \in [m]$, the noise variance

$$\sigma^2_{ij} = h(s_{ij}),$$

for some monotonically decreasing function $h : [0, 1] \to [0, \infty)$. We assume that this function h is known; this assumption is reasonable as the function can, in principle, be learned from the data from the past conferences.

We note that the model (11) does not consider reviewers' biases. However, some reviewers might be more stringent while others are more lenient. This difference results in score of any reviewer i for any paper j being centered not at θ^*_j, but at $(\theta^*_j + b_i)$. A common approach to reduce biases in reviewers' scores is a post-processing. For example, Ge et al. (2013) compared different statistical models of reviewers in attempt to calibrate the biases; the techniques developed in that work may be extended to the reviewer model (11). Thus, we leave that bias term out for simplicity.

6.2 Estimator

Given a valid assignment $A \in \mathcal{A}$, the goal of an estimator is to recover the top k papers. A natural way to do so is to compute the estimates of true paper scores θ^*_j and return

4. Recall that the similarities can capture not only affinity in research areas but may also incorporate the bids or preferences of reviewers, past history of review quality, etc.

top k papers with respect to these estimated scores. The described estimation procedure is a significantly simplified version of what is happening in the real-world conferences. Nevertheless, this fully-automated procedure may serve as a guideline for area chairs, providing a first-order estimate of the total ranking of submitted papers. In what follows, we refer to any estimator as $\widehat{\theta}$ and to the estimated score of any paper j as $\widehat{\theta}_j$. Specifically, we consider the following two estimators:

- Maximum likelihood estimator (MLE) $\widehat{\theta}^{\text{MLE}}$

$$\widehat{\theta}_j^{\text{MLE}} = \frac{1}{\sum\limits_{i \in \mathcal{R}_A(j)} \frac{1}{\sigma_{ij}^2}} \sum_{i \in \mathcal{R}_A(j)} \frac{y_{ij}}{\sigma_{ij}^2} \sim \mathcal{N}\left(\theta_j^*, \frac{1}{\sum\limits_{i \in \mathcal{R}_A(j)} \frac{1}{\sigma_{ij}^2}}\right). \tag{12}$$

  Under the model (11), $\widehat{\theta}_j^{\text{MLE}}$ is known to have minimal variance across all linear unbiased estimations. The choice of $\widehat{\theta}^{\text{MLE}}$ follows a paradigm that more experienced reviewers should have higher weight in decision making.

- Mean score estimator (MEAN) $\widehat{\theta}^{\text{MEAN}}$

$$\widehat{\theta}_j^{\text{MEAN}} = \frac{1}{\lambda} \sum_{i \in \mathcal{R}_A(j)} y_{ij} \sim \mathcal{N}\left(\theta_j^*, \frac{1}{\lambda^2} \sum_{i \in \mathcal{R}_A(j)} \sigma_{ij}^2\right). \tag{13}$$

  The mean score estimator is convenient in practice because it is not tied to the assumed statistical model, and in the past has been found to be predictive of final acceptance decisions in peer-review settings such as National Science Foundation grant proposals (Cole et al., 1981) and homework grading (Sajjadi et al., 2016). This observation is supported by the program chair of ICML 2012 John Langford, who notices in his blog (Langford, 2012) that in ICML 2012 the decisions on the acceptance were "surprisingly uniform as a function of average score in reviews".

6.3 Analysis

Here we present statistical guarantees for both $\widehat{\theta}^{\text{MLE}}$ and $\widehat{\theta}^{\text{MEAN}}$ estimators and for both exact top k recovery and recovery under a Hamming error tolerance.

6.3.1 Exact Top k Recovery

Let us use (k) and $(k+1)$ to denote the indices of the papers that are respectively ranked k^{th} and $(k+1)^{\text{th}}$ according to their true qualities. Similar to the past work by Shah and Wainwright (2015) on top k item recovery, a central quantity in our analysis is a *k-separation threshold* Δ_k defined as:

$$\Delta_k := \theta^*_{(k)} - \theta^*_{(k+1)} > 0. \tag{14}$$

Intuitively, if the difference between k^{th} and $(k+1)^{\text{th}}$ papers is large enough, it should be easy to recover top k papers. To formalize this intuition, for any value of a parameter $\delta \geq 0$,

consider a family $\mathcal{F}_k$ of papers' scores

$$\mathcal{F}_k(\delta) := \left\{ (\theta_1, \dots, \theta_m) \in \mathbb{R}^m \middle| \theta_{(k)} - \theta_{(k+1)} \geq \delta \right\}. \tag{15}$$

For the first half of this section, we assume that function h is bounded, that is, $h : [0,1] \to [0,1]$.[5] This assumption implicitly assumes that every reviewer $i \in [n]$ can provide a minimum level of expertise while reviewing any paper $j \in [m]$ even if she/he has zero similarity $s_{ij} = 0$ with that paper.

In addition to the gap Δ_k, the hardness of the problem also depends on the similarities between reviewers and papers. For instance, if all reviewers have near-zero similarity with all the papers, then recovery is impossible unless the gap is extremely large. In order to quantify the tractability of the problem in terms of the similarities we introduce the following set $\mathcal{S}$ of families of similarity matrices parameterized by a non-negative value q:

$$\mathcal{S}(q) := \left\{ S \in [0,1]^{n \times m} \middle| \Gamma^S_{1-h}\left(A^{\mathrm{HARD}}_{1-h}\right) \geq q \right\}. \tag{16}$$

In words, if similarity matrix S belongs to $\mathcal{S}(q)$, then the fairness of the optimally fair (with respect to $f = 1 - h$) assignment is at least q.

Finally, we define a quantity τ_q that captures the quality of approximation provided by PeerReview4All:

$$\tau_q := \inf_{S \in \mathcal{S}(q)} \frac{\Gamma^S_{1-h}\left(A^{\mathrm{PR4A}}_{1-h}\right)}{\Gamma^S_{1-h}\left(A^{\mathrm{HARD}}_{1-h}\right)}. \tag{17}$$

Note that Theorem 1 gives lower bounds on the value of τ_q.

Having defined all the necessary notation, we are ready to present the first result of this section on recovering the set of top k papers $\mathcal{T}^*_k$.

Theorem 3 *(a) For any $\epsilon \in (0, 1/4)$, $q \in [\lambda(1 - h(0)), \lambda]$ and any monotonically decreasing $h : [0,1] \to [0,1]$, if $\delta > \frac{2\sqrt{2}}{\lambda}\sqrt{(\lambda - q\tau_q)\ln\frac{m}{\sqrt{\epsilon}}}$, then for*

$$\left(A, \widehat{\theta}\right) \in \left\{ \left(A^{PR4A}_{1-h}, \widehat{\theta}^{MEAN}\right), \left(A^{PR4A}_{h^{-1}}, \widehat{\theta}^{MLE}\right) \right\}$$

we have

$$\sup_{\substack{(\theta^*_1, \dots, \theta^*_m) \in \mathcal{F}_k(\delta) \\ S \in \mathcal{S}(q)}} \mathbb{P}\left\{ \mathcal{T}_k\left(A, \widehat{\theta}\right) \neq \mathcal{T}^*_k \right\} \leq \epsilon. \tag{18}$$

(b) Conversely, for any continuous strictly monotonically decreasing $h : [0,1] \to [0,1]$ and any $q \in [\lambda(1 - h(0)), \lambda]$, there exists a universal constant $c > 0$ such that if $m > 6$ and $\delta < \frac{c}{\lambda}\sqrt{(\lambda - q)\ln m}$, then

$$\sup_{S \in \mathcal{S}(q)} \inf_{\left(\widehat{\theta}, A \in \mathcal{A}\right)} \sup_{(\theta^*_1, \dots, \theta^*_m) \in \mathcal{F}_k(\delta)} \mathbb{P}\left\{ \mathcal{T}_k\left(A, \widehat{\theta}\right) \neq \mathcal{T}^*_k \right\} \geq \frac{1}{2}.$$

5. More generally, we could consider bounded function h with range $[0, c]$ for some $c > 0$. Without loss of generality, we set $c = 1$ which can always be achieved by appropriate scaling.

Remarks: 1. The PeerReview4All assignment algorithm thus leads to a strong minimax guarantee on the recovery of the top k papers: the upper and lower bounds differ by at most a $\tau_q \geq \frac{1}{\lambda}$ term in the requirement on δ and constant pre-factor. Also note that as discussed in Section 5.1, approximation factor τ_q of the PeerReview4All algorithm can be much better than $1/\lambda$ for various similarity matrices.

2. In addition to quantifying the performance of PeerReview4All, an important contribution of Theorem 3 is a sharp minimax analysis of the performance of *every* assignment algorithm. Indeed, the approximation ratio τ_q (17) can be defined for any assignment algorithm, by substituting corresponding assignment instead of A^{PR4A}_{1-h}. For example, if one has access to the optimal assignment A^{HARD} (e.g., by using PeerReview4All if $\lambda = 1$) then we will have corresponding approximation ratio $\tau_q = 1$ thereby yielding bounds that are sharp up to constant pre-factors.

3. While on one hand the estimator $\widehat{\theta}^{\mathrm{MLE}}$ is preferred over $\widehat{\theta}^{\mathrm{MEAN}}$ when model (11) is correct, on the other hand, if $h(s) \in [0, 1]$, then the estimator $\widehat{\theta}^{\mathrm{MEAN}}$ is more robust to model mismatches.

4. The technical assumption $q \in [\lambda\,(1 - h(0))\,, \lambda]$ is made without loss of any generality, because values of q outside this range are vacuous. In more detail, for any similarity matrix $S \in [0, 1]^{n\times m}$, it must be that $\Gamma^S_{1-h}\left(A^{\mathrm{HARD}}_{1-h}\right) \geq \lambda\,(1 - h(0))$. Moreover, the co-domain of function h comprises only non-negative real values, implying that $\Gamma^S_{1-h}\left(A^{\mathrm{HARD}}_{1-h}\right) \leq \lambda$ for any similarity matrix $S \in [0, 1]^{n\times m}$.

5. The upper bound of the theorem holds for a slightly more general model of reviewers — reviewers with sub-Gaussian noise. Formally, in addition to the Gaussian noise model (11), the proof of Theorem 3(a) also holds for the following class of distributions of the score y_{ij}:

$$y_{ij} = \theta^*_{ij} + sG\left(h(s_{ij})\right), \tag{19}$$

where $sG\left(\sigma^2\right)$ is an arbitrary mean zero sub-Gaussian random variable with scale parameter σ^2.

The conditions of Theorem 3 require function h to be bounded. We now relax our earlier boundedness assumption on h and consider $h : [0, 1] \to [0, \infty)$.

In what follows we restrict our attention to MLE estimator $\widehat{\theta}^{\mathrm{MLE}}$ which represents the paradigm that reviewers with higher similarity should have more weight in the final decision. In order to demonstrate that our PeerReview4All algorithm is able to adapt to different structures of similarity matrices — from hard cases when optimal assignment provides only one strong reviewer for some of the papers, to ideal cases when there are λ strong reviewers for every paper — let us consider the following set $\mathcal{S}_\kappa$ of families of similarity matrices parametrized by a non-negative value v and integer parameter $\kappa \in [\lambda]$:

$$\mathcal{S}_\kappa(v) := \left\{ S \in [0, 1]^{n\times m} \middle| s^*_\kappa \geq v \right\}. \tag{20}$$

Here s^*_κ is as defined in (6).

In words, the parameter v defines the notion of strong reviewer while parameter κ denotes the maximum number of strong (with similarity higher than v) reviewers that can be assigned to each paper without violating the (μ, λ) conditions.

Then the following adaptive analogue of Theorem 3 holds:

Corollary 4 *(a) For any $\epsilon \in (0, 1/4)$, $v \in [0,1]$, $\kappa \in [\lambda]$ and any monotonically decreasing $h : [0,1] \to [0,\infty)$, if $\delta > 2\sqrt{2}\sqrt{\frac{h(v)h(0)}{\kappa h(0)+(\lambda-\kappa)h(v)}} \ln \frac{m}{\sqrt{\epsilon}}$, then*

$$\sup_{\substack{(\theta_1^*,\ldots,\theta_m^*)\in\mathcal{F}_k(\delta)\\ S\in\mathcal{S}_\kappa(v)}} \mathbb{P}\left\{\mathcal{T}_k(A_{h^{-1}}^{PR4A}, \widehat{\theta}^{MLE}) \neq \mathcal{T}_k^*\right\} \leq \epsilon.$$

(b) Conversely, for any continuous strictly monotonically decreasing $h : [0,1] \to [0,\infty)$, any $v \in [0,1]$, and any $\kappa \in [\lambda]$, there exists a universal constant $c > 0$ such that if $m > 6$ and $\delta \leq c\sqrt{\frac{h(v)h(0)}{\kappa h(0)+(\lambda-\kappa)h(v)}} \ln m$, then

$$\sup_{S\in\mathcal{S}_\kappa(v)} \inf_{\left(\widehat{\theta}, A\in\mathcal{A}\right)} \sup_{(\theta_1^*,\ldots,\theta_m^*)\in\mathcal{F}_k(\delta)} \mathbb{P}\left\{\mathcal{T}_k(A,\widehat{\theta}) \neq \mathcal{T}_k^*\right\} \geq \frac{1}{2}.$$

Remarks: 1. Observe that there is no approximation factor in the upper bound. Thus, the PeerReview4All algorithm together with $\widehat{\theta}^{\text{MLE}}$ are simultaneously minimax optimal up to a constant pre-factor in classes of similarity matrices $\mathcal{S}_\kappa(v)$ for all $\kappa \in [\lambda]$, $v \in [0,1]$.

2. Corollary 4(a) remains valid for generalized sub-Gaussian model of reviewer (19).

3. Corollary 4 together with Theorem 3 show that our PeerReview4All algorithm produces the assignment $A_{h^{-1}}^{\text{PR4A}}$ which is simultaneously minimax (near-)optimal for various classes of similarity matrices. We thus see that our PeerReview4All algorithm is able to adapt to the underlying structure of similarity matrix S in order to construct an assignment in which even the most disadvantaged paper gets reviewers with sufficient expertise to estimate the true quality of the paper.

6.3.2 Approximate Recovery under Hamming Error

Although our ultimate goal is to recover set $\mathcal{T}_k^*$ of top k papers exactly, we note that often scores of boundary papers are close to each other so it may be impossible to distinguish between the k^{th} and $(k+1)^{\text{th}}$ papers in the total ranking. Thus, a more realistic goal would be to try to accept papers such that the set of accepted papers is in some sense "close" to the set $\mathcal{T}_k^*$. In this work we consider the standard notion of Hamming distance (1) as a measure of closeness. We are interested in minimizing the quantity:

$$\mathbb{P}\left\{\mathcal{D}_H\left(\mathcal{T}_k\left(A,\widehat{\theta}\right), \mathcal{T}_k^*\right) > 2t\right\}$$

for some user-defined value of $t \in [k-1]$.

Similar to the exact recovery setup, the key role in the analysis is played by generalized separation threshold (compare with equation 14):

$$\Delta_{k,t} := \theta^*_{(k-t)} - \theta^*_{(k+t+1)},$$

where $(k-t)$ and $(k+t+1)$ are indices of papers that take $(k-t)^{\text{th}}$ and $(k+t+1)^{\text{th}}$ positions respectively in the underlying total ranking. For any value of $\delta > 0$ we consider the following generalization of the set $\mathcal{F}_k(\delta)$ defined in (15):

$$\mathcal{F}_{k,t}(\delta) := \left\{(\theta_1,\ldots,\theta_m) \in \mathbb{R}^m \middle| \theta_{(k-t)} - \theta_{(k+t+1)} \geq \delta\right\}.$$

Also recall the family of matrices $\mathcal{S}(q)$ from (16) and the approximation factor τ_q from (17) for any parameter q. With this notation in place, we now present the analogue of Theorem 3 in case of approximate recovery under the Hamming error.

Theorem 5 *(a) For any $\epsilon \in (0, 1/4)$, $q \in [\lambda(1-h(0)), \lambda]$, $t \in [k-1]$, and any monotonically decreasing $h : [0,1] \to [0,1]$, if $\delta > \frac{2\sqrt{2}}{\lambda}\sqrt{(\lambda - q\tau_q)\ln\frac{m}{\sqrt{\epsilon}}}$, then for $\left(A, \widehat{\theta}\right) \in \left\{\left(A^{PR4A}_{1-h}, \widehat{\theta}^{MEAN}\right), \left(A^{PR4A}_{h^{-1}}, \widehat{\theta}^{MLE}\right)\right\}$*

$$\sup_{\substack{(\theta^*_1,\dots,\theta^*_m)\in\mathcal{F}_{k,t}(\delta)\\ S\in\mathcal{S}(q)}} \mathbb{P}\left\{\mathcal{D}_H\left(\mathcal{T}_k\left(A, \widehat{\theta}\right), \mathcal{T}^*_k\right) > 2t\right\} \le \epsilon.$$

(b) Conversely, for any continuous strictly monotonically decreasing $h : [0,1] \to [0,1]$, any $q \in [\lambda(1-h(0)), \lambda]$, and any $0 < t < k$, there exists a universal constant $c > 0$ such that for given constants $\nu_1 \in (0;1)$ and $\nu_2 \in (0,1)$ if $2t \le \frac{1}{1+\nu_2}\min\left\{m^{1-\nu_1}, k, m-k\right\}$ and $\delta \le \frac{c}{\lambda}\sqrt{(\lambda - q)\,\nu_1\nu_2 \ln m}$, then for m larger than some (ν_1,ν_2)-dependent constant,

$$\sup_{S\in\mathcal{S}(q)} \inf_{\left(\widehat{\theta}, A\in\mathcal{A}\right)} \sup_{(\theta^*_1,\dots,\theta^*_m)\in\mathcal{F}_{k,t}(\delta)} \mathbb{P}\left\{\mathcal{D}_H\left(\mathcal{T}_k\left(A, \widehat{\theta}\right), \mathcal{T}^*_k\right) > 2t\right\} \ge \frac{1}{2}.$$

Remark: This theorem provides a strong minimax characterization of the PeerReview4All algorithm for approximate recovery. Note that upper and lower bounds differ by the approximation factor τ_q, which is at most $\frac{1}{\lambda}$, and a pre-factor which depends only on the constants ν_1 and ν_2.

To conclude the section, we state the result for the family $\mathcal{S}_\kappa(v)$ of similarity matrices defined in (20) for any parameter v, showing that adaptive behavior of PeerReview4All algorithm (Corollary 4) also carries over to the Hamming error metric.

Corollary 6 *(a) For any $\epsilon \in (0, 1/4)$, $v \in [0,1]$, $\kappa \in [\lambda]$, $t \in [k-1]$, and any monotonically decreasing $h : [0,1] \to [0,\infty)$, if $\delta > 2\sqrt{2}\sqrt{\frac{h(v)h(0)}{\kappa h(0) + (\lambda-\kappa)h(v)}\ln\frac{m}{\sqrt{\epsilon}}}$, then*

$$\sup_{\substack{(\theta^*_1,\dots,\theta^*_m)\in\mathcal{F}_{k,t}(\delta)\\ S\in\mathcal{S}_\kappa(v)}} \mathbb{P}\left\{\mathcal{D}_H\left(\mathcal{T}_k\left(A^{PR4A}_{h^{-1}}, \widehat{\theta}^{MLE}\right), \mathcal{T}^*_k\right) > 2t\right\} \le \epsilon.$$

(b) Conversely, for any continuous strictly monotonically decreasing $h : [0,1] \to [0,\infty)$, any $v \in [0,1]$, $\kappa \in [\lambda]$ and any $t \in [k-1]$, there exists a universal constant $c > 0$ such that for given constants $\nu_1 \in (0;1)$ and $\nu_2 \in (0,1)$ if $2t \le \frac{1}{1+\nu_2}\min\left\{m^{1-\nu_1}, k, m-k\right\}$ and $\delta \le c\sqrt{\frac{h(v)h(0)}{\kappa h(0) + (\lambda-\kappa)h(v)}\nu_1\nu_2 \ln m}$, then for m larger than some (ν_1,ν_2)-dependent constant,

$$\sup_{S\in\mathcal{S}_\kappa(v)} \inf_{\left(\widehat{\theta}, A\in\mathcal{A}\right)} \sup_{(\theta^*_1,\dots,\theta^*_m)\in\mathcal{F}_{k,t}(\delta)} \mathbb{P}\left\{\mathcal{D}_H\left(\mathcal{T}_k\left(A, \widehat{\theta}\right), \mathcal{T}^*_k\right) > 2t\right\} \ge \frac{1}{2}.$$

The results established in this section thus show that our PeerReview4All algorithm produces an assignment which is minimax (near-)optimal for both exact and approximate recovery of the top k papers.

7. Subjective-score Model

In the previous section, we analyzed the performance of our PeerReview4All assignment algorithm under a model with objective scores. Indeed, various past works on peer-review (as well as various other domains of machine learning) assume existence of some "true" objective scores or ranking of the underlying items (papers). However, in practice, reviewers' opinions on the quality of any paper are typically highly subjective (Kerr et al., 1977; Mahoney, 1977; Ernst and Resch, 1994; Bakanic et al., 1987; Lamont, 2009). Even two highly experienced researchers with vast experience and expertise may have considerably differing opinions about the contributions of a paper. Following this intuition, we wish to move away from the assumption of some true objective scores $\{\theta^*_j\}_{j\in[m]}$ of the paper.

With this motivation, in this section we develop a novel model to capture such subjective opinions and present a statistical analysis of our assignment algorithm under this subjective-score model.

7.1 Model

The key idea behind our subjective score model is to separate out the subjective part in any reviewer's opinion from the noise inherent in it. Our model is best described by first considering a hypothetical situation where every reviewer *spends an infinite time and effort on reviewing every paper, gaining a perfect expertise in the field of that paper and a perfect understanding of the paper's content.* We let $\widetilde{\theta}_{ij} \in \mathbb{R}$ denote the score that this fully competent version of reviewer $i \in [n]$ would provide to paper $j \in [m]$, and denote the matrix of reviewers subjective scores as $\widetilde{\Theta} = \left\{\widetilde{\theta}_{ij}\right\}_{i\in[n],j\in[m]}$. Continuing momentarily in this hypothetical world, when all the reviewers are fully competent in evaluating all the papers, every feasible reviewer-assignment is of the same quality since there is no noise in the reviewers' scores. Since all reviewers have an equal, full competence, a natural choice of scoring any paper $j \in [m]$ is to take the mean score provided by the fully competent reviewers who review that paper:

$$\widetilde{\theta}^\star_j(A) := \frac{1}{\lambda} \sum_{i\in\mathcal{R}_A(j)} \widetilde{\theta}_{ij}. \tag{21}$$

Let us now exit our hypothetical world and return to reality. In a real conference peer-review setting the reviews will be noisy. Following the previous noise assumptions, we assume that score of any reviewer $i \in [n]$ for any paper $j \in [m]$ that she/he reviews is distributed as

$$y_{ij} \sim \mathcal{N}(\widetilde{\theta}_{ij}, h(s_{ij})),$$

for some known continuous strictly monotonically decreasing function $h : [0, 1] \to [0, 1]$. Under this model, the higher the similarity s_{ij}, the better the score y_{ij} represents the subjective score $\widetilde{\theta}_{ij}$ which reviewer $i \in [n]$ would give to paper $j \in [m]$ if she/he had infinite expertise.

The goal under this model is to assign reviewers to papers such that reviewers are of enough ability to convey their opinions $\widetilde{\theta}_{ij}$ from the hypothetical full-competence world

to the real world with scores y_{ij}. In other words, the goal of the assignment is to ensure the recovery of the top k papers in terms of the mean full-competence subjective scores $\{\widetilde{\theta}^\star_j\}_{j\in[m]}$.

7.2 Analysis

In this section we present statistical guarantees for $\widehat{\theta}^{\text{MEAN}}$ in context of subjective-score model.

7.2.1 Exact Top k Recovery

Since the true scores for any reviewer-paper pair are subjective, and since we are interested in mean full-competence subjective scores, a natural choice for estimating $\{\widetilde{\theta}^\star_j\}$ from the actual provided scores $\{y_{ij}\}$ is the averaging estimator $\widehat{\theta}^{\text{MEAN}}$ which for every paper $j \in [m]$ estimates $\widetilde{\theta}^\star_j$ as $\widehat{\theta}^{\text{MEAN}}_j = \frac{1}{\lambda}\sum\limits_{i\in\mathcal{R}_A(j)} y_{ij}$. Having defined the model and estimator, we now provide a sharp minimax analysis for the subjective-score model. In order to state our main result, we recall the family of similarity matrices $\mathcal{S}(q)$ defined earlier in (16) and the approximation ratio τ_q defined in (17), both parameterized by some non-negative value q.

Note that the notion of the k-separation threshold (14) does not carry over directly from the objective score model to the subjective score model. The reason is that the ranking now is induced by the assignment and changes as we change the assignment. Consequently, we introduce the following family of papers' scores that are governed by the assignment A and parametrized by a positive real value δ:

$$\mathcal{F}_k(A,\delta) = \left\{\widetilde{\Theta}\in\mathbb{R}^{n\times m}\middle|\widetilde{\theta}^\star_{(k)}(A) - \widetilde{\theta}^\star_{(k+1)}(A) \geq \delta\right\}. \tag{22}$$

Since in this section we consider only mean score estimator $\widehat{\theta}^{\text{MEAN}}$, we omit index $1-h$ from A^{PR4A}_{1-h}, but always imply that assignment A^{PR4A} is built with respect to the function $1-h$. For every feasible assignment A, we augment the notation $\mathcal{T}^*_k$ with $\mathcal{T}^\star_k\left(A,\widetilde{\theta}^\star(A)\right)$ to highlight that the set of the top k papers is induced by the assignment A. Let us now present the main result of this section.

Theorem 7 *(a) For any $\epsilon\in(0,1/4)$, $q\in[\lambda(1-h(0)),\lambda]$ and any monotonically decreasing $h:[0,1]\to[0,1]$, if $\delta > \frac{2\sqrt{2}}{\lambda}\sqrt{(\lambda-q\tau_q)\ln\frac{m}{\sqrt{\epsilon}}}$, then*

$$\sup_{\substack{\widetilde{\Theta}\in\mathcal{F}_k(A^{PR4A},\delta)\\ S\in\mathcal{S}(q)}} \mathbb{P}\left\{\mathcal{T}_k(A^{PR4A},\widehat{\theta}^{MEAN}) \neq \mathcal{T}^\star_k\left(A^{PR4A},\widetilde{\theta}^\star(A^{PR4A})\right)\right\} \leq \epsilon.$$

(b) Conversely, for any continuous strictly monotonically decreasing $h:[0,1]\to[0,1]$ and any $q\in[\lambda(1-h(0)),\lambda]$, there exists a universal constant $c>0$ such that if $m>6$ and $\delta < \frac{c}{\lambda}\sqrt{(\lambda-q)\ln m}$, then

$$\sup_{S\in\mathcal{S}(q)}\inf_{\left(\widehat{\theta},A\in\mathcal{A}\right)}\sup_{\widetilde{\Theta}\in\mathcal{F}_k(A,\delta)} \mathbb{P}\left\{\mathcal{T}_k(A,\widehat{\theta}) \neq \mathcal{T}^\star_k\left(A,\widetilde{\theta}^\star(A)\right)\right\} \geq \frac{1}{2}.$$

We thus see that our assignment algorithm PeerReview4All not only leads to the strong guarantees under the objective-score model but simultaneously also under the setting where the opinions of reviewers may be subjective.

7.2.2 Approximate Recovery under Hamming Error

We now present guarantees for approximate recovering under the Hamming error for the PeerReview4All algorithm. We generalize the family of score matrices (22), for which we consider any integer error tolerance parameter $t \in \{0, \dots, k-1\}$ and any any feasible assignment A. Then we define the following family of subjective papers' scores, parameterized by non-negative value δ:

$$\mathcal{F}_{k,t}(A, \delta) = \left\{ \widetilde{\Theta} \in \mathbb{R}^{n \times m} \middle| \widetilde{\theta}^{\star}_{(k-t)}(A) - \widetilde{\theta}^{\star}_{(k+t+1)}(A) \geq \delta \right\}.$$

Observe that the class $\mathcal{F}_{k,t}(A, \delta)$ coincides with the class $\mathcal{F}_k(\delta)$ from (22) when $t = 0$.

Theorem 8 *(a) For any $\epsilon \in (0, 1/4)$, $q \in [0, \lambda]$, $t \in [k-1]$, and any monotonically decreasing $h : [0,1] \to [0,1]$, if $\delta > \frac{2\sqrt{2}}{\lambda}\sqrt{(\lambda - q\tau_q) \ln \frac{m}{\sqrt{\epsilon}}}$, then*

$$\sup_{\substack{\widetilde{\Theta} \in \mathcal{F}_{k,t}(A^{PR4A}, \delta) \\ S \in \mathcal{S}(q)}} \mathbb{P}\left\{ \mathcal{D}_H\left(\mathcal{T}_k\left(A^{PR4A}, \widehat{\theta} \right), \mathcal{T}_k^*\left(A^{PR4A}, \widetilde{\theta}^{\star}(A^{PR4A}) \right) \right) > 2t \right\} \leq \epsilon.$$

Conversely, for any continuous strictly monotonically decreasing $h : [0,1] \to [0,1]$, any $q \in [\lambda(1 - h(0)), \lambda]$, and any $0 < t < k$, there exists a universal constant $c > 0$ such that for given constants $\nu_1 \in (0,1)$ and $\nu_2 \in (0,1)$ if $2t \leq \frac{1}{1+\nu_2} \min\left\{ m^{1-\nu_1}, k, m-k \right\}$ and $\delta \leq \frac{c}{\lambda}\sqrt{(\lambda - q)\nu_1\nu_2 \ln m}$, then for m larger than some (ν_1, ν_2)-dependent constant,

$$\sup_{S \in \mathcal{S}(q)} \inf_{(\widehat{\theta}, A \in \mathcal{A})} \sup_{\widetilde{\Theta} \in \mathcal{F}_{k,t}(A, \delta)} \mathbb{P}\left\{ \mathcal{D}_H\left(\mathcal{T}_k\left(A, \widehat{\theta} \right), \mathcal{T}_k^*\left(A, \widetilde{\theta}^{\star}(A) \right) \right) > 2t \right\} \geq \frac{1}{2}.$$

Similar to Theorem 7, Theorem 8 shows that PeerReview4All algorithm is minimax optimal up to a constant pre-factor and approximation factor *given* that reviewers' subjective scores $\widetilde{\Theta}$ belong to the class $\mathcal{F}_{k,t}(A, \delta)$.

8. Experiments

In this section we conduct empirical evaluations of the PeerReview4All algorithm and compare it with the TPMS (Charlin and Zemel, 2013), ILPR (Garg et al., 2010) and Hard algorithms. Our implementation of the PeerReview4All algorithm picks max-flow with maximum cost in Step 6 of Subroutine 1.

Previous work on the conference paper assignment problem (Garg et al., 2010; Long et al., 2013; Karimzadehgan et al., 2008; Tang et al., 2010) conducted evaluations of the proposed algorithms in terms of various objective functions that measure the quality of the assignment. For example, Garg et al. (2010) compared fairness from reviewers' perspective using the number of satisfied bids as a criteria. While these evaluations allow to compare algorithms in terms of particular objective, we note that the main goal of the peer-review

system is to accept the best papers. It is not straightforward whether an improvement of some other objective will lead to the improvement of the quality of the paper acceptance process.

In contrast to the prior works, in this section we not only consider the fairness objective (Subsections 8.2 and 8.3), but also design experiments (Subsections 8.1 and 8.4) to directly evaluate the accuracy resulting from the assignment procedures.

8.1 Synthetic Simulations

We begin with synthetic simulations. Our goal in this section is to understand easy and difficult scenarios for different algorithms. Please refer to Sections 8.2–8.4 for simulations that attempt to capture more realistic scenarios. We consider the instance of the reviewer assignment problem with $m = n = 100$ and $\lambda = \mu = 4$. We select the moderate values of m and n to keep track of the optimal assignment A^{HARD} which we find as a solution of the corresponding integer linear programming problem. For every real-valued constant c, we denote the matrix with all entries being equal to c as $\mathbf{c}$. Similarly, we denote the matrix with entries independently sampled from a Beta distribution with parameters (α, β) as $\mathcal{B}(\alpha, \beta)$.

We consider the objective-score model of reviewers (11) with $h(s) = 1 - s$ together with estimator $\widehat{\theta}^{\text{MLE}}$. Thus, assignments A^{PR4A}, A^{ILPR} and A^{HARD} aim to optimize $\Gamma^S_{(1-s)^{-1}}(A)$ while assignment A^{TPMS} aims to maximize the cumulative sum of similarities $G^S(A)$ as defined in (2).

In what follows we simulate the following problem instances:

(C1) **Non-mainstream papers.** There are $m_1 = 80$ conventional papers for which there exist $n_1 = 80$ expert reviewers with high similarity, and $m_2 = 20$ non-mainstream papers for which all the reviewers have similarity smaller than or equal to 0.5. There are also $n_2 = 20$ weak reviewers who have moderate similarities with papers from the first group and low similarities with papers from the second group. The similarities are given by the block matrix:

$$S_1 = \left[\begin{array}{c|c} \mathbf{0.9} & \mathbf{0.5} \\ \hline \underbrace{\mathbf{0.5}}_{80} & \underbrace{\mathbf{0.15}}_{20} \end{array}\right] \begin{matrix} \} 80 \\ \} 20 \end{matrix}$$

(C2) **Many weak reviewers.** In this scenario there are $n_1 = 25$ strong reviewers with high similarity with every paper and $n_2 = 75$ weak reviewers with small similarity with every paper:

$$S_2 = \left[\underbrace{\begin{array}{c} \mathbf{0.8} + 0.2 \times \mathcal{B}(1,3) \\ \hline \mathbf{0.1} + 0.2 \times \mathcal{B}(1,3) \end{array}}_{100}\right] \begin{matrix} \} 25 \\ \} 75 \end{matrix}$$

(C3) **Few super-strong reviewers.** The following example tests the algorithms in scenario when some small number of the reviewers are much stronger than the others. Similarities for this scenario are given by the block matrix:

$$S_3 = \left[\begin{array}{c|c} \mathbf{0.98} & \mathbf{0.9} \\ \hline \mathbf{0} & \mathbf{0.7} \\ \hline \underbrace{\mathbf{0.9}}_{60} & \underbrace{\mathbf{0.9}}_{40} \end{array}\right] \begin{matrix} \} 10 \\ \} 50 \\ \} 40 \end{matrix}$$

	Fairness $\Gamma^S_{(1-s)^{-1}}(A)$					Sum of Similiarities $G^S(A)$				
	C1	C2	C3	C4	C5	C1	C2	C3	C4	C5
A^{TPMS}	4.7	5.1	13.3	4.0	10.9	300	168	295	296	311
A^{HARD}	8.0	13.1	26.6	14.0	10.9	296	162	232	234	175
A^{ILPR}	8.0	5.0	4.0	14.0	10.9	296	165	188	293	296
A^{PR4A}	8.0	13.1	22.0	6.5	10.9	296	166	239	290	309

Table 2: Comparison of assignment produced by PeerReview4All, Hard, ILPR and TPMS algorithms in terms of the fairness and the sum of similarities (higher values are better).

(C4) **Adverse case.** Having analyzed the inner working of our PeerReview4All algorithm, we construct a similarity matrix which is hard for the algorithm to compute the fair assignment.[6]

(C5) **Sparse similarities.** Each entry of similarity matrix S_5 is zero with probability 0.8 or otherwise is drawn independently and uniformly at random from $[0.1, 0.9]$.

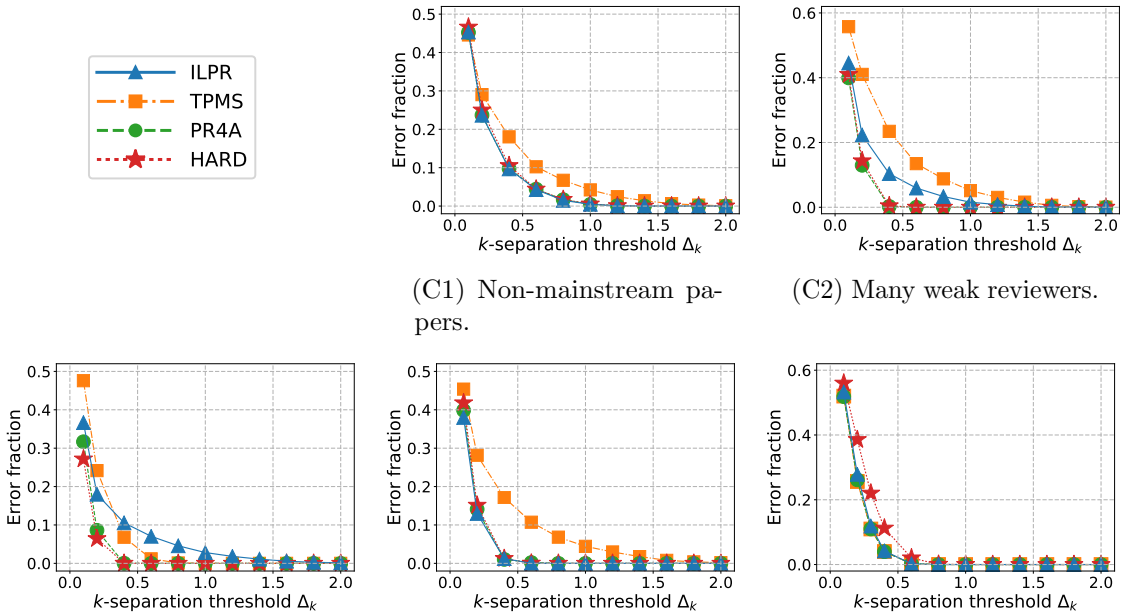
8.1.1 Fairness

In this section we analyze the quality of assignments produced by PeerReview4All, Hard, ILPR and TPMS algorithms and for all the five cases described above. The results are summarized in Table 2 where we compute the measures of fairness $\Gamma^S_{(1-s)^{-1}}(A)$ and the conventional sum of similarities $G^S(A)$ for each of the assignments.

The results in Table 2 show that in all five cases PeerReview4All algorithm finds an assignment A^{PR4A} with at least as much fairness as A^{TPMS}. At the same time, the max cost heuristic that we use in Step 6 of Subroutine 1 helps the average quality (total sum similarity) of the assignment A^{PR4A} to be either close to or larger than average quality of both A^{ILPR} and A^{HARD}.

In Case (C1), the TPMS algorithm sacrifices the quality of reviewers for non-mainstream papers, assigning them to weak reviewers. In contrast, all other algorithms assign four best possible reviewers to these unconventional papers in order to maintain fairness. In Case (C2), the PeerReview4All and Hard algorithms assign one strong reviewer for each paper while TPMS, in attempt to maximize the value of its goal function, assigns strong reviewers according to their highest similarities which leads to an unfair assignment. The ILPR algorithm fails to find a fair assignment in Cases (C2) and (C3): the poor performance of ILPR algorithm is caused by the fact that some of the reviewers in our

6. We do not give an explicit expression of the matrix S_4 for this case, due to its complicated structure. The interested reader may find the code for reconstructing this matrix in the Jupyter Notebook that accompanies our implementation of the PeerReview4All algorithm (available on the first author's website).



(C1) Non-mainstream papers.

(C2) Many weak reviewers.

(C3) Few super-strong reviewers.

(C4) Adverse case.

(C5) Sparse similarities.

Figure 2: Fraction of papers incorrectly accepted by $\widehat{\theta}^{\text{MLE}}$ based on assignments produced by PeerReview4All, Hard, ILPR and TPMS for different values of the separation threshold. Error bars are too small to be visible.

examples have similarities close to maximal, making the value of $f(s) = \frac{1}{1-s}$ large, which, in turn, makes the approximation guarantee (9) of ILPR algorithm weak. In Case (C4), the PeerReview4All algorithm was unable to recover the fair assignment. Instead, the assignment within approximation ratio 1/3, which is a bit better than the worst case $1/\lambda = 1/4$ approximation, was discovered. Finally, in Case (C5), the all algorithms managed to recover fair assignment. However, we note that the total sum similarity of the A^{HARD} assignment is low as compared to other algorithms. The reason is that the corresponding solution of the integer linear programming problem in the Hard algorithm is optimized for the fairness towards the worst-off paper and does not try to continue optimization, once the assignment for that paper is fixed. In contrast, both PeerReview4All and ILPR algorithms try to maximize the fate of the second worst-off paper, when the assignment for the most worst-off paper is fixed.

8.1.2 Statistical Accuracy

As we have pointed out, the main goal of the assignment procedure is to ensure the acceptance of the k best papers $\mathcal{T}_k^*$. While in real conferences the acceptance process is complicated and involves discussions between reviewers and/or authors, here we consider a simplified scenario. Namely, we assume an objective-score model defined in Section 6 and reviewer model (11) with $h(s) = 1 - s$.

The experiment executes 1,000 iterations of the following procedure. We randomly choose $k = 20$ indices of the "true best" papers $\mathcal{T}_k^* = \{j_1, \dots, j_k\} \subset [m]$. Each of these papers $j \in \mathcal{T}_k^*$ is assigned score $\theta_j^* = 1$, while for each of the remaining papers $j \in [m] \backslash \mathcal{T}_k^*$ we set $\theta_j^* = 1 - \Delta_k$, where $\Delta_k \in (0, 2]$. Next, given the similarity matrix S, we compute assignments A^{PR4A}, A^{HARD}, A^{ILPR} and A^{TPMS}. For each of these assignments we compute the estimations of the set of top k papers using the $\widehat{\theta}^{\text{MLE}}$ estimator and calculate the fraction of wrongly accepted papers.

For every similarity matrix $S_r, r \in [5]$, and for every value of $\Delta_k \in \{0.1k \mid k \in [20]\}$, we compute the mean of the obtained values over the 1,000 iterations. Figure 2 summarizes the dependence of the fraction of incorrectly accepted papers on the value of separation threshold Δ_k for all five cases (C1)-(C5).

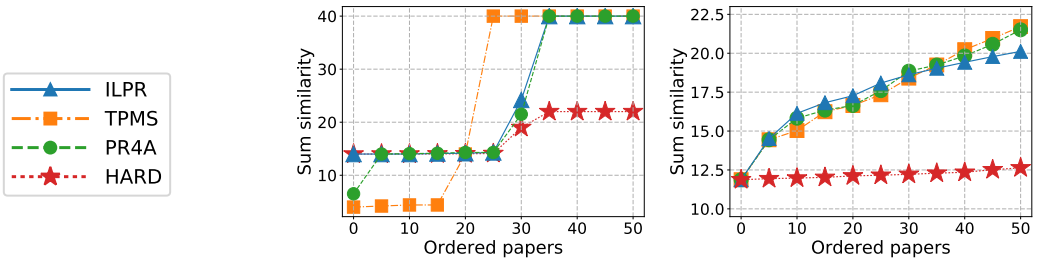
The obtained results suggest that the increase in fairness of the assignment leads to an increase in the accuracy of the acceptance procedure, provided that the average sum similarity of the assignment does not decrease dramatically. The PeerReview4All algorithm significantly outperforms TPMS both in terms of fairness and in terms of fraction of incorrectly accepted papers for the first four cases. The low fairness of assignments computed by ILPR in Cases (C2) and (C3) lead to the large fraction of errors in the acceptance procedure. As we noted earlier, the ILPR algorithm has weak approximation guarantees when the function f is allowed to be unbounded. In section 8.4 we will consider the mean score estimator ($f(s) = s$) which is more suitable scenario for ILPR algorithm.

Interestingly, in Case (C4), the PeerReview4All algorithm recovers sub-optimal assignment in terms of fairness, but still performs well in terms of the accuracy of the acceptance procedure. To understand this effect, for each of the assignments $A^{\text{TPMS}}, A^{\text{HARD}}, A^{\text{ILPR}}$ and A^{PR4A} we compute the sum similarity for all papers in the assignments and plot these values for 50 the most worst-off papers in each of the assignment in Figure 3. Despite the inability of PeerReview4All to find the fair assignment for the most worst-off paper, Corollary 2 guarantees that sum similarities for the remaining papers will not be too far from the optimal. We see this aspect in Figure 3(C4): the sum similarity for all but tiny fraction of papers in A^{PR4A} is large enough, ensuring the low fraction of incorrect decisions.

Finally, note that in Case (C5), the Hard algorithm, while having optimal fairness, has a lower accuracy as compared to other algorithms. As Figure 3(C5) demonstrates, the Hard algorithm does not optimize for the second worst off paper and recovers sub-optimal assignment for all but the most disadvantaged paper. In contrast, the ILPR and PeerReview4All algorithms do not stop their work after the most disadvantaged paper is satisfied, but instead continue to optimize the assignment for the remaining papers and eventually ensure not only fairness, but also high average quality of the assignment.

8.2 Experiment on the Approximation of ICLR Similarity Matrix

In absence of publicly available similarity matrices from conferences, we are unable to compare the assignment computed by the PeerReview4All algorithm to the actual conference assignment. To circumvent this issue, we use an approximate version of the similarity matrix from the International Conference on Learning Representations (ICLR'18) that was constructed by Xu et al. (2019b) and compare the performance of the PeerReview4All and TPMS assignment algorithms on this matrix.



(C4) Adverse case. (C5) Sparse similarities.

Figure 3: Sum similarity for the 50 most worst-off papers in assignments produced by PeerReview4All, Hard, ILPR and TPMS.

8.2.1 Matrix Construction

The similarity matrix we use for comparison was constructed by Xu et al. (2019b) as follows. OpenReview (`openreview.net`) — increasingly popular conference management system — maintains a public database of all papers (with author identities being visible) submitted to the ICLR'18 conference, thereby giving access to the pool of submissions. Next, it was assumed that all authors of submissions are simultaneously reviewers and that there are no additional reviewers. The publication profiles of reviewers were constructed by scraping the data from databases of scientific publications. Finally, the open-source code (`bitbucket.org/lcharlin/tpms/`) and the material of the original paper (Charlin and Zemel, 2013) were used to compute the similarity matrix according to the TPMS procedure.

The process outlined above resulted in the similarity matrix S that has $n = 2435$ reviewers and $m = 911$ papers. Additionally, it was assumed that any reviewer has a conflict of interests with the submitted papers that she/he has authored; these conflicts are represented by a binary matrix C whose $(i, j)^{\text{th}}$ entry equals 1 if and only if reviewer i has a conflict with paper j. Similarity matrix S possesses a considerable heterogeneity as demonstrated by some papers having mean similarity with non-conflicting reviewers almost four times larger than others.

The large size of the similarity matrix makes computation of the optimally fair assignment infeasible, and hence in this section we do not compute the Hard assignment. Additionally, our implementation of the ILPR assignment algorithms was computationally inefficient and in absence of the publicly available source code we also exclude this algorithm from comparison.

8.2.2 Evaluation

Having defined the similarity matrix and matrix of conflicts, we compute assignments of papers to reviewers with $\lambda = 4$ (each paper is assigned to 4 reviewers) and $\mu = 2$ (each reviewer is allocated at most 2 papers) using the TPMS and PeerReview4All assignment algorithms with the identity transformation function $f(s) = s$. In addition to the standard load constraints, we require that no paper is assigned to a conflicting reviewer. Finally, as pointed out in the comment on the early stopping guarantees made after Theorem 1, the

Algorithm	Fairness $\Gamma^S(A)$	Mean sum of sim. $\frac{1}{m}G^S(A)$
A^{TPMS}	0.12	0.413
A_1^{PR4A} (one iteration)	0.15 (+25%)	0.408 (−1%)
A^{PR4A} (full)	0.15 (+25%)	0.406 (−2%)

Table 3: Results of the experiment on the approximation of ICLR'18 similarity matrix. Values in brackets represent relative changes as compared to the TPMS assignment.

fairness guarantees of Theorem 1 are achieved after the first iteration of Steps 2 to 7 of Algorithm 1. Hence, we include the corresponding assignment for comparison and denote it as A_1^{PR4A}.

Table 3 summarizes the results of the experiment, comparing the resulting assignments in terms of fairness (3) and cumulative similarity (2). We see that the fairness of the assignment computed by the PeerReview4All algorithm is significantly higher than the fairness of the TPMS algorithm. Similar to the case of synthetic simulations, the max cost heuristic used in Step 6 of Subroutine 1 helps our algorithm to maintain a high value of cumulative similarity, which is only marginally below the optimal value.

The large size of the similarity matrix at hands makes evaluation of the optimal fairness achieved by A^{HARD} computationally prohibitive. However, we can still find an upper bound on $\Gamma^S\left(A^{\text{HARD}}\right)$ by dropping reviewer load constraints and allowing all reviewers to review unlimited number of papers. The resulting bound allows us to compute a lower bound on the approximation ratio of the PeerReview4All algorithm:

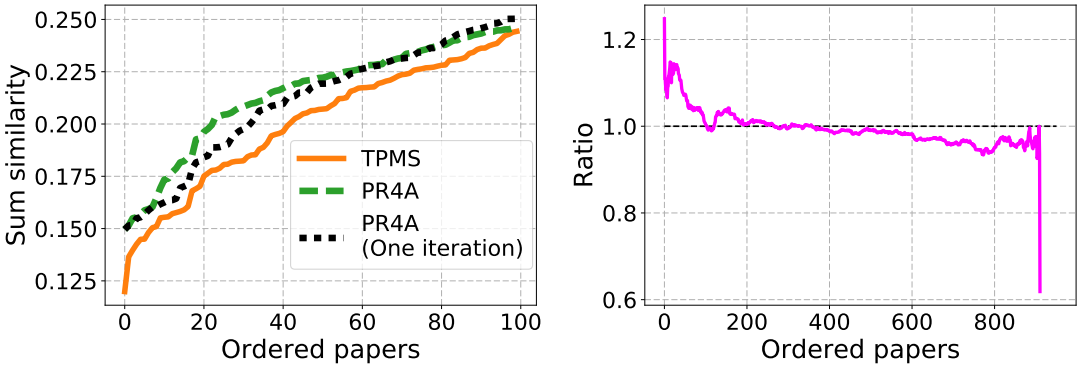
$$\frac{\Gamma^S\left(A^{\text{PR4A}}\right)}{\Gamma^S\left(A^{\text{HARD}}\right)} \geq 0.98,$$

which shows that in practice the approximation factor of the PeerReview4All algorithm can be much better than the worst-case approximation factor $\frac{1}{\lambda}$ guaranteed by Theorem 1.

Continuing the analysis, for each of the assignments A^{TPMS}, A_1^{PR4A} and A^{PR4A} we compute the sum similarity for all papers in the assignments and plot these values for 100 the most worst-off papers in each of the assignment in Figure 4a. This figure demonstrates that while the fairness guarantees of Theorem 1 can be achieved by a single iteration of Steps 2 to 7, subsequent iterations help to improve the assignment for the second worst-off paper and so on. Finally, for each of the assignments A^{TPMS} and A^{PR4A} we sort papers in order of increasing sum similarity of assigned reviewers and plot the ratios (PeerReview4All to TPMS) of these sums in Figure 4b. Figure 4b shows that the PeerReview4All algorithm indeed balances the assignment by improving the quality for the worst-off papers at the expense of decreasing the quality for the most benefiting papers.

8.3 Experiment on MIDL and CVPR Similarity Matrices

Subsequent to the publication of the first version of this paper (Stelmakh et al., 2019b), a follow-up paper by Kobren et al. (2019) has been published. There authors propose two novel assignment algorithms that also aim at ensuring the fairness of the assignment. In



(a) Sum similarity for 100 most disadvantaged papers in each assignment.

(b) Ratio of ordered sum similarities in A^{PR4A} to ordered sum similarities in A^{TPMS}.

Figure 4: Comparison on the approximation of ICLR'18 similarity matrix.

that work, the PeerReview4All algorithm with the identity transformation function ($f(s) = s$) was compared with other assignment algorithms on similarity matrices from three real conferences: Medical Imaging and Deep Learning Conference (MIDL'18), and two editions of the Conference on Computer Vision and Pattern Recognition (CVPR'17 and CVPR'18). With the kind permission of Ari Kobren, we describe the results of their experiments in which our algorithm was evaluated.

8.3.1 Brief Discussion of the Algorithms by Kobren et al.

We begin with a brief theoretical comparison of the PeerReview4All algorithm with the algorithms proposed by Kobren et al. (2019). Recall that the PeerReview4All algorithm aims at optimizing fairness of the assignment (3) and does not directly optimize for the total sum similarity. However, when in its inner workings the algorithm faces a choice between different suitable similarity matrices (Step 6 of the Subroutine 1), it can heuristically optimize for the total sum similarity by using the max cost heuristic. In contrast, Kobren et al. (2019) consider a problem of optimizing for the total sum similarity of the assignment *with an additional constraint of each paper having the sum similarity larger than some threshold* T, which can be specified by user or found by the binary search. They design two novel algorithms which we refer to as FairIr and FairFlow.

Given a feasible instance of the reviewer assignment problem, the FairIr algorithm is able to compute the assignment with the optimal value of the total sum similarity, violating the fairness constraints by an additive factor which is upper bounded by the maximum entry of the similarity matrix. In that, fairness guarantees of FairIr are equivalent to those of ILPR (and hence may become vacuous when similarity matrix is significantly heterogeneous), but additionally the FairIr algorithm achieves the highest possible value of sum similarity.[7] The FairFlow algorithm is a heuristic which does not have theoretical guarantees, but in return has much lower computational complexity.

7. Observe that this value is lower than those achieved by TPMS as FairIr has additional constraint on the fairness of the assignment.

Conference	Parameters	Algorithm	Time (s)	Fairness	Mean sum of sim.
MIDL'18	$n = 177,\ \mu = 4$ $m = 118,\ \lambda = 3$	A^{TPMS}	0.1	0.90	1.71
		A^{PR4A}	293.8	0.92	1.67
		A^{FairIr}	1.6	0.93	1.71
		A^{FairFlow}	1.2	0.94	1.68
CVPR'17	$n = 1373,\ \mu = 6$ $m = 2623,\ \lambda = 3$	A^{TPMS}	47	0	2.08
		A_1^{PR4A}	3251	0.77	1.96
		A^{FairIr}	595	0.27	2.05
		A^{FairFlow}	225	0.77	1.69
CVPR'18	$n = 2840,\ \mu = 9$ $m = 5062,\ \lambda = 3$	A^{TPMS}	257	1.37	22.23
		A_1^{PR4A}	8684	12.68	21.48
		A^{FairIr}	3786	7.19	22.18
		A^{FairFlow}	910	11.12	17.98

Table 4: Results of the experiment conducted by Kobren et al. on similarity matrices from real conferences. On large instances only a single iteration of the PeerReview4All algorithm was computed and the corresponding assignment is denoted A_1^{PR4A}.

Another difference between PeerReview4All and the algorithms proposed by Kobren et al. (2019) is that both FairIr and FairFlow allow to specify a *lower bound on reviewer load*, thereby ensuring that each reviewer reviews at least some number of papers. In our work, we do not study such constraints and PeerReview4All does not support such constraints as is. Hence, below we report only those comparisons in which our algorithm was evaluated by Kobren et al. (2019), that is, the comparisons in which the lower bound on reviewer load was not enforced.

Overall, the FairIr and FairFlow algorithms aim at balancing the fairness and the total sum similarity of the assignment. By choosing an appropriate heuristic in Step 6 of the Subroutine 1, one can ensure that PeerReview4All also heuristically optimizes for the total sum similarity. Let us now report the experimental results of Kobren et al. (2019) that allows to compare the algorithms on both objectives of fairness and total sum similarity.

8.3.2 Summary of the Experiments

The key summary statics of the Kobren et al. (2019) experiments are represented in Table 4.[8] For each similarity matrix, the assignments respecting the corresponding paper and reviewer load constraints were computed by the TPMS, PeerReview4All, FairIr and FairFlow algorithms. These assignments were then compared based on (a) running time of the algorithm, (b) fairness of the assignment and (c) mean sum similarity of the assignment. First, we notice that our naive implementation of the PeerReview4All algorithm is significantly slower than all other algorithms, and for large instances only a single iteration

8. We omit some statistics which are not of direct interest (for example, max sum similarity in the assignment).

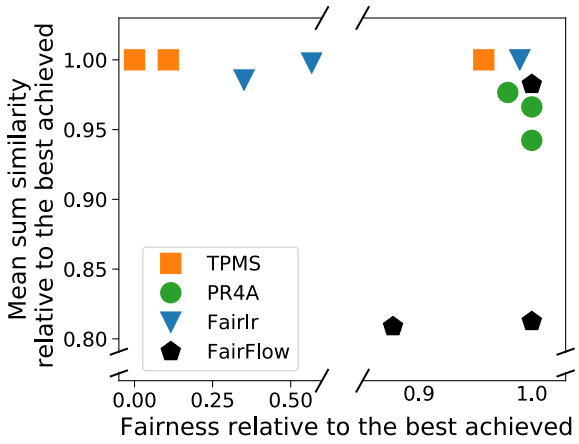


Figure 5: Visualization of comparison of the algorithms based on fairness and total sum similarity. Each point corresponds to a (conference, algorithm) pair and the closeness to the top-right corner indicates superior performance.

of the algorithm can be computed in a reasonable time (recall that even one iteration is sufficient to satisfy the fairness guarantees of Theorem 1). Nonetheless, even on the largest instance with more than 5,000 papers the running time of the first iteration of our algorithm took less than three hours which is still feasible given that the full assignment procedure needs to be run only once in the conference timeline.

The remaining two dimensions of comparison represent two notions of quality of the assignment: fairness and total sum similarity. Ideally, we would like to have an algorithm which simultaneously optimizes both of these notions. Figure 5 visualizes the comparison of the algorithms and is constructed as follows. For each of the three experiments, we compute the maximum value of fairness achieved by any of the algorithms. Using this value, for each algorithm we compute its "competetiveness" as the fairness achieved by that algorithm divided by the maximum fairness. We then repeat the same for the total sum similarity. As a result, in each experiment the performance of each algorithm can be represented as a data point in two-dimensional space where x-axis represents the competitiveness in terms of fairness and y-axis represents the competitiveness in terms of the total sum similarity.

Figure 5 demonstrates that in each of the three experiments the PeerReview4All algorithm (even with one iteration) was able to achieve maximum or close-to-maximum values of both fairness and total sum similarity. In contrast, each of the other algorithms under consideration achieved considerably lower value of either fairness or total sum similarity in two out of three experiments.

Overall, we conclude that while being considerably (but not prohibitively) slower than other algorithms, PeerReview4All managed to achieve the best balance of fairness and total sum similarity, despite optimizing the latter objective only heuristically.

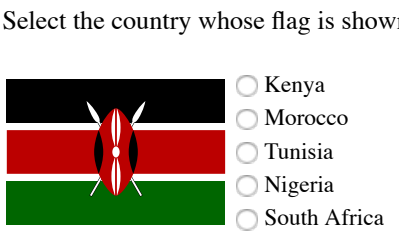


Figure 6: Question interface

8.4 Experiment on Amazon Mechanical Turk

Even if peer-review data from conferences was available to us, it would not allow for an objective evaluation of any assignment algorithm with respect to accuracy of the acceptance procedure. There are two reasons for this hinderance: (a) No ground truth ranking is available; and (b) The data contains only reviews that correspond to one particular assignment and has missing reviews for other assignments.

In this section we present an experiment which we carefully design to overcome the fundamental issues with objective empirical evaluations of reviewer assignments. Our experiment allows us to directly measure the accuracy of final decisions to evaluate any assignment. We execute our experiment on the Amazon Mechanical Turk (`mturk.com`) crowdsourcing platform.

8.4.1 Design of Experiment

We designed the experiment in a manner that allows us to objectively evaluate the performance of any assignment algorithm. Specifically, the experiment should provide us access to some similarities between reviewers and papers, execute any assignment algorithm, and eventually objectively evaluate the final outcome.

The experiment considers crowdsourcing workers as reviewers and a number of general knowledge questions as papers. Specifically, 80 workers were recruited and presented with a list of 60 flags of different countries. The workers were asked to determine the country of each flag, choosing one of five options for each question. The interface of the task is represented in Figure 6. Unknown to the worker, the 60 countries comprised 10 countries each from 6 different geographic regions. Three participants did not attempt some of the questions and their responses were discarded from the data set. The data set is available on the first author's website.

8.4.2 Evaluation

After obtaining the data from Amazon Mechanical Turk, we executed the following procedure for 1,000 iterations. In each of the 6 regions, we first split the 10 questions into two sets: a "gold standard" set of 8 questions chosen uniformly at random and an "unresolved" set comprising the 2 remaining questions. The set of all 12 unresolved questions are analogous to papers in the peer-review setting ($m = 12$). We computed the similarity of any worker to any paper (question) as the fraction of questions that the worker answered correctly

Algorithm	Error fraction	Error increase	Fairness	Sum of sim.
A^{RAND}	0.394	+275%	6.4	171.1
A^{TPMS}	0.113	+8%	20.8	274.6
A^{HARD}	0.110	+5%	21.9	269.8
A^{ILPR}	0.108	+3%	21.7	270.4
A^{PR4A}	0.105	—	21.6	272.9

Table 5: Results of the experiment on Amazon Mechanical Turk. Fairness $\Gamma^S(A)$ and sum of similarities $G^S(A)$ are averaged over 1,000 iterations.

among the 8 gold standard questions for the region corresponding to that paper (question). Having computed the similarities, we selected $n = 40$ of the workers uniformly at random and created five assignments $A^{\text{TPMS}}, A^{\text{PR4A}}$, A^{ILPR}, A^{HARD} and A^{RAND}, with identity transformation function $f(s) = s$, where A^{RAND} is a random feasible assignment. In each of these assignments, every question was answered by $\lambda = 3$ workers and every worker answered at most $\mu = 2$ questions. Finally, for each assignment, we computed the answers for the remaining $m = 12$ questions by taking a majority vote of the responses from workers assigned to each question. Ties are also considered as mistakes.

At the end of all iterations, we computed the fraction of questions whose final answers are estimated incorrectly under the five assignments as well as the mean fairness $\Gamma^S(A)$ and conventional sum of similarities $G^S(A)$. We summarize the results in Table 5. We see that all non-trivial algorithms significantly outperform random assignment. However, A^{TPMS} incurs about 8% increased error as compared to A^{PR4A}.

Similar to Case (C5) of synthetic experiments, the optimally fair assignment A^{HARD} turns out to incur larger fraction of errors as compared to approximations A^{PR4A} and A^{ILPR}. The reason is that the assignment A^{HARD} maximizes the quality of the assignment with respect to the most "disadvantaged" question, but in contrast to A^{PR4A} and A^{ILPR}, does not care about the fate of remaining questions.

We also see that A^{PR4A} slightly outperforms A^{ILPR} in terms of the fraction of errors while having slightly smaller average fairness. One reason for this is that in parallel with $\Gamma^S\left(A^{\text{PR4A}}\right)$ being close to optimal, PeerReview4All algorithm managed to achieve the high value of conventional sum of similarities, thus maintaining a balance between the fairness $\Gamma^S(A)$ and the global objective $G^S(A)$.

We find these observations to be of notable interest for the actual conference peer-review scenarios. The task of identifying flags in the experiment involved a rather homogeneous set of similarities (in the sense that each worker either knew many or only few flags) where optimizing (2) or (3) would yield similar results. In contrast, the significantly higher heterogeneity in peer-review, the presence of many non-mainstream papers as well as both very strong and very weak reviewers, is expected to further amplify the observed improvements offered by the PeerReview4All algorithm as compared to TPMS and ILPR.

9. Proofs

We now present the proofs of our main results.

9.1 Proof of Theorem 1

We prove the result in three steps. First, we establish a lower bound on the fairness of the PeerReview4All algorithm. Then we establish an upper bound on the fairness of the optimal assignment. Finally, we combine these bounds to obtain the result (7).

9.1.1 Lower Bound for the PeerReview4All Algorithm.

We show a lower bound for the intermediate assignment $\widetilde{A}$ at Step 3 during the first iteration of Steps 2 to 7. We denote this particular assignment as $\widetilde{A}_1$. Note that in Step 4 we fix the assignment for $\widetilde{A}_1$'s worst-off papers into the final output, and hence we have $\Gamma_f^S\left(\widetilde{A}_1\right) \geq \Gamma_f^S\left(A_f^{\text{PR4A}}\right)$. On the other hand, by keeping track of A_0 (Step 7), we ensure that in all of the subsequent iterations of Steps 2 to 7, the temporary assignment $\widetilde{A}$ will be at least as fair as $\widetilde{A}_1$, which implies $\Gamma_f^S\left(\widetilde{A}_1\right) = \Gamma_f^S\left(A_f^{\text{PR4A}}\right)$.

Getting back to the first iteration of Steps 2 to 7, we note that when Step 2 is completed, we have λ assignments $A_1, \ldots, A_\lambda$ as candidates. Notice that for every $\kappa \in [\lambda]$, assignment A_κ is constructed with a two-step procedure by joining the outputs A_κ^1 and A_κ^2 of Subroutine 1. Recalling the definition (6) of s_κ^*, we now show that for every value of $\kappa \in [\lambda]$, the assignment A_κ^1 satisfies:

$$\min_{j\in[m]} \min_{i\in\mathcal{R}_{A_\kappa^1}(j)} s_{ij} = s_\kappa^*.$$

Consider any value of $\kappa \in [\lambda]$. The definition of s_κ^* ensures that there exist an assignment, say A^*, which assigns κ reviewers to each paper in a way that minimum similarity in this assignment equals s_κ^*. Now note that Subroutine 1, called in Step 2b of the algorithm, adds edges to the flow network in order of decreasing similarities. Thus, at the time all edges with similarity higher or equal to s_κ^* are added, we have that no edges with similarity smaller that s_κ^* are added, and that all edges which correspond to the assignment A^* are also added to the network. Thus, a maximum flow of size $m\kappa$ is achieved and hence each assigned (reviewer, paper) pair has similarity at least s_κ^*.

Recalling that s_∞^* is the lowest similarity in similarity matrix S, one can deduce that $\Gamma_f^S\left(A_\kappa\right) \geq \kappa f(s_\kappa^*) + (\lambda - \kappa)\, f(s_\infty^*)$ due to the monotonicity of f. Consequently, we have

$$\Gamma_f^S\left(A_f^{\text{PR4A}}\right) \geq \Gamma_f^S\left(A_\kappa\right) \geq \kappa f(s_\kappa^*) + (\lambda - \kappa)\, f(s_\infty^*), \tag{23}$$

for all $\kappa \in [\lambda]$. Taking a maximum over all values of $\kappa \in [\lambda]$ concludes the proof.

9.1.2 Upper Bound for the Optimal Assignment A_f^{HARD}.

Consider any value of $\kappa \in [\lambda]$. By definition (6) of s_κ^*, for any feasible assignment $A \in \mathcal{A}$, there exists some paper $j_\kappa^* \in [m]$ for which at most $(\kappa - 1)$ reviewers have similarity strictly greater than s_κ^*. Let us now consider assignment A_f^{HARD} and corresponding paper j_κ^*. This paper is assigned to at most $(\kappa - 1)$ reviewers with similarity greater than s_κ^* and to at least

$(\lambda - \kappa + 1)$ reviewers with similarity smaller or equal to s^*_κ. Recalling that s^*_0 is the largest possible similarity, we conclude that due to monotonicity of f, the following upper bound holds:

$$\Gamma^S_f\left(A^{\text{HARD}}_f\right) = \min_{j\in[m]} \sum_{i\in\mathcal{R}_{A^{\text{HARD}}_f}(j)} f(s_{ij}) \le \sum_{i\in\mathcal{R}_{A^{\text{HARD}}_f}(j^*_\kappa)} f(s_{ij^*_\kappa}) \le (\kappa-1)\, f(s^*_0) + (\lambda-\kappa+1)\, f(s^*_\kappa). \quad (24)$$

Taking a minimum over all values of $\kappa \in [\lambda]$, then yields an upper bound on the fairness of A^{HARD}_f.

9.1.3 Putting it Together.

To conclude the argument, it remains to plug in the obtained bounds (23) and (24) into ratio $\frac{\Gamma^S_f\left(A^{\text{PR4A}}_f\right)}{\Gamma^S_f\left(A^{\text{HARD}}_f\right)}$:

$$\frac{\Gamma^S_f\left(A^{\text{PR4A}}_f\right)}{\Gamma^S_f\left(A^{\text{HARD}}_f\right)} \ge \frac{\max\limits_{\kappa\in[\lambda]}\left(\kappa f(s^*_\kappa) + (\lambda-\kappa)\, f(s^*_\infty)\right)}{\min\limits_{\kappa\in[\lambda]}\left((\kappa-1) f(s^*_0) + (\lambda-\kappa+1)\, f(s^*_\kappa)\right)}.$$

Setting $\kappa = 1$ in both numerator and denominator and recalling that $f(s) \ge 0\ \forall s \in [0,1]$, we obtain a worst-case approximation in terms of required paper load: $\frac{\Gamma^S\left(A^{\text{PR4A}}\right)}{\Gamma^S(A^{\text{HARD}})} \ge \frac{1}{\lambda}$.

9.2 Proof of Corollary 2

Let us pause the PeerReview4All algorithm at the beginning of the r^{th} iteration of Steps 2 to 7 and inspect its state.

- The set $\mathcal{M}$ consists of papers that are not yet assigned:
$$\mathcal{M} = [m] \setminus \left(\bigcup_{l=1}^{r-1} \mathcal{J}_l\right).$$
- The vector of reviewers' loads $\overline{\mu}$ is adjusted with respect to assigned papers. For every reviewer $i \in [n]$, we have:
$$\overline{\mu}_i = \mu - \text{card}\left(\left\{ j \in \bigcup_{l=1}^{r-1} \mathcal{J}_l \,\middle|\, i \in \mathcal{R}_{A^{\text{PR4A}}_f}(j)\right\}\right).$$
- The similarity matrix S_r consists of columns of the initial similarity matrix S which correspond to papers in $\mathcal{M}$.

The only thing that connects the algorithm with the previous iterations is the assignment A_0, computed in Step 7 of the previous iteration. However, we note that the sum similarity for the worst-off papers, determined in Step 4 of the current iteration (in other words, fairness of $\widetilde{A}_r$), is lower-bounded by the largest fairness of the candidate assignments $A_1, \dots, A_\lambda$, which are computed in Step 2.

We now repeat the proof of Theorem 1 with the following changes. Instead of the similarity matrix S, we use the updated matrix S_r; instead of considering all papers m we consider only papers from $\mathcal{M}$; instead of assuming that each reviewer $i \in [n]$ can review at most μ papers, we allow reviewer $i \in [n]$ to review at most $\overline{\mu}_i$ papers. Hence, we arrive to the bound (7) on the fairness of $\widetilde{A}_r$, where A^{HARD} should be read as $A^{\mathrm{HARD}}(\mathcal{M}) = A^{\mathrm{HARD}}\left(\mathcal{J}_{\{r:p\}}\right)$ and values $s^*_\kappa, \kappa \in \{0, \ldots, \lambda\} \cup \{\infty\}$ are computed for similarity matrix S_r and constraints on reviewers' loads $\overline{\mu}$. Thus, we obtain (8) and conclude the proof of the corollary.

9.3 Proof of Theorem 3

Before we prove the theorem, let us formulate an auxiliary lemma which will help us show the claimed upper bound. We give the proof of this lemma subsequently in Section 9.3.3.

Lemma 9 *Consider any valid assignment $A \in \mathcal{A}$ and any estimator $\widehat{\theta} \in \left\{\widehat{\theta}^{MLE}, \widehat{\theta}^{MEAN}\right\}$. Then for every $\delta > 0$, the error incurred by $\widehat{\theta}$ is upper bounded as*

$$\sup_{\left(\theta^*_1,\ldots,\theta^*_m\right)\in\mathcal{F}_k(\delta)} \mathbb{P}\left\{\mathcal{T}_k\left(A,\widehat{\theta}\right) \neq \mathcal{T}^*_k\right\} \leq k(m-k)\exp\left\{-\left(\frac{\delta}{2\widetilde{\sigma}(A,\widehat{\theta})}\right)^2\right\},$$

where

$$\widetilde{\sigma}^2(A,\widehat{\theta}) = \begin{cases} \max\limits_{j\in[m]}\left(\sum\limits_{i\in\mathcal{R}_A(j)}\frac{1}{\sigma^2_{ij}}\right)^{-1} & \text{if } \widehat{\theta} = \widehat{\theta}^{MLE} \\ \max\limits_{j\in[m]}\left(\frac{1}{\lambda^2}\sum\limits_{i\in\mathcal{R}_A(j)}\sigma^2_{ij}\right) & \text{if } \widehat{\theta} = \widehat{\theta}^{MEAN}. \end{cases}$$

9.3.1 Proof of Upper Bound

First, recall from (13) the distribution of $\widehat{\theta}^{\mathrm{MEAN}}_j, j \in [m]$. Then the PeerReview4All algorithm called with $f = 1-h$ simultaneously tries to maximize the fairness of the assignment with respect to f and minimize the maximum variance of the estimated scores $\widehat{\theta}^{\mathrm{MEAN}}_j, j \in [m]$. Similarly, the choice of $f = h^{-1}$ ensures that together with optimizing the corresponding fairness, the algorithm also minimizes the maximum variance of $\widehat{\theta}^{\mathrm{MLE}}_j, j \in [m]$, defined in (12). Thus, the choice of the estimator defines the choice of the transformation function f which minimizes the maximum variance of the estimated scores. To maintain brevity, we denote $A_{\mathrm{MEAN}} = A^{\mathrm{PR4A}}_{1-h}$, $A_{\mathrm{MLE}} = A^{\mathrm{PR4A}}_{h^{-1}}$, $A_{\mathrm{MEAN}}(j) = \mathcal{R}_{A_{\mathrm{MEAN}}}(j)$ and $A_{\mathrm{MLE}}(j) = \mathcal{R}_{A_{\mathrm{MLE}}}(j)$.

Let now $S \in \mathcal{S}(q)$. We begin with the pair of assignment and estimator $\left(A_{\mathrm{MEAN}}, \widehat{\theta}^{\mathrm{MEAN}}\right)$. Notice that for arbitrary feasible assignment $A \in \mathcal{A}$ and estimator $\widehat{\theta}^{\mathrm{MEAN}}$,

$$\begin{aligned}\widetilde{\sigma}^2(A,\widehat{\theta}^{\mathrm{MEAN}}) &= \max_{j\in[m]}\left(\frac{1}{\lambda^2}\sum_{i\in\mathcal{R}_A(j)}\sigma^2_{ij}\right) = \frac{1}{\lambda^2}\max_{j\in[m]}\left(\sum_{i\in\mathcal{R}_A(j)} 1-(1-h(s_{ij}))\right) \\ &= \frac{1}{\lambda^2}\left(\lambda - \min_{j\in[m]}\sum_{i\in\mathcal{R}_A(j)}(1-h(s_{ij}))\right) = \frac{1}{\lambda^2}\left(\lambda - \Gamma^S_{1-h}(A)\right).\end{aligned}$$

Now we can write

$$\begin{aligned}\sup_{S\in\mathcal{S}(q)} \widetilde{\sigma}^2(A_{\mathrm{MEAN}}, \widehat{\theta}^{\mathrm{MEAN}}) &= \frac{1}{\lambda^2}\left(\lambda - q \inf_{S\in\mathcal{S}(q)} \frac{\Gamma^S_{1-h}(A_{\mathrm{MEAN}})}{q}\right)\\ &\le \frac{1}{\lambda^2}\left(\lambda - q \inf_{S\in\mathcal{S}(q)} \frac{\Gamma^S_{1-h}(A_{\mathrm{MEAN}})}{\Gamma^S_{1-h}\left(A^{\mathrm{HARD}}_{1-h}\right)}\right)\\ &= \frac{\lambda - q\tau_q}{\lambda^2}.\end{aligned}$$

Using Lemma 9, we conclude the proof for the mean score estimator:

$$\begin{aligned}&\sup_{\substack{(\theta^*_1,\dots,\theta^*_m)\in\mathcal{F}_k(\delta)\\ S\in\mathcal{S}(q)}} \mathbb{P}\left\{\mathcal{T}_k\left(A_{\mathrm{MEAN}}, \widehat{\theta}^{\mathrm{MEAN}}\right) \ne \mathcal{T}^*_k\right\}\\ &\qquad\le k(m-k)\exp\left\{-\left(\frac{\delta}{2\sup\limits_{S\in\mathcal{S}(q)} \widetilde{\sigma}(A_{\mathrm{MEAN}}, \widehat{\theta}^{\mathrm{MEAN}})}\right)^2\right\} \qquad (25a)\\ &\qquad\le m^2\exp\left\{-\frac{\lambda^2\delta^2}{4(\lambda - q\tau_q)}\right\} \le m^2\exp\left\{-\ln\frac{m^2}{\epsilon}\right\} \le \epsilon. \qquad (25b)\end{aligned}$$

Let us now consider the pair $\left(A_{\mathrm{MLE}}, \widehat{\theta}^{\mathrm{MLE}}\right)$. It suffices to show that

$$\sup_{S\in\mathcal{S}(q)} \widetilde{\sigma}^2(A_{\mathrm{MLE}}, \widehat{\theta}^{\mathrm{MLE}}) \le \sup_{S\in\mathcal{S}(q)} \widetilde{\sigma}^2(A_{\mathrm{MEAN}}, \widehat{\theta}^{\mathrm{MEAN}}). \qquad (26)$$

Let us consider $S \in \mathcal{S}(q)$. Recall from the proof of Theorem 1 that the fairness of the resulting assignment is determined in the first iteration of Steps 2 to 7. After completion of Step 2, we have λ candidate assignments $A_1, \dots, A_\lambda$. Observe that Subroutine 1 in Step 6 uses the same heuristic for both A_{MEAN} and A_{MLE}. Hence, the λ candidate assignments yielded when PeerReview4All constructs A_{MEAN} coincide with the candidate assignments yielded when PeerReview4All constructs A_{MLE}. Depending on the choice of f, in Step 3 the algorithm picks one assignment that maximizes fairness (4) with respect to f. Thus,

$$\Gamma^S_{1-h}(A_{\mathrm{MEAN}}) = \max_{\kappa\in[\lambda]} \Gamma^S_{1-h}(A_\kappa) \quad \text{and} \quad \Gamma^S_{h-1}(A_{\mathrm{MLE}}) = \max_{\kappa\in[\lambda]} \Gamma^S_{h-1}(A_\kappa). \qquad (27)$$

Hence, we have

$$\begin{aligned}\widetilde{\sigma}^2(A_{\mathrm{MLE}}, \widehat{\theta}^{\mathrm{MLE}}) &= \max_{j\in[m]}\left(\sum_{i\in A_{\mathrm{MLE}}(j)} \frac{1}{\sigma^2_{ij}}\right)^{-1} = \max_{j\in[m]}\left(\frac{1}{\sum\limits_{i\in A_{\mathrm{MLE}}(j)} \frac{1}{h(s_{ij})}}\right)\\ &= \frac{1}{\Gamma^S_{h-1}(A_{\mathrm{MLE}})} \le \frac{1}{\Gamma^S_{h-1}(A_{\mathrm{MEAN}})}.\end{aligned}$$

where the last ineqaulity is due to (27). Recalling the definition of the fairness (4) and using Jensen's inequality, we continue:

$$\widetilde{\sigma}^2(A_{\mathrm{MLE}}, \widehat{\theta}^{\mathrm{MLE}}) \le \max_{j\in[m]} \left(\frac{1}{\lambda^2} \sum_{i\in A_{\mathrm{MEAN}}(j)} h(s_{ij}) \right) = \max_{j\in[m]} \left(\frac{\lambda - \sum\limits_{i\in A_{\mathrm{MEAN}}(j)} (1-h(s_{ij}))}{\lambda^2} \right)$$
$$= \frac{\lambda - \Gamma^S_{1-h}(A_{\mathrm{MEAN}})}{\lambda^2} = \widetilde{\sigma}^2(A_{\mathrm{MEAN}}, \widehat{\theta}^{\mathrm{MEAN}}).$$

Taking a supremum over all $S \in \mathcal{S}(q)$, we obtain (26) which together with Lemma 9 and the first part of the statement concludes the proof.

9.3.2 Proof of Lower Bound

Proof of our lower bound is based on Fano's inequality (Cover and Thomas, 2005) which provides a lower bound for probability of error in L-ary hypothesis testing problems.

Without loss of generality we assume that $k \le \frac{1}{2}m$. Otherwise, the result will hold by symmetry of the problems.

We first claim that there exists a value $s \in [0,1]$ such that $h(s) = 1 - \frac{q}{\lambda}$. Indeed, by assumptions of the theorem, h is continuous strictly monotonically decreasing function and $\frac{q}{\lambda} \ge 1 - h(0)$. Thus, $h(0) \ge 1 - \frac{q}{\lambda}$. On the other hand, if $h(1) > 1 - \frac{q}{\lambda}$, then for every similarity matrix S we have

$$\Gamma^S_{1-h}(A) \le \lambda(1-h(1)) < q.$$

The last inequality contradicts with the definition (16) of $\mathcal{S}(q)$, verifying that

$$h(0) \ge 1 - \frac{q}{\lambda} \ge h(1).$$

Given that h is continuous strictly monotonically decreasing function, we conclude that these exists $s = h^{-1}\left(1 - \frac{q}{\lambda}\right) \in [0,1]$.

Consider the similarity matrix $\widetilde{S} = \left\{h^{-1}\left(1-\frac{q}{\lambda}\right)\right\}^{n\times m}$. Observe that $\widetilde{S} \in \mathcal{S}(q)$, since every feasible assignment $A \in \mathcal{A}$ has fairness

$$\Gamma^{\widetilde{S}}_{1-h}(A) = \min_{j\in[m]} \sum_{i\in\mathcal{R}_A(j)} (1-h(s_{ij})) = \min_{j\in[m]} \sum_{i\in\mathcal{R}_A(j)} \left\{1 - h\left(h^{-1}\left(1-\frac{q}{\lambda}\right)\right)\right\} = q.$$

Thus, in any feasible assignment each paper $j \in [m]$ receives λ reviewers with similarity exactly $h^{-1}\left(1-\frac{q}{\lambda}\right)$.

To apply Fano's inequality, we need to reduce our problem to a hypothesis testing problem. To do so, let us introduce the set $\mathcal{P}$ of $(m-k+1)$ instances of the paper accepting/rejecting problem: every problem instance in this set has the same similarity matrix $\widetilde{S}$, but differs in the set of top k papers $\mathcal{T}^*_k$. We now consider the problem of distinguishing between these problem instances, which is equivalent to the problem of correctly recovering the top k papers. More concretely, we denote the $(m-k+1)$ problem instances as, $\mathcal{P} = \{1, 2, \ldots, m-k+1\}$, where for any problem $\ell \in \mathcal{P}$ the set of top k papers

is denoted as $\mathcal{T}_k^*(\ell)$ and set as $\{1, 2, \ldots, k-1\} \cup \{k-1+\ell\}$. The true quality of any paper $j \in [m]$ in any problem instance $\ell \in \mathcal{P}$ is

$$\theta_j^*(\ell) = \begin{cases} \delta & \text{if} \quad j \in \mathcal{T}_k^*(\ell) \\ 0 & \text{otherwise,} \end{cases}$$

thereby ensuring that $(\theta_1^*(\ell), \ldots, \theta_m^*(\ell)) \in \mathcal{F}_k(\delta)$, for every instance $\ell \in \mathcal{P}$.

Let P denote a random variable which is uniformly distributed over elements of $\mathcal{P}$. Then given $P = \ell$, we denote a random matrix of reviewers' scores as $Y^{(\ell)} \in \mathbb{R}^{\lambda \times m}$ whose $(r,j)^{\text{th}}$ entry is a score given by reviewer $i_r, r \in [\lambda]$, assigned to paper j and

$$Y_{rj}^{(\ell)} \sim \begin{cases} \mathcal{N}\left(\delta, 1 - \frac{q}{\lambda}\right) & \text{if} \quad j \in \mathcal{T}_k^*(\ell) \\ \mathcal{N}\left(0, 1 - \frac{q}{\lambda}\right) & \text{otherwise.} \end{cases} \tag{28}$$

We denote the distribution of random matrix $Y^{(\ell)}$ as $\mathbb{P}^{(\ell)}$. Note that $Y^{(\ell)}$ does not depend on the selected assignment $A \in \mathcal{A}$. Indeed, recall from (11), that assignment A affects only variances of observed scores. On the other hand, for any reviewer $i \in [n]$ and for any paper $j \in [m]$, the score y_{ij} has variance $1 - \frac{q}{\lambda}$. Thus, for any feasible assignment A and any $\ell \in \mathcal{P}$, the distribution of random matrix Y^{ℓ} has the form (28).

Now let us consider the problem of determining the index $P = \ell \in \mathcal{P}$, given the observation $Y^{(\ell)}$ following the distribution $\mathbb{P}^{(\ell)}$. Fano's inequality provides a lower bound for probability of error of every estimator $\varphi : \mathbb{R}^{\lambda \times m} \to \mathcal{P}$ in terms of Kullback-Leibler divergence between distributions $\mathbb{P}^{(\ell_1)}$ and $\mathbb{P}^{(\ell_2)}$ $(\ell_1 \neq \ell_2,\ \ell_1, \ell_2 \in [m-k+1])$:

$$\mathbb{P}\left\{\varphi(Y) \neq P\right\} \geq 1 - \frac{\max\limits_{\ell_1 \neq \ell_2 \in \mathcal{P}} \mathrm{KL}\left[\mathbb{P}^{(\ell_1)} || \mathbb{P}^{(\ell_2)}\right] + \log 2}{\log\left(\mathrm{card}(\mathcal{P})\right)}, \tag{29}$$

where $\mathrm{card}(\mathcal{P})$ denotes the cardinality of $\mathcal{P}$ and equals $(m-k+1)$ for our construction.

Let us now derive an upper bound on the quantity

$$\max_{\ell_1 \neq \ell_2 \in \mathcal{P}} \mathrm{KL}\left[\mathbb{P}^{(\ell_1)} || \mathbb{P}^{(\ell_2)}\right]. \tag{30}$$

First, note that for each $\ell \in [m-\kappa+1]$, entries of $Y^{(\ell)}$ are independent. Second, for arbitrary $\ell_1 \neq \ell_2$, the distributions of $Y^{(\ell_1)}$ and $Y^{(\ell_2)}$ differ only in two columns. Thus,

$$\mathrm{KL}\left[\mathbb{P}^{(\ell_1)} || \mathbb{P}^{(\ell_2)}\right] = \lambda \left\{ \mathrm{KL}\left[\mathcal{N}\left(\delta, 1 - \frac{q}{\lambda}\right) || \mathcal{N}\left(0, 1 - \frac{q}{\lambda}\right)\right] + \mathrm{KL}\left[\mathcal{N}\left(0, 1 - \frac{q}{\lambda}\right) || \mathcal{N}\left(\delta, 1 - \frac{q}{\lambda}\right)\right] \right\}.$$

Some simple algebraic manipulations yield:

$$\mathrm{KL}\left[\mathcal{N}\left(\delta, 1 - \frac{q}{\lambda}\right) || \mathcal{N}\left(0, 1 - \frac{q}{\lambda}\right)\right] = \mathrm{KL}\left[\mathcal{N}\left(0, 1 - \frac{q}{\lambda}\right) || \mathcal{N}\left(\delta, 1 - \frac{q}{\lambda}\right)\right] = \frac{\delta^2}{2\left(1 - \frac{q}{\lambda}\right)}. \tag{31}$$

Finally, substituting (31) in (29), for $m > 6$ and for a sufficiently small constant c, we have

$$\mathbb{P}\left\{\varphi(Y) \neq P\right\} \geq 1 - \frac{\frac{\lambda^2 \delta^2}{\lambda - q} + \log 2}{\log\left(m - k + 1\right)} \geq 1 - \frac{c^2 \ln m + 1}{\log\left(\frac{m}{2} + 1\right)} \geq \frac{1}{2}.$$

This lower bound implies

$$\sup_{S \in \mathcal{S}(q)} \inf_{\left(\widehat{\theta}, A \in \mathcal{A}\right)} \sup_{\left(\theta_1^*, \ldots, \theta_m^*\right) \in \mathcal{F}_k(\delta)} \mathbb{P}\left\{\mathcal{T}_k\left(A, \widehat{\theta}\right) \neq \mathcal{T}_k^*\right\} \geq \frac{1}{2}.$$

9.3.3 Proof of Lemma 9

First, let $\widehat{\theta} = \widehat{\theta}^{\mathrm{MEAN}}$. Then given a valid assignment A, the estimates $\widehat{\theta}_j^{\mathrm{MEAN}}, j \in [m]$, are distributed as

$$\widehat{\theta}_j^{\mathrm{MEAN}} \sim \mathcal{N}\left(\theta_j^*, \frac{1}{\lambda^2} \sum_{i \in \mathcal{R}_A(j)} \sigma_{ij}^2\right) = \mathcal{N}\left(\theta_j^*, \bar{\sigma}_j^2\right),$$

where we have defined $\bar{\sigma}_j^2 = \frac{1}{\lambda^2} \sum_{i \in \mathcal{R}_A(j)} \sigma_{ij}^2$. Now let us consider two papers j_1, j_2 such that j_1 belongs to the top k papers $\mathcal{T}_k^*$ and $j_2 \notin \mathcal{T}_k^*$. The probability that paper j_2 receives higher score than paper j_1 is upper bounded as

$$\begin{aligned}
\mathbb{P}\left\{\widehat{\theta}_{j_1}^{\mathrm{MEAN}} \leq \widehat{\theta}_{j_2}^{\mathrm{MEAN}}\right\} &= \mathbb{P}\left\{\left(\widehat{\theta}_{j_1}^{\mathrm{MEAN}} - \widehat{\theta}_{j_2}^{\mathrm{MEAN}}\right) - \mathbb{E}\left\{\widehat{\theta}_{j_1}^{\mathrm{MEAN}} - \widehat{\theta}_{j_2}^{\mathrm{MEAN}}\right\} \leq -\mathbb{E}\left\{\widehat{\theta}_{j_1}^{\mathrm{MEAN}} - \widehat{\theta}_{j_2}^{\mathrm{MEAN}}\right\}\right\} \\
&\overset{(i)}{\leq} \exp\left\{-\frac{\left(\mathbb{E}\left\{\widehat{\theta}_{j_1}^{\mathrm{MEAN}} - \widehat{\theta}_{j_2}^{\mathrm{MEAN}}\right\}\right)^2}{2\left(\bar{\sigma}_{j_1}^2 + \bar{\sigma}_{j_2}^2\right)}\right\} \overset{(ii)}{\leq} \exp\left\{-\left(\frac{\delta}{2\widetilde{\sigma}(A, \widehat{\theta}^{\mathrm{MEAN}})}\right)^2\right\},
\end{aligned}$$

where inequality (i) is due to Hoeffding's inequality, and inequality (ii) holds because $\mathbb{E}\left\{\widehat{\theta}_{j_1}^{\mathrm{MEAN}} - \widehat{\theta}_{j_2}^{\mathrm{MEAN}}\right\} = \theta_{j_1}^* - \theta_{j_2}^* \geq \delta$ and $\widetilde{\sigma}^2(A, \widehat{\theta}^{\mathrm{MEAN}}) = \max_{j \in [m]} \bar{\sigma}_j^2$. The estimator makes a mistake if and only if at least one paper from $\mathcal{T}_k^*$ receives lower score than at least one paper from $[m] \backslash \mathcal{T}_k^*$. A union bound across every paper from $\mathcal{T}_k^*$, paired with $(m-k)$ papers from $[m] \backslash \mathcal{T}_k^*$, yields our claimed result.

Let us now consider $\widehat{\theta} = \widehat{\theta}^{\mathrm{MLE}}$. Then it is not hard to see that

$$\widehat{\theta}_j^{\mathrm{MEAN}} \sim \mathcal{N}\left(\theta_j^*, \left(\sum_{i \in \mathcal{R}_A(j)} \frac{1}{\sigma_{ij}^2}\right)^{-1}\right) = \mathcal{N}\left(\theta_j^*, \bar{\sigma}_j^2\right),$$

where we denoted $\bar{\sigma}_j^2 = \left(\sum_{i \in \mathcal{R}_A(j)} \frac{1}{\sigma_{ij}^2}\right)^{-1}$. Proceeding in a manner similar to the proof for the averaging estimator yields the claimed result.

9.4 Proof of Corollary 4

The proof of Corollary 4 follows along similar lines as the proof of Theorem 3.

9.4.1 Proof of Upper Bound

Let us consider some $\kappa \in [\lambda]$ and $S \in \mathcal{S}_\kappa(v)$. We apply Lemma 9 to proof the upper bound and in order to do so, we need to derive an upper bound on $\widetilde{\sigma}(A^{\text{PR4A}}_{h^{-1}}, \widehat{\theta}^{\text{MLE}})$.

$$\widetilde{\sigma}^2(A^{\text{PR4A}}_{h^{-1}}, \widehat{\theta}^{\text{MLE}}) = \max_{j\in[m]} \left(\sum_{i \in \mathcal{R}_{A^{\text{PR4A}}_{h^{-1}}}(j)} \frac{1}{\sigma^2_{ij}} \right)^{-1} = \left(\min_{j\in[m]} \sum_{i \in \mathcal{R}_{A^{\text{PR4A}}_{h^{-1}}}(j)} h^{-1}(s_{ij}) \right)^{-1}$$
$$\leq \frac{1}{\frac{\kappa}{h(v)} + \frac{\lambda-\kappa}{h(0)}} = \frac{h(v)h(0)}{\kappa h(0) + (\lambda-\kappa)h(v)}.$$

Thus,

$$\sup_{S\in\mathcal{S}_\kappa(v)} \widetilde{\sigma}^2(A^{\text{PR4A}}_{h^{-1}}, \widehat{\theta}^{\text{MLE}}) \leq \frac{h(v)h(0)}{\kappa h(0) + (\lambda-\kappa)h(v)}. \tag{32}$$

It remains to apply Lemma 9 to complete our proof, and we do so by applying the chain of arguments (25a) and (25b) to the bound (32), where the pair $(A^{\text{PR4A}}_{1-h}, \widehat{\theta}^{\text{MEAN}})$ in (25a) and (25b) is substituted with the pair $(A^{\text{PR4A}}_{h^{-1}}, \widehat{\theta}^{\text{MLE}})$.

9.4.2 Proof of Lower Bound

To prove the lower bound, we use the Fano's ineqaulity in the same way as we did when proved Theorem 3(b). However, we now need to be more careful with construction of working similarity matrix $\widetilde{S} \in \mathcal{S}_\kappa(v)$.

As in the proof of Theorem 3(b), we assume $k \leq \frac{m}{2}$. If the converse holds, than the result holds by symmetry of the problem. Next, consider arbitrary feasible assignment $\widetilde{A} \in \mathcal{A}_\kappa$. Recall, that $\mathcal{A}_\kappa$ consists of assignments which assign each paper $j \in [m]$ to κ instead of λ reviewers such that each reviewer reviews at most μ papers.

Now we define a similarity matrix $\widetilde{S}$ as follows:

$$s_{ij} = \begin{cases} v & \text{if} \quad i \in \mathcal{R}_{\widetilde{A}}(j) \\ 0 & \text{otherwise.} \end{cases} \tag{33}$$

Thus, for each paper $j \in [m]$ there exist exactly κ reviewers with non-zero similarity v and in every feasible assignment $A \in \mathcal{A}$ each paper $j \in [m]$ is assigned to at most κ reviewers with non-zero similarity. Note that $\widetilde{S} \in \mathcal{S}_\kappa(v)$.

Now let us consider the set of $(m-k+1)$ problem instances $\mathcal{P}$ defined in Section 9.3.2. For every feasible assignment $A \in \mathcal{A}$, if $Y^{(A,\ell)}$ is a matrix of observed reviewers' scores for instance $\ell \in \mathcal{P}$, then $(r,j)^{\text{th}}$ entry of $Y^{(A,\ell)}$ follows the distribution

$$Y^{(A,\ell)}_{rj} = \begin{cases} \mathcal{N}\big(\delta \times \mathbb{I}\left\{j \in \mathcal{T}^*_k(\ell)\right\}, h(v)\big) & \text{if} \quad \widetilde{A}_{i_r j} = 1 \\ \mathcal{N}\big(\delta \times \mathbb{I}\left\{j \in \mathcal{T}^*_k(\ell)\right\}, h(0)\big) & \text{if} \quad \widetilde{A}_{i_r j} = 0, \end{cases} \tag{34}$$

where $i_r, r \in [\lambda]$ is reviewer assigned to paper j in assignment A.

We denote the distribution of random matrix $Y^{(A,\ell)}$ as $\mathbb{P}^{(A,\ell)}$. Note that in contrast to the proof of Theorem 3, here $Y^{(A,\ell)}$ does depend on the selected assignment $A \in \mathcal{A}$. Thus, instead of (30), we need to derive an upper bound on the quantity

$$\sup_{A\in\mathcal{A}} \max_{\ell_1\neq\ell_2\in\mathcal{P}} \mathrm{KL}\left[\mathbb{P}^{(A,\ell_1)}||\mathbb{P}^{(A,\ell_2)}\right].$$

First, note that for each $\ell \in [m-k+1]$ and for each feasible assignment $A \in \mathcal{A}$, the entries of $Y^{(A,\ell)}$ are independent. Second, for arbitrary $\ell_1 \neq \ell_2$, the distributions of $Y^{(A,\ell_1)}$ and $Y^{(A,\ell_2)}$ differ only in two columns. Thus, for any feasible assignment $A \in \mathcal{A}$, we have

$$\begin{aligned}\mathrm{KL}\left[\mathbb{P}^{(A,\ell_1)}||\mathbb{P}^{(A,\ell_2)}\right] \leq{}& \gamma_{\ell_1}\mathrm{KL}\left[\mathcal{N}\big(\delta,h(v)\big)||\mathcal{N}\big(0,h(v)\big)\right]+(\lambda-\gamma_{\ell_1})\,\mathrm{KL}\left[\mathcal{N}\big(\delta,h(0)\big)||\mathcal{N}\big(0,h(0)\big)\right]\\ &+\gamma_{\ell_2}\mathrm{KL}\left[\mathcal{N}\big(0,h(v)\big)||\mathcal{N}\big(\delta,h(v)\big)\right]+(\lambda-\gamma_{\ell_2})\,\mathrm{KL}\left[\mathcal{N}\big(0,h(0)\big)||\mathcal{N}\big(\delta,h(0)\big)\right]\\ ={}&(\gamma_{\ell_1}+\gamma_{\ell_2})\,\frac{\delta^2}{2h(v)}+(2\lambda-\gamma_{\ell_1}-\gamma_{\ell_2})\,\frac{\delta^2}{2h(0)},\end{aligned} \tag{35}$$

where γ_{ℓ_1} is the number of reviewers with similarity v assigned to paper $(k-1+\ell_1)$ in A and γ_{ℓ_2} is the number of reviewers with similarity v assigned to paper $(k-1+\ell_2)$. By construction of similarity matrix $\widetilde{S}$, for each $\ell \in [m-k+1]$ and for each $A \in \mathcal{A}$, we have $\gamma_\ell \leq \kappa$. Note that two summands in (35) are proportional to a convex combination of $\frac{\delta^2}{2h(v)}$ and $\frac{\delta^2}{2h(0)}$. Moreover, by monotonicity of h, we have $\frac{\delta^2}{2h(v)} \geq \frac{\delta^2}{2h(0)}$, and hence

$$\sup_{A\in\mathcal{A}} \max_{\ell_1\neq\ell_2\in\mathcal{P}} \mathrm{KL}\left[\mathbb{P}^{(A,\ell_1)}||\mathbb{P}^{(A,\ell_2)}\right] \leq \frac{\kappa\delta^2}{h(v)}+\frac{(\lambda-\kappa)\,\delta^2}{h(0)} = \delta^2\left(\frac{\kappa h(0)+(\lambda-\kappa)\,h(v)}{h(v)h(0)}\right).$$

Applying Fano's ineqaulity (29), we conclude that for all feasible assignments $A \in \mathcal{A}$, if $m > 6$ and universal constant c is sufficiently small, then

$$\mathbb{P}\left\{\varphi(Y)\neq P\right\} \geq 1-\frac{\delta^2\left(\frac{\kappa h(0)+(\lambda-\kappa)h(v)}{h(v)h(0)}\right)+\log 2}{\log\left(m-k+1\right)} \geq 1-\frac{c^2\ln m+1}{\log\left(\frac{m}{2}+1\right)} \geq \frac{1}{2}.$$

This bound thus implies

$$\sup_{S\in\mathcal{S}_\kappa(v)}\inf_{\left(\widehat{\theta},A\in\mathcal{A}\right)}\sup_{\left(\theta_1^*,\ldots,\theta_m^*\right)\in\mathcal{F}_k(\delta)}\mathbb{P}\left\{\mathcal{T}_k\left(A,\widehat{\theta}\right)\neq\mathcal{T}_k^*\right\}\geq\frac{1}{2}.$$

9.5 Proof of Theorem 5

Before we prove the theorem, we state an auxiliary proposition which will help us to prove a lower bound.

Lemma 10 (Shah and Wainwright, 2015) *Let $t > 0$ be an integer such that $2t \leq \frac{1}{1+\nu_2}\min\left\{m^{1-\nu_1},k,m-k\right\}$ for some constants $\nu_1,\nu_2 \in (0;1)$ and m is larger than some (ν_1,ν_2)-dependent constant. Then there exist a set of binary strings $\left\{b^1,b^2,\ldots,b^L\right\} \subseteq \{0,1\}^{m/2}$ with cardinality $L > \exp\left\{\frac{9}{10}\nu_1\nu_2 t\log m\right\}$ such that*

$$\mathcal{D}_H\left(b^{\ell_1},\mathbf{0}_{m/2}\right)=2(1+\nu_2)t \quad \textit{and} \quad \mathcal{D}_H\left(b^{\ell_1},b^{\ell_2}\right)>4t \quad \forall\ell_1\neq\ell_2\in[L]$$

The proof of Lemma 10 relies on a coding-theoretic result due to Levenshtein (1971) which gives a lower bound on the number of codewords of fixed length m and Hamming weights c_1 with Hamming distance between each pair of codewords higher than c_2.

9.5.1 Proof of Upper Bound

Without loss of generality we assume that the true underlying ranking of the papers is $1, 2, \ldots, k, \ldots, m$. We prove the claim for pair $\left(A^{\text{PR4A}}_{1-h}, \widehat{\theta}^{\text{MEAN}}\right)$ below, and proof for $\left(A^{\text{PR4A}}_{h^{-1}}, \widehat{\theta}^{\text{MLE}}\right)$ follows from the proof of the corresponding part of Theorem 3(a).

From the proof of Lemma 9 and Section 9.3.1, we know that under conditions of the theorem, for every paper $j_1 \le k - t$ and for every paper $j_2 \ge k + t + 1$,

$$\sup_{S \in \mathcal{S}(q)} \mathbb{P}\left\{\widehat{\theta}^{\text{MEAN}}_{j_1} - \widehat{\theta}^{\text{MEAN}}_{j_2} \le 0\right\} \le \exp\left\{-\left(\frac{\delta}{2 \sup\limits_{S \in \mathcal{S}(q)} \widetilde{\sigma}(A^{\text{PR4A}}_{1-h}, \widehat{\theta}^{\text{MEAN}})}\right)^2\right\} \tag{36a}$$

where

$$\sup_{S \in \mathcal{S}(q)} \widetilde{\sigma}^2(A^{\text{PR4A}}_{1-h}, \widehat{\theta}^{\text{MEAN}}) \le \frac{\lambda - \tau_q q}{\lambda^2}. \tag{36b}$$

Taking a union bound across every paper from the top $(k-t)$ papers, paired with the bottom $(m-k-t)$ papers, we obtain

$$\begin{aligned}\sup_{S \in \mathcal{S}(q)} \mathbb{P}\left\{\exists j_1 \le k - t, j_2 \ge k + t + 1 \quad \text{such that } \widehat{\theta}^{\text{MEAN}}_{j_1} \le \widehat{\theta}^{\text{MEAN}}_{j_2}\right\} &\le m^2 \exp\left\{-\frac{\lambda^2 \delta^2}{4(\lambda - \tau_q q)}\right\} \\ &\le \epsilon.\end{aligned}$$

In other words, for every similarity matrix $S \in \mathcal{S}(q)$, with probability at least $(1-\epsilon)$, the top $(k-t)$ papers will receive higher score than bottom $(m-k-t)$ papers. Thus, among accepted papers $\mathcal{T}_k\left(A^{\text{PR4A}}_{1-h}, \widehat{\theta}^{\text{MEAN}}\right)$, at most t papers will not belong to $\mathcal{T}^*_k$, thereby ensuring that

$$\mathcal{D}_H\left(\mathcal{T}_k\left(A^{\text{PR4A}}_{1-h}, \widehat{\theta}^{\text{MEAN}}\right), \mathcal{T}^*_k\right) \le 2t$$

with probability at least $1 - \epsilon$.

9.5.2 Proof of Lower Bound

To prove the lower bound, we follow similar path as we used when we derived a lower bound in Theorem 3. However, we now need more advanced technique to construct necessary set of instances.

As in the proof of Theorem 3(b), we assume that $k \le \frac{m}{2}$. If the converse holds, than the result holds by the symmetry of the problem. Next, consider similarity matrix $\widetilde{S} = \left\{h^{-1}\left(1 - \frac{q}{\lambda}\right)\right\}^{n \times m} \in \mathcal{S}(q)$. To apply Fano's inequality, it remains to construct a set $\mathcal{P} = \{1, 2, \ldots, L\}$ of suitable instances of paper accepting/rejecting problem: every problem instance in this set has the same similarity matrix $\widetilde{S}$, but differs in the set of top k papers

$\mathcal{T}_k^*$. We note that in contrast to the proof of Theorem 3(b), it is not enough to create $(m-k+1)$ instances where the sets of top k papers differ only in a single paper. As we will see below, it suffices to construct instances such that for every $\ell_1, \ell_2 \in \mathcal{P}$, the sets of top k papers satisfy $\mathcal{D}_H\left(\mathcal{T}_k^*(\ell_1), \mathcal{T}_k^*(\ell_2)\right) > 4t$.

Note that requirements of Lemma 10 are satisfied by the conditions of Theorem 5. Let $\left\{b^1, b^2, \ldots, b^L\right\}$ be the corresponding binary strings. For every problem $\ell \in \mathcal{P}$, consider the following binary string:

$$\widetilde{b}^\ell = \underbrace{\overbrace{1, 1, \ldots, 1}^{k-2(1+\nu_2)t}, \overbrace{0, 0, \ldots, 0}^{m/2}}, b_1^\ell, b_2^\ell, \ldots, b_{m/2}^\ell. \tag{37}$$

First, note that $2t \leq \frac{1}{1+\nu_2}k$, and hence $k - 2(1+\nu_2)t \geq 0$, thereby ensuring that the construction (37) is not vacuous. Now let $\mathcal{T}_k^*(\ell)$ be the set of indices such that their corresponding elements in string $\widetilde{b}^\ell$ equal 1. By construction, the cardinality of $\mathcal{T}_k^*(\ell)$ is k so it is a valid set of top k papers. Finally, we need to set the scores of papers. Let for every paper $j \in [m]$:

$$\theta_j^*(\ell) = \begin{cases} \delta & \text{if} \quad \widetilde{b}_j^\ell = 1 \\ 0 & \text{if} \quad \widetilde{b}_j^\ell = 0, \end{cases}$$

which ensures that for every $\ell \in \mathcal{P}$, $(\theta_1^*(\ell), \theta_2^*(\ell), \ldots, \theta_m^*(\ell)) \in \mathcal{F}_k \subset \mathcal{F}_{k,t}$.

The strategy for the remaining part of the proof is the following. We first show that the problem instances defined above are well-separated in a sense that for any two of them, the corresponding sets of the top k papers differ in sufficiently many elements. We then assume that there exists an (assignment algorithm, estimator) pair which for every similarity matrix $S \in \mathcal{S}(q)$ recovers the set of top k papers with at most t errors with high probability. Then this pair must be able to determine with high probability the problem instance ℓ, sampled uniformly at random from $\mathcal{P}$, by observing corresponding reviewers' scores. We then apply Fano's inequality to show the impossibility of the last implication.

Following the plan described above, we note that for every two distinct instances $\ell_1, \ell_2 \in \mathcal{P}$, we have

$$\mathcal{D}_H\left(\mathcal{T}_k^*(\ell_1), \mathcal{T}_k^*(\ell_2)\right) > 4t.$$

Consequently, for every set $\mathcal{T}_k^*$ of k papers, $\mathcal{D}_H\left(\mathcal{T}_k^*, \mathcal{T}_k^*(\ell)\right) \leq 2t$ for at most one instance $\ell \in \mathcal{P}$. Now, we will complete the proof of this theorem with a proof by contradiction. For this, assume for the sake of contradiction that for every similarity matrix $S \in \mathcal{S}(q)$, there exists an assignment $\widetilde{A} = \widetilde{A}(S)$ and estimator $\widehat{\theta} = \widehat{\theta}(S)$ such that for arbitrarily large value of m

$$\sup_{\left(\theta_1^*, \ldots, \theta_m^*\right) \in \mathcal{F}_k(\delta)} \mathbb{P}\left\{\mathcal{D}_H\left(\mathcal{T}_k\left(\widetilde{A}, \widehat{\theta}\right), \mathcal{T}_k^*\right) > 2t\right\} < \frac{1}{2}. \tag{38}$$

This assumption implies that estimator $\widehat{\theta}(\widetilde{S})$ might be used to determine the problem $P = \ell$ sampled uniformly at random from $\mathcal{P}$ correctly with probability greater than $1/2$. Indeed,

notice that similarity matrix $\widetilde{S}$ was constructed in a way that $\mathcal{T}_k\left(\widetilde{A}, \widehat{\theta}\right)$ does not depend on assignment $\widetilde{A}$.

Given $P = \ell$, let $Y^{(\ell)}$ be the random matrix of reviewers' scores. The distribution $\mathbb{P}^{(\ell)}$ of components of $Y^{(\ell)}$ is defined in (28). To apply Fano's inequality (29), it remains to derive an upper bound on the quantity $\max_{\ell_1 \neq \ell_2 \in \mathcal{P}} \mathrm{KL}\left[\mathbb{P}^{(\ell_1)} || \mathbb{P}^{(\ell_2)}\right]$.

First, note that entries of $Y^{(\ell)}$ are independent. Second, note that for every pair $\ell_1 \neq \ell_2 \in \mathcal{P}$ and for every $j \in [m/2]$, the distribution of the j^{th} column of $Y^{(A,\ell_1)}$ is identical to the distribution of the j^{th} column of $Y^{(A,\ell_2)}$. Among the last $m/2$ columns, the distributions of at most $4(1+\nu_2)t$ columns of $Y^{(A,\ell_1)}$ differ from the distributions of the corresponding columns in $Y^{(A,\ell_2)}$. Thus, for arbitrary $\ell_1 \neq \ell_2 \in \mathcal{P}$

$$\mathrm{KL}\left[\mathbb{P}^{(\ell_1)} || \mathbb{P}^{(\ell_2)}\right] \leq 2(1+\nu_2)t\lambda \Bigg\{ \mathrm{KL}\left[\mathcal{N}\left(\delta, 1-\frac{q}{\lambda}\right) || \mathcal{N}\left(0, 1-\frac{q}{\lambda}\right)\right] + \mathrm{KL}\left[\mathcal{N}\left(0, 1-\frac{q}{\lambda}\right) || \mathcal{N}\left(\delta, 1-\frac{q}{\lambda}\right)\right] \Bigg\}.$$

Recalling (31), we deduce that

$$\max_{\ell_1 \neq \ell_2 \in \mathcal{P}} \mathrm{KL}\left[\mathbb{P}^{(\ell_1)} || \mathbb{P}^{(\ell_2)}\right] \leq 4(1+\nu_2)t\lambda \frac{\lambda\delta^2}{2(\lambda - q)} = 2(1+\nu_2)t\frac{\lambda^2\delta^2}{\lambda - q} \leq 4c^2\nu_1\nu_2 t \ln m.$$

Finally, Fano's inequality together with Lemma 10 ensures that for every estimator $\varphi : Y \to \mathcal{P}$

$$\mathbb{P}\left\{\varphi(Y) \neq P\right\} \geq 1 - \frac{4c^2\nu_1\nu_2 t \ln m + \log 2}{\frac{9}{10}\nu_1\nu_2 t \log m} \geq 1 - \frac{40}{9}c^2\frac{\ln m}{\log m} - \frac{1}{\frac{9}{10}\nu_1\nu_2 t \log m} \geq \frac{1}{2}$$

for m larger than some (ν_1, ν_2)-dependent constant and small enough universal constant c. This leads to a contradiction with (38), thus proving the theorem.

9.6 Proof of Corollary 6

The proof of the Corollary 6 is based on the ideas of the proofs of Theorem 5 and Corollary 4 and repeats them with minor changes.

9.6.1 Proof of Upper Bound

To show the required upper bound, we repeat the proof of Theorem 6(a) from Section 9.5.1 with the following changes. Equation (36a) should be substituted with:

$$\sup_{S \in \mathcal{S}_\kappa(v)} \mathbb{P}\left\{\widehat{\theta}^{\text{MLE}}_{j_1} - \widehat{\theta}^{\text{MLE}}_{j_2} \leq 0\right\} \leq \exp\left\{-\left(\frac{\delta}{2\sup\limits_{S \in \mathcal{S}_\kappa(v)} \widetilde{\sigma}(A^{\text{PR4A}}_{h^{-1}}, \widehat{\theta}^{\text{MLE}})}\right)^2\right\}.$$

Equation (36b) should be substituted with:

$$\sup_{S \in \mathcal{S}_\kappa(v)} \widetilde{\sigma}^2(A^{\text{PR4A}}, \widehat{\theta}^{\text{MLE}}) \leq \frac{h(v)h(0)}{\kappa h(0) + (\lambda - \kappa)h(v)}.$$

In the remaining part of the proof, pair $(A^{\text{PR4A}}_{1-h}, \widehat{\theta}^{\text{MEAN}})$ should be substituted with the pair $(A^{\text{PR4A}}_{h^{-1}}, \widehat{\theta}^{\text{MLE}})$.

9.6.2 Proof of Lower Bound

To prove the lower bound, we use the set of problems $\mathcal{P}$ constructed in Section 9.5.2 and the similarity matrix $\widetilde{S}$ as defined in (33).

Given $P = \ell$ and any feasible assignment $A \in \mathcal{A}$, let $Y^{(A,\ell)}$ be the random matrix of reviewers' scores. The distribution $\mathbb{P}^{(A,\ell)}$ of components of $Y^{(A,\ell)}$ is defined in (34). Since the distribution of reviewers' scores now depends on the assignment, to apply Fano's inequality (29), we need to derive an upper bound on the quantity $\sup_{A\in\mathcal{A}} \max_{\ell_1\neq\ell_2\in\mathcal{P}} \mathrm{KL}\left[\mathbb{P}^{(A,\ell_1)}||\mathbb{P}^{(A,\ell_2)}\right]$.

First, note that entries of $Y^{(A,\ell)}$ are mutually independent. Second, note that for every pair $\ell_1 \neq \ell_2 \in \mathcal{P}$ and for every $j \in [m/2]$, the distribution of the j^{th} column of $Y^{(A,\ell_1)}$ is identical to the distribution of the j^{th} column of $Y^{(A,\ell_2)}$. Among the last $m/2$ columns, the distributions of at most $4(1+\nu_2)t$ columns of $Y^{(A,\ell_1)}$ differ from the distributions of the corresponding columns in $Y^{(A,\ell_2)}$. Next, consider arbitrary feasible assignment $A \in \mathcal{A}$. Let $\gamma_{\ell_1}^{(r)}, r \in [2(1+\nu_2)t]$, denote the number of strong reviewers (with similarity v) assigned in A to paper $j_1^{(r)} \in \mathcal{T}_k^*(\ell_1)$, where paper $j_1^{(r)}$ corresponds to the the second part of the string $\widetilde{b}^{\ell_1}$ defined in (37). Recall now that there are at most $4(1+\nu_2)t$ papers that belong to exactly one of the sets $\mathcal{T}_k^*(\ell_1)$ and $\mathcal{T}_k^*(\ell_2)$. Hence, the equation for upper bound of the Kullback-Leibler divergence between $\mathbb{P}^{(A,\ell_1)}$ and $\mathbb{P}^{(A,\ell_2)}$ is obtained by assuming that all the papers that belong to the $\mathcal{T}_k^*(\ell_1)$ and correspond to the second half of the string $\widetilde{b}^{\ell}$ do not belong to $\mathcal{T}_k^*(\ell_2)$ and vice versa. Thus, similar to how we got (35), for arbitrary $\ell_1 \neq \ell_2 \in \mathcal{P}$ and for arbitrary feasible assignment $A \in \mathcal{A}$, we have

$$
\begin{aligned}
&\mathrm{KL}\left[\mathbb{P}^{(A,\ell_1)}||\mathbb{P}^{(A,\ell_2)}\right] \\
&\quad\leq \sum_{r=1}^{2(1+\nu_2)t} \left\{\gamma_{\ell_1}^{(r)}\mathrm{KL}\left[\mathcal{N}\big(\delta,h(v)\big)||\mathcal{N}\big(0,h(v)\big)\right] + \left(\lambda-\gamma_{\ell_1}^{(r)}\right)\mathrm{KL}\left[\mathcal{N}\big(\delta,h(0)\big)||\mathcal{N}\big(0,h(0)\big)\right]\right\} \\
&\quad+ \sum_{r=1}^{2(1+\nu_2)t} \left\{\gamma_{\ell_2}^{(r)}\mathrm{KL}\left[\mathcal{N}\big(0,h(v)\big)||\mathcal{N}\big(\delta,h(v)\big)\right] + \left(\lambda-\gamma_{\ell_2}^{(r)}\right)\mathrm{KL}\left[\mathcal{N}\big(0,h(0)\big)||\mathcal{N}\big(\delta,h(0)\big)\right]\right\} \\
&\quad= \left(\sum_{r=1}^{2(1+\nu_2)t}\left(\gamma_{\ell_1}^{(r)}+\gamma_{\ell_2}^{(r)}\right)\right)\frac{\delta^2}{2h(v)} + \left(4(1+\nu_2)t\lambda - \sum_{r=1}^{2(1+\nu_2)t}\left(\gamma_{\ell_1}^{(r)}+\gamma_{\ell_2}^{(r)}\right)\right)\frac{\delta^2}{2h(0)}.
\end{aligned}
$$

Noting that $\frac{\delta^2}{2h(v)} \geq \frac{\delta^2}{2h(0)}$, we obtain

$$
\begin{aligned}
\sup_{A\in\mathcal{A}} \max_{\ell_1\neq\ell_2\in\mathcal{P}} \mathrm{KL}\left[\mathbb{P}^{(A,\ell_1)}||\mathbb{P}^{(A,\ell_2)}\right] &\leq 2(1+\nu_2)t\left(\frac{\kappa\delta^2}{h(v)} + \frac{(\lambda-\kappa)\,\delta^2}{h(0)}\right) \\
&= 2(1+\nu_2)t\delta^2\left(\frac{\kappa h(0) + (\lambda-\kappa)\,h(v)}{h(v)h(0)}\right) \\
&\leq 4c^2\nu_1\nu_2 t\ln m.
\end{aligned}
$$

Applying Fano's inequality (29), we obtain the desired lower bound.

9.7 Proof of Theorem 7

Note that Theorem 7 is similar in nature with Theorem 3, the only difference is that now we are trying to recover a ranking which is induced by the assignment.

9.7.1 Proof of Upper Bound

Given any feasible assignment A, the "ground truth" ranking that we try to recover is given by

$$\widetilde{\theta}^\star_j(A) = \frac{1}{\lambda} \sum_{i \in \mathcal{R}_A(j)} \widetilde{\theta}_{ij}. \tag{39}$$

Then the estimates $\widehat{\theta}^{\text{MEAN}}_j, j \in [m]$, are distributed as

$$\widehat{\theta}^{\text{MEAN}}_j \sim \mathcal{N}\left(\frac{1}{\lambda} \sum_{i \in \mathcal{R}_A(j)} \widetilde{\theta}_{ij}, \frac{1}{\lambda^2} \sum_{i \in \mathcal{R}_A(j)} \sigma^2_{ij} \right) = \mathcal{N}\left(\widetilde{\theta}^\star_j(A), \bar{\sigma}^2_j \right), \tag{40}$$

where $\bar{\sigma}^2_j = \frac{1}{\lambda^2} \sum\limits_{i \in \mathcal{R}_A(j)} \sigma^2_{ij}$. Now observe that Lemma 9, with $\mathcal{T}^\star_k\left(A, \widetilde{\theta}^\star(A)\right)$ substituted for $\mathcal{T}^*_k$, also holds for the subjective score model and the averaging estimator $\widehat{\theta}^{\text{MEAN}}$. Thus, repeating the proof of the upper bound for averaging estimator in Theorem 3(a) and substituting $\mathcal{T}^*_k$ with $\mathcal{T}^\star_k\left(A^{\text{PR4A}}, \widetilde{\theta}^\star(A^{\text{PR4A}})\right)$ in (25a), yields the claimed result.

9.7.2 Proof of Lower Bound

The lower bound directly follows from Theorem 3(b). To see this, consider the following matrix of reviewers' subjective scores: $\widetilde{\Theta} = \left\{\widetilde{\theta}_{ij}\right\}_{i \in [n], j \in [m]}$, where $\widetilde{\theta}_{ij} = \theta^*_j$. Under this assumption, the total ranking induced by assignment A does not depend on the assignment: $\widetilde{\theta}^\star_j(A) = \theta^*_j$. Now we can conclude that such choice of $\widetilde{\Theta}$ brings us to the objective model setup in which true underlying ranking exists and does not depend on the assignment. Thus, the lower bound of Theorem 3(b) transfers to the subjective score model.

9.8 Proof of Theorem 8

The proof of the Theorem 8 is based on the ideas of the proofs of Theorem 5 and Theorem 7 and repeats them with minor changes.

9.8.1 Proof of Upper Bound

Having equations (39) and (40), we note that the goal now mimics the goal we achieved when proved an upper bound for averaging estimator in Theorem 5.

9.8.2 Proof of Lower Bound

The argument from Section 9.7.2 ensures that the lower bound established in Theorem 5 directly transfers to the to the subjective score model.

10. Discussion

Researchers submit papers to conferences expecting a fair outcome from the peer-review process. This expectation is often not met, as is illustrated by the difficulties that non-mainstream or inter-disciplinary research faces in present peer-review systems. We design a reviewer-assignment algorithm PeerReview4All to address the crucial issues of fairness and accuracy. Our guarantees impart promise for deploying the algorithm in conference peer-reviews.

There are number of open problems suggested by our work. The first direction is associated with approximation algorithms and corresponding guarantees established in this work. One goal is to determine whether there exists a polynomial-time algorithm with worst case approximation guarantees better than $1/\lambda$ established in this paper (7b). It would also be useful to obtain a deeper understanding of the adaptive behavior of our algorithm with bounds more nuanced than (7a). Next, an interesting direction is to consider other notions of fairness (including group fairness) and study the performance of the PeerReview4All algorithm in these settings. Finally, we leave the task of improving the computational efficiency of our PeerReview4All algorithm out of the scope of this work. However, we suggest that optimal implementation of Subroutine 1 should not be based on the general max-flow algorithm and instead should rely on algorithms specifically designed to work fast on layered graphs.

The second direction is related to the statistical part of our work. In this paper we provide a minimax characterization of the simplified version of the paper acceptance problem. This simplified procedure may be considered as an initial estimate that can be used as a guideline for the final decisions. However, there remain a number of other factors, such as self-reported confidence of reviewers or inter-reviewer discussions, that may additionally be included in the model.

Finally, an important related problem is to improve the assessment of similarities between reviewers and papers. It will be interesting to see whether the problems of assessing similarities and assigning reviewers can be addressed jointly in an active manner possibly incorporating feedback from the previous iterations of the conference. Additionally, in this work, we follow major machine learning conferences such as NeurIPS, ICML, and AAAI, and define the measure of assignment quality with respect to a paper as a sum of similarities between the paper and assigned reviewers. However, it may also be useful to come up with a more principled notion of assignment quality that takes into account various idiosyncrasies of the review process such as discussion and rebuttal.

Acknowledgments

This work was supported in parts by NSF grants CRII: CIF: 1755656, CIF: 1563918, and CIF: 1763734.

Appendix

We provide supplementary materials and additional discussion.

Appendix A. Discussion of Approximation Results

In this section we discuss the approximation-related results. In what follows we consider function $f(s) = s$ and for any value $c \in \mathbb{R}$, we denote the matrix all of whose entries are c as $\mathbf{c}$.

A.1 Example for ILPR Algorithm.

We begin by construction a series of similarity matrices for various λ such that $\Gamma^S\left(A^{\text{ILPR}}\right) = 0$ while assignments A^{PR4A} and A^{HARD} have non-trivial fairness.

Proposition 11 *For every positive integer λ, there exists a similarity matrix S such that $\Gamma^S\left(A^{ILPR}\right) = 0$ and $\Gamma^S\left(A^{PR4A}\right) \geq \frac{1}{\lambda}\Gamma^S\left(A^{HARD}\right) > 0$.*

Proof Given any positive integer $\lambda \in \mathbb{N}$, consider an instance of reviewer assignment problem with $m = n$, $\mu = \lambda$ and similarities given by the block matrix

$$S = \left[\begin{array}{c|c|c} \mathbf{1} & \mathbf{1} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & (\widetilde{s} - \varepsilon) \cdot \mathbf{1} \\ \hline \underbrace{(\widetilde{s} - \varepsilon) \cdot \mathbf{1}}_{m_1} & \underbrace{(\widetilde{s} - \varepsilon) \cdot \mathbf{1}}_{m_1} & \underbrace{\widetilde{s} \cdot \mathbf{1}}_{m_1} \end{array}\right] \begin{array}{l} \} n_1 \\ \} n_2 \\ \} n_3 \end{array} \tag{41}$$

Here $\widetilde{s} = \frac{n_1}{n_1+n_2}$, the value $\varepsilon > 0$ is some small constant strictly smaller than $\widetilde{s}$, and $n_r = m_r > 0$ for every $r \in \{1, 2, 3\}$. We also require $n_3 > \lambda$ and

$$n_2 = (\lambda - 1)\, n_1 + 1. \tag{42}$$

We refer to the first m_1 papers and n_1 reviewers as belonging to the first group, the second m_2 papers and n_2 reviewers as belonging to the second group, and so on.

The ILPR algorithm involves two steps. The first step consists of solving a linear programming relaxation and finding the most fair fractional assignment. The second step then performs a rounding procedure in order to obtain integer assignments. Let us first see the output of the first step of the ILPR algorithm — the fractional assignment with the highest fairness — on the similarity matrix (41). Observe that for each of the m_3 papers in the third group, the sum of the similarities of any λ reviewers is at most $\lambda\widetilde{s}$, and furthermore, that this value is achieved with equality if and only if they are reviewed by λ reviewers from the third group. Next, the n_1 reviewers from the first group can together review λn_1 papers. Dividing this amount equally over the $m_1 + m_2$ papers in the first two groups (in any arbitrary manner) and complementing the assignment with reviewers from the second group, we see that each paper from the first and the second groups receives a sum similarity $\lambda\frac{n_1}{m_1+m_2} = \lambda\widetilde{s}$. It is not hard to see that any deviation from the assignment introduced above will lead to a strict decrease of the fairness.

	$\lambda = 1$	$\lambda = 2$	$\lambda = 3$	$\lambda = 4$
$\Gamma^S\left(A^{\mathrm{ILPR}}\right)$	0	0	0	0
$\Gamma^S\left(A^{\mathrm{HARD}}\right)$	0.49	0.65	0.72	0.76
$\Gamma^S\left(A^{\mathrm{PR4A}}\right)$	0.49	0.65	0.72	0.76

Table 6: Fairness of various assignment algorithms for the class of similarity matrices (41).

The second step of the ILPR algorithm is a rounding procedure that constructs a feasible assignment from the fractional assignment (solution of linear programming relaxation) obtained in the previous step. The rounding procedure is guaranteed to assign λ reviewers to each paper, respecting the following condition: any reviewer assigned to any paper $j \in [m]$ in the resulting feasible assignment must have a non-zero fraction allocated to that paper in the fractional assignment.

Now notice that aforementioned condition ensures that all papers from the third group must be assigned to reviewers from the third group. Next, recall that on one hand, reviewers from the first group can together review at most λn_1 different papers. On the other hand, in each optimally fair fractional assignment, the first $m_1 + m_2$ papers are assigned to reviewers from the first two groups. Thus, in the resulting integral assignment these papers also must be assigned to reviewers from the first two groups. These two facts together with the inequality $\lambda n_1 < m_1 + m_2$ that we obtain from (42) ensure that at least one paper in the resulting integral assignment will be reviewed by λ reviewers with zero similarity. Hence, the assignment computed by the ILPR algorithm has zero fairness $\Gamma^S\left(A^{\mathrm{ILPR}}\right) = 0$.

On the other hand, it is not hard to see that $\Gamma^S\left(A^{\mathrm{HARD}}\right) \geq \widetilde{s} - \varepsilon$. Indeed, let us assign one reviewer to each paper by the following procedure: the m_1 papers from the first group and some $m_2 - 1$ papers from the second group are all assigned one arbitrary reviewer each from the first group of reviewers. Such an assignment is possible since $\lambda n_1 = m_1 + m_2 - 1$ due to (42). The remaining paper from the second group is assigned one arbitrary reviewer from the third group. At this point, there are m_3 papers (in the third group) which are not yet assigned to any reviewer, and $n_3 + n_2 - 1 \geq m_3$ reviewers who have not been assigned any paper and have similarity higher than $\widetilde{s} - \varepsilon$ with these m_3 papers in the third group. Assigning one reviewer each from this set to each of these m_3 papers, we obtain an assignment in which each paper is allocated to one reviewer with similarity at least $\widetilde{s} - \varepsilon$. Completing the remaining assignments in an arbitrary fashion, we conclude that $\Gamma^S\left(A^{\mathrm{PR4A}}\right) \geq \frac{1}{\lambda}\Gamma^S\left(A^{\mathrm{HARD}}\right) \geq \widetilde{s} - \varepsilon > 0$ where first inequality is due to Theorem 1. ■

The results of simulations for $\lambda \in \{1, 2, 3, 4\}$, parameters $n_1 = 1, n_2 = \lambda, n_3 = \lambda + 1, \varepsilon = 0.01$ and similarity matrices $\widetilde{S}$ defined in (41) are depicted in Table 6. Interestingly, for these choices of parameters, our PeerReview4All algorithm is not only superior to ILPR, but is also able to exactly recover the fair assignment.

A.2 Sub-optimality of TPMS

In this section we show that assignment obtained from optimizing the objective (2) can be highly sub-optimal with respect to the criterion (4) even when f is the identity function.

Proposition 12 *For any $\lambda \geq 1$, there exists a similarity matrix S such that $\Gamma^S\left(A^{PR4A}\right) = \Gamma^S\left(A^{HARD}\right) \geq \frac{\lambda}{4}$ and $\Gamma^S\left(A^{TPMS}\right) = 0$.*

Proof Consider an instance of the problem with $m = n = 2\lambda$, and similarities given by the block matrix

$$S = \left[\begin{array}{c|c} \mathbf{1} & \mathbf{0.4} \\ \hline \underbrace{\mathbf{0.4}}_{\lambda} & \underbrace{\mathbf{0}}_{\lambda} \end{array}\right] \begin{matrix} \} \lambda \\ \} \lambda \end{matrix} \tag{43}$$

Then A^{TPMS} assigns the first λ reviewers to the first λ papers (in some arbitrary manner) and the remaining reviewers to the remaining papers, obtaining

$$\sum_{j \in [m]} \sum_{i \in \mathcal{R}_{A^{\text{TPMS}}}(j)} s_{ij} = \lambda^2 \text{ and}$$
$$\Gamma^S\left(A^{\text{TPMS}}\right) = 0$$

In contrast, assignments A^{PR4A} and A^{HARD} assign the first $\frac{1}{2}n$ reviewers to the second group of papers and the remaining reviewers to the remaining papers. This assignment yields

$$\sum_{j \in [m]} \sum_{i \in \mathcal{R}_{A^{\text{PR4A}}}(j)} s_{ij} = \sum_{j \in [m]} \sum_{i \in \mathcal{R}_{A^{\text{HARD}}}(j)} s_{ij} = 0.8\lambda^2 \text{ and}$$
$$\Gamma^S\left(A^{\text{PR4A}}\right) = \Gamma^S\left(A^{\text{HARD}}\right) = 0.4\lambda \geq \frac{\lambda}{4}.$$

This concludes the proof. ■

A.3 Example of $1/\lambda$ Approximation Factor for A^{PR4A}

Let us consider an instance of fair assignment problem with $m = n = 4,\ \lambda = \mu = 2$ and similarities represented in Table 7.

First, note that $\Gamma^S\left(A^{\text{HARD}}\right) \leq 0.6$. This is because in every feasible assignment $A \in \mathcal{A}$ paper 1 in the best case is assigned to reviewers 1 and 2. Moreover, there exists a feasible assignment represented as A^{HARD} in Table 8 which achieves a max-min fairness of 0.6 and hence we have $\Gamma^S\left(A^{\text{HARD}}\right) = 0.6$.

Let us now analyze the performance of PeerReview4All algorithm. Again, the fairness of the resulting assignment is determined in the first iteration of Step 2 to 7 of Algorithm 1, so we restrict our attention to that part of the algorithm. It is not hard to see that after Step 2 is executed, we have two candidates assignments, A_1 and A_2, represented in Table 8 (up to not important randomness in braking ties). Computing the fairness of these assignments, we obtain

$$\Gamma^S(A_1) = 0.3 + \varepsilon \quad \text{and} \quad \Gamma^S(A_2) = 0.2.$$

	Paper a	Paper b	Paper c	Paper d
Reviewer 1	$0.3+\epsilon$	1	1	0
Reviewer 2	$0.3-\epsilon$	0	1	1
Reviewer 3	0	0.1	0	0.3
Reviewer 4	0	0.1	0	0.3

Table 7: An example of similarities that yield $1/\lambda$ approximation factor of the PeerReview4All algorithm.

	A^{HARD}		A_1		A_2	
	1st Rev.	2nd Rev.	1st Rev.	2nd Rev.	1st Rev.	2nd Rev.
Paper a	1	2	1	3	1	2
Paper b	1	3	1	3	3	4
Paper c	2	4	2	4	1	2
Paper d	3	4	2	4	3	4

Table 8: The optimal assignment as well as and PeerReview4All 's intermediate assignments for the similarities in Table 7.

which implies that

$$\frac{\Gamma^S\left(A^{\text{PR4A}}\right)}{\Gamma^S\left(A^{\text{HARD}}\right)} = \frac{\max\left\{\Gamma^S\left(A_1\right), \Gamma^S\left(A_2\right)\right\}}{\Gamma^S\left(A^{\text{HARD}}\right)} = \frac{1}{2} + \frac{\varepsilon}{0.6}.$$

Setting ϵ small enough, we can see that the approximation factor is very close to $1/2 = 1/\lambda$.

Appendix B. Computational Aspects

A naïve implementation of the PeerReview4All algorithm has a polynomial computational complexity (under either an arbitrary choice or one computable in polynomial-time in Step 6) and requires $\mathcal{O}\left(\lambda m^2 n\right)$ iterations of the max-flow algorithm. There are a number of additional ways that the algorithm may be optimized for improved computational complexity while retaining all the approximation and statistical guarantees.

One may use Orlin's method (Orlin, 2013; King et al., 1992) to compute the max-flow which yields a computational complexity of the entire algorithm at most $\mathcal{O}\left(\lambda(m+n)m^3n^2\right)$. Instead of adding edges is Step 3 of the subroutine one by one, a binary search may be implemented, reducing the number of max-flow iterations to $\mathcal{O}\left(\lambda m \log mn\right)$ and the total complexity to $\widetilde{\mathcal{O}}\left(\lambda(m+n)m^2n\right)$.

Finally, note that the max-min approximation guarantees (Theorem 1), as well as statistical results (Theorems 3 to 8 and corresponding corollaries) remain valid even for the assignment $\widetilde{A}$ computed in Step 3 of Algorithm 1 during the *first* iteration of the algorithm.

The algorithm may thus be stopped at any time after the first iteration if there is a strict time-deadline to be met. However, the results of Corollary 2 on optimizing the assignment for papers beyond the most worst-off will not hold any more.[9] The computational complexity of each of the iterations is at most $\widetilde{\mathcal{O}}\left(\lambda(m+n)mn\right)$, and stopping the algorithm after a constant number of iterations makes it comparable to the complexity of TPMS algorithm which is successfully implemented in many large scale conferences.

Let us now briefly compare the computational cost of PeerReview4All and ILPR algorithms. The full version of ILPR algorithm requires $\mathcal{O}(m^2)$ solutions of linear programming problems. Given that finding a max-flow in a graph constructed by our subroutine can be casted as linear programming problem (with constraints similar to those in Garg et al. 2010), we conclude that slightly optimized implementation of our algorithm results in $\mathcal{O}(\lambda m \log mn)$ solutions of linear programming problems, which is asymptotically better. To be fair, the ILPR algorithm also can be terminated in an earlier stage with theoretical guarantees satisfied, which brings both algorithms on a similar footing with respect to the computational complexity.

Appendix C. Topic Coverage

In this section we discuss an additional benefit of "topic coverage" that can be gained from the special choice of heuristic in Step 6 of Subroutine 1 of our PeerReview4All algorithm.

Research is now increasingly inter-disciplinary and consequently many papers submitted to modern conferences make contributions to multiple research fields and cannot be clearly attributed to any single research area. For instance, computer scientists often work in collaboration with physicists or medical researchers resulting in papers spanning different areas of research. Thus, it is important to maintain a broad topic coverage, that is, to ensure that such multidisciplinary papers are assigned to reviewers who not only have high similarities with the paper, but also represent the different research areas related to the paper. For example, if a paper proposes an algorithm to detect new particles in the CERN collider, then that paper should ideally be evaluated by competent physicists, computer scientists, and statisticians.

There are prior works both in peer-review (Long et al., 2013) and in text mining (Lin and Bilmes, 2011) which propose a submodular objective function to incentivize topic coverage. According to Long et al. (2013), the appropriate measure of coverage is a number of distinct topics of the paper covered, summed across the all papers. Let us introduce a piece of notation to formally describe the underlying optimization problem. For every paper $j \in [m]$, let $T(j) = \{t_1^{(j)}, \dots, t_{r_j}^{(j)}\}$ be related research topics and for every reviewer $i \in [n]$, let $T(i) = \{t_1^{(i)}, \dots, t_{r_i}^{(i)}\}$ be the topics of expertise of reviewer i. For every assignment A, we define $\omega(A)$ to be the total number of distinct topics of all papers covered by the assigned reviewers:

$$\omega(A) = \sum_{j \in [m]} \operatorname{card}\left(\bigcup_{i \in \mathcal{R}_A(j)} \Big(T(j) \bigcap T(i) \Big) \right), \tag{44}$$

9. If the algorithm is terminated after p' iterations, then bound (8) from Corollary 2 holds for $r \in [p']$.

where card($\mathcal{C}$) denotes the number of elements in the set $\mathcal{C}$. The goal in Long et al. (2013) is to find an assignment that maximizes $\omega(A)$ and respects the constraints on the paper/reviewer load. However, instead of the requirement that each paper is assigned to λ reviewers as in our work, Long et al. (2013) consider a relaxed version and require each paper to be reviewed by at most λ reviewers.

Using the submodular nature of the objective (44), Long et al. (2013) propose a greedy algorithm that is guaranteed to achieve a constant-factor approximation of the optimal coverage (44). This greedy algorithm, however, has the following two important drawbacks:

(i) Like the TPMS algorithm, the greedy algorithm aims at optimizing the global functional, and consequently may fare poorly in terms of fairness. Indeed, in order to optimize the global objective (44), the greedy algorithm may sacrifice the topic coverage for some of the papers, assigning relevant reviewers to other papers.

(ii) While guaranteed to achieve a constant factor approximation of the objective (44), the greedy algorithm may yield an assignment in which papers are reviewed by (much) less than λ reviewers. It is not even guaranteed that in the resulting assignment each paper has at least one reviewer.

Nevertheless, both the PeerReview4All algorithm and the algorithm of Long et al. (2013) can benefit from each other if the latter is used as a heuristic to choose a feasible assignment in Step 6 of the subroutine of the former. In what follows we detail the procedure to combine the two algorithms. The greedy algorithm of Long et al. (2013) picks (reviewer, paper) pairs one-by-one and adds them to the assignment. At each step, it picks the pair that yields the largest incremental gain to (44) while still meeting the paper/reviewer load constraints. In Step 6 of the subroutine of PeerReview4All, we may use the greedy algorithm, restricted to the (reviewer, paper) pairs added to the network in the previous steps, to find an assignment that approximately maximizes (44). Next, for every (reviewer, paper) pair that belongs to this assignment, we set the cost of the corresponding edge in the flow network to 1 and the costs of the remaining edges to 0. Finally, we compute the maximum flow with maximum cost in the resulting network and fix (reviewer, paper) pairs that correspond to edges employed in that flow in the final output of the subroutine.

Let us now discuss the benefits of this approach. First, in PeerReview4All we modify only the procedure of tie-breaking among max-flows, and hence all the guarantees established in the paper continue to hold. Second, the introduced procedure allows to overcome the issue (ii), because the max-flow guarantees that each paper is assigned with exactly requested number of reviewers. Third, by setting the cost of selected edges to 1, we encourage the topic coverage (although the approximation guarantee of the greedy algorithm no longer holds). Finally, we do not allow the algorithm of Long et al. (2013) to sacrifice some papers in order to maximize the global coverage (44), because the subroutine ensures that in the resulting assignment all the papers are assigned to pre-selected reviewers with high similarity, thereby overcoming (i).

References

A. Asadpour and A. Saberi. An approximation algorithm for max-min fair allocation of indivisible goods. *SIAM Journal on Computing*, 39(7):2970–2989, 2010. doi: 10.1137/

080723491. URL https://doi.org/10.1137/080723491.

Von Bakanic, Clark McPhail, and Rita J Simon. The manuscript review and decision-making process. *American Sociological Review*, pages 631–642, 1987.

Stefano Balietti, Robert L Goldstone, and Dirk Helbing. Peer review and competition in the art exhibition game. *Proceedings of the National Academy of Sciences*, 113(30):8414–8419, 2016.

Nikhil Bansal and Maxim Sviridenko. The Santa Claus problem. In *Proceedings of the Thirty-eighth Annual ACM Symposium on Theory of Computing*, STOC '06, pages 31–40, New York, NY, USA, 2006. ACM. ISBN 1-59593-134-1. doi: 10.1145/1132516.1132522. URL http://doi.acm.org/10.1145/1132516.1132522.

Salem Benferhat and Jerome Lang. Conference paper assignment. *International Journal of Intelligent Systems*, 16(10):1183–1192, 10 2001. ISSN 1098-111X. doi: 10.1002/int.1055.

Federico Bianchi and Flaminio Squazzoni. Is three better than one? Simulating the effect of reviewer selection and behavior on the quality and efficiency of peer review. In *Proceedings of the 2015 Winter Simulation Conference*, pages 4081–4089. IEEE Press, 2015.

Nick Black, Susan Van Rooyen, Fiona Godlee, Richard Smith, and Stephen Evans. What makes a good reviewer and a good review for a general medical journal? *Jama*, 280(3): 231–233, 1998.

Thomas Bonald, Laurent Massoulié, Alexandre Proutiere, and Jorma Virtamo. A queueing analysis of max-min fairness, proportional fairness and balanced fairness. *Queueing systems*, 53(1-2):65–84, 2006.

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

L. Charlin, R. S. Zemel, and C. Boutilier. A framework for optimizing paper matching. *CoRR*, abs/1202.3706, 2012. URL http://arxiv.org/abs/1202.3706.

Stephen Cole, Gary A Simon, et al. Chance and consensus in peer review. *Science*, 214 (4523):881–886, 1981.

T. M. Cover and J. A. Thomas. *Entropy, Relative Entropy, and Mutual Information*, pages 13–55. John Wiley & Sons, Inc., 2005. ISBN 9780471748823. doi: 10.1002/047174882X.ch2. URL http://dx.doi.org/10.1002/047174882X.ch2.

W. Dai, G. Z. Jin, J. Lee, and M. Luca. Optimal aggregation of consumer ratings: An application to yelp.com. Working Paper 18567, National Bureau of Economic Research, November 2012. URL http://www.nber.org/papers/w18567.

Edzard Ernst and Karl-Ludwig Resch. Reviewer bias: a blinded experimental study. *The Journal of laboratory and clinical medicine*, 124(2):178–182, 1994.

T Fiez, N Shah, and L Ratliff. A SUPER* algorithm to optimize paper bidding in peer review. In *ICML workshop on Real-world Sequential Decision Making: Reinforcement Learning And Beyond*, 2019.

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. *SIGKDD Explor. Newsl.*, 11(2):63–67, May 2010. ISSN 1931-0145. doi: 10.1145/1809400.1809413. URL http://doi.acm.org/10.1145/1809400.1809413.

Yang Gao, Steffen Eger, Ilia Kuznetsov, Iryna Gurevych, and Yusuke Miyao. Does my rebuttal matter? insights from a major nlp conference. *arXiv preprint arXiv:1903.11367*, 2019.

Robert S. Garfinkel. Technical note. An improved algorithm for the bottleneck assignment problem. *Operations Research*, 19(7):1747–1751, 1971. doi: 10.1287/opre.19.7.1747.

N. Garg, T. Kavitha, A. Kumar, K. Mehlhorn, and J. Mestre. Assigning papers to referees. *Algorithmica*, 58(1):119–136, Sep 2010. ISSN 1432-0541. doi: 10.1007/s00453-009-9386-0. URL https://doi.org/10.1007/s00453-009-9386-0.

Hong Ge, Max Welling, and Zoubin Ghahramani. A Bayesian model for calibrating conference review scores, 2013. URL http://mlg.eng.cam.ac.uk/hong/unpublished/nips-review-model.pdf. http://mlg.eng.cam.ac.uk/hong/unpublished/nips-review-model.pdf [Online; accessed 13-Nov-2019].

Judy Goldsmith and Robert H. Sloan. The AI conference paper assignment problem. WS-07-10:53–57, 12 2007.

Ellen L. Hahne. Round-robin scheduling for max-min fairness in data networks. *IEEE Journal on Selected Areas in communications*, 9(7):1024–1039, 1991.

David Hartvigsen, Jerry C. Wei, and Richard Czuchlewski. The conference paper-reviewer assignment problem. *Decision Sciences*, 30(3):865–876, 1999. ISSN 1540-5915. doi: 10.1111/j.1540-5915.1999.tb00910.x. URL http://dx.doi.org/10.1111/j.1540-5915.1999.tb00910.x.

Maryam Karimzadehgan, ChengXiang Zhai, and Geneva Belford. Multi-aspect expertise matching for review assignment. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 1113–1122, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-991-3. doi: 10.1145/1458082.1458230. URL http://doi.acm.org/10.1145/1458082.1458230.

Steven Kerr, James Tolliver, and Doretta Petree. Manuscript characteristics which influence acceptance for management and social science journals. *Academy of Management Journal*, 20(1):132–141, 1977.

V. King, S. Rao, and R. Tarjan. A faster deterministic maximum flow algorithm. In *Proceedings of the Third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '92, pages

157–164, Philadelphia, PA, USA, 1992. Society for Industrial and Applied Mathematics. ISBN 0-89791-466-X. URL http://dl.acm.org/citation.cfm?id=139404.139438.

Ari Kobren, Barna Saha, and Andrew McCallum. Paper matching with local fairness constraints. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 1247–1257, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-6201-6. doi: 10.1145/3292500.3330899. URL http://doi.acm.org/10.1145/3292500.3330899.

Michèle Lamont. *How professors think*. Harvard University Press, 2009.

John Langford. ICML acceptance statistics, 2012. http://hunch.net/?p=2517 (visited on 05/15/2018).

Ron Lavi, Ahuva Mu'Alem, and Noam Nisan. Towards a characterization of truthful combinatorial auctions. In *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, pages 574–583. IEEE, 2003.

N. Lawrence and C. Cortes. The NIPS Experiment. http://inverseprobability.com/2014/12/16/the-nips-experiment, 2014. [Online; accessed 3-June-2017].

J. K. Lenstra, D. B. Shmoys, and É. Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical Programming*, 46(1):259–271, Jan 1990. ISSN 1436-4646. doi: 10.1007/BF01585745. URL https://doi.org/10.1007/BF01585745.

V. I. Levenshtein. Upper-bound estimates for fixed-weight codes. *Problemy Peredachi Informatsii*, 7(4):3–12, 1971.

Lei Li, Yan Wang, Guanfeng Liu, Meng Wang, and Xindong Wu. Context-aware reviewer assignment for trust enhanced peer review. *PLOS ONE*, 10(6):1–28, 06 2015. doi: 10.1371/journal.pone.0130493. URL https://doi.org/10.1371/journal.pone.0130493.

Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 510–520, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-87-9. URL http://dl.acm.org/citation.cfm?id=2002472.2002537.

Xiang Liu, Torsten Suel, and Nasir Memon. A robust model for paper reviewer assignment. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 25–32, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2668-1. doi: 10.1145/2645710.2645749. URL http://doi.acm.org/10.1145/2645710.2645749.

Cheng Long, Raymond Wong, Yu Peng, and Liangliang Ye. On good and fair paper-reviewer assignment. In *Proceedings - IEEE International Conference on Data Mining, ICDM*, pages 1145–1150, 12 2013. ISBN 978-0-7695-5108-1.

Michael J Mahoney. Publication prejudices: An experimental study of confirmatory bias in the peer review system. *Cognitive therapy and research*, 1(2):161–175, 1977.

M. McGlohon, N. Glance, and Z. Reiter. Star quality: Aggregating reviews to rank products and merchants. In *Proceedings of Fourth International Conference on Weblogs and Social Media (ICWSM)*, 2010.

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

David Mimno and Andrew McCallum. Expertise modeling for matching papers with reviewers. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 500–509, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-609-7. doi: 10.1145/1281192.1281247. URL http://doi.acm.org/10.1145/1281192.1281247.

Ritesh Noothigattu, Nihar Shah, and Ariel Procaccia. Choosing how to choose papers. *arXiv preprint arxiv:1808.09057*, 2018.

James B. Orlin. Max flows in $\mathcal{O}(nm)$ time, or better. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 765–774, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2029-0. doi: 10.1145/2488608.2488705. URL http://doi.acm.org/10.1145/2488608.2488705.

Alan L Porter and Frederick A Rossini. Peer review of interdisciplinary research proposals. *Science, technology, & human values*, 10(3):33–38, 1985.

John Rawls. *A theory of justice: Revised edition*. Harvard university press, 1971.

Marko A. Rodriguez and Johan Bollen. An algorithm to determine peer-reviewers. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 319–328, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-991-3. doi: 10.1145/1458082.1458127. URL http://doi.acm.org/10.1145/1458082.1458127.

Marko A Rodriguez, Johan Bollen, and Herbert Van de Sompel. Mapping the bid behavior of conference referees. *Journal of Informetrics*, 1(1):68–82, 2007.

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

Mehdi S.M. Sajjadi, Morteza Alamgir, and Ulrike von Luxburg. Peer grading in a course on algorithms and data structures: Machine learning algorithms do not improve over simple baselines. In *Proceedings of the Third (2016) ACM Conference on Learning @ Scale*, L@S '16, pages 369–378, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3726-7. doi: 10.1145/2876034.2876036. URL http://doi.acm.org/10.1145/2876034.2876036.

N. B. Shah and M. J. Wainwright. Simple, robust and optimal ranking from pairwise comparisons. *CoRR*, abs/1512.08949, 2015. URL http://arxiv.org/abs/1512.08949.

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.

Flaminio Squazzoni and Claudio Gandelli. Saint Matthew strikes again: An agent-based model of peer review and the scientific community structure. *Journal of Informetrics*, 6 (2):265–275, 2012.

Ivan Stelmakh, Nihar Shah, and Aarti Singh. On testing for biases in peer review. In *NeurIPS*, 2019a.

Ivan Stelmakh, Nihar B. Shah, and Aarti Singh. Peerreview4all: Fair and accurate reviewer assignment in peer review. In Aurélien Garivier and Satyen Kale, editors, *Proceedings of the 30th International Conference on Algorithmic Learning Theory*, volume 98 of *Proceedings of Machine Learning Research*, pages 828–856, Chicago, Illinois, 22–24 Mar 2019b. PMLR.

Wenbin Tang, Jie Tang, and Chenhao Tan. Expertise matching via constraint-based optimization. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '10, pages 34–41, Washington, DC, USA, 2010. IEEE Computer Society. ISBN 978-0-7695-4191-4. doi: 10.1109/WI-IAT.2010.133. URL `http://dx.doi.org/10.1109/WI-IAT.2010.133`.

C. J. Taylor. On the optimal assignment of conference papers to reviewers. Technical report, Department of Computer and Information Science, University of Pennsylvania, 2008.

Warren Thorngate and Wahida Chowdhury. By the numbers: Track record, flawed reviews, journal space, and the fate of talented authors. In *Advances in Social Simulation*, pages 177–188. Springer, 2014.

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

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.

H. D. Tran, G. Cabanac, and G. Hubert. Expert suggestion for conference program committees. In *2017 11th International Conference on Research Challenges in Information Science (RCIS)*, pages 221–232, May 2017. doi: 10.1109/RCIS.2017.7956540.

G David L Travis and Harry M Collins. New light on old boys: Cognitive and institutional particularism in the peer review system. *Science, Technology, & Human Values*, 16(3): 322–341, 1991.

Jingyan Wang and Nihar B Shah. Your 2 is my 1, your 3 is my 9: Handling arbitrary miscalibrations in ratings. In *AAMAS*, 2019.

Yichong Xu, Han Zhao, Xiaofei Shi, and Nihar Shah. On strategyproof conference review. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, 2019a.

Yichong Xu, Han Zhao, Xiaofei Shi, Jeremy Zhang, and Nihar Shah. On strategyproof conference review. *Arxiv preprint*, 2019b.