Group Fairness in Peer Review

Haris Aziz
UNSW Sydney
Sydney, Australia
haris.aziz@unsw.edu.au

Evi Micha
University of Toronto
Toronto, Canada
emicha@cs.toronto.edu

Nisarg Shah
University of Toronto
Toronto, Canada
nisarg@cs.toronto.edu

ABSTRACT

In the past few years, conferences like AAAI and NeurIPS have grown tremendously. While on the one hand this has attracted submissions from a large number of communities, on the other hand it has also resulted in a poor reviewing experience for some communities, whose submissions end up being assigned to less qualified reviewers outside of their communities. An often-advocated solution is to break up such large conferences into smaller conferences to decentralize the reviewing process. However, this can lead to isolation of various communities and slower emergence of truly interdisciplinary ideas.

In this work, we tackle this challenge by introducing a notion of group fairness, namely the core, to the peer review setting. A reviewing assignment is in the core if every subset of researchers (a possible community) is treated in such a manner such that they cannot achieve a better outcome by breaking off and organizing a smaller conference on their own.

We study a simple peer review model, prove that it always admits a reviewing assignment in the core, and design an efficient algorithm to find one such assignment. On the negative side, we show that the core is incompatible with achieving a good worst-case approximation of social welfare, an often-sought desideratum. We complement these results by conducting experiments with real data. We observe that our algorithm, in addition to satisfying the core, generates good social welfare on average. In contrast, existing review assignment systems violate the core, treat many communities unfairly, and significantly incentivize them to disengage.

1 INTRODUCTION

Computer Science is a rapidly advanced field, and therefore peer-reviewed conferences are at the heart of the research progress, since their reviewing time is usually quite fast [6, 29]. In many of these conferences, such as AAMAS, AAAI, and NeurIPS, the assignment of the papers to reviewers is usually an automated procedure, due to their massive scale. Famous automated systems that are used in practise are the Toronto Paper Matching System [1], Microsoft CMT[1], and OpenReview[2]. The authors of the submissions are usually very interested to receive useful feedback from their peers, regarding how they could improve their paper [13, 20, 28]. Naturally, the overall experience of an author for a peer review procedure highly depends on the quality of the reviews that her manuscripts receive.

In many large conferences, the typical procedure of selecting the reviewers of each manuscript is the following one. First, for each paper-reviewer pair is calculated a similarity score based on various parameters such as the subject area of the paper and the reviewer, the bidding of the reviewer, etc. [1, 11, 12, 18, 27]. Then, an assignment is calculated through an optimization problem where the usual objectives are either to maximize the utilitarian social welfare, which is equal to the total similarity, or the egalitarian social welfare, which is equal to the minimum score of each submission, subject to constraints related to the total number of papers that each reviewer can review and the total number of reviewers that each paper should be assigned to. Under both objectives, it is possible that the review quality on some papers is sacrificed. To see that, consider the case that there are four submissions, p_1, p_2, p_3 and p_4, and four reviewers, 1, 2, 3 and 4 who can review up one paper each.

	1/2	3/4
p_1	1	ϵ
p_2	1	ϵ
p_3	0.9	0
p_4	0.9	0

Assume that the first two reviewers have similarity score equal to 1 for p_1 and p_2 and equal to 0.9 for p_3 and p_4, while the other two reviewers have similarity score equal to a negligible quantity $\epsilon > 0$ for p_1 and p_2 and have zero similarity score for p_3 and p_4. This may happen, when 1 and 2 work on topics that these papers consider, while 3 and 4 belong in a different community. If the goal is to maximize the utilitarian social welfare, then p_1 and p_2 are assigned to the first two reviewers, while p_3's and p_4's utilities are completely sacrificed, while if the goal is to maximize the egalitarian social welfare, the opposite happens. Papers that are assigned to inappropriate reviewers may receive poor feedback or even may be unfairly rejected, which may cause their authors to be significantly unsatisfied with the whole procedure. Thus, finding reviewing assignments that are fair is very important, and the last years researchers have focused in this direction [19].

Peng et al. [17] recently mentioned that a major problem with the prestigious mega conferences is that they constitute the main venues for several communities, and as a result, in some cases, people are asked to review submissions that are beyond their main areas of work. They claim that a reasonable solution is to move to a de-centralized publication process by creating more specialized conferences appropriate for different communities. In particular, they say that by this way "*Reviewers and reviewees will be peers, collaborators, and problem-specific interlocutors, not generic members of a large anonymized community.*". However, this solution could cause the isolation of different communities which in its turn could cause various other problems such as the difficulty of emerging interdisciplinary ideas. Moreover, it usually takes several years until a conference becomes famous and acceptable across the members of a community. So, a reasonable question is

> ...how can we treat each group of researchers in a fair way in the current review and publication processes?

To answer this question, we use the concept of fairness, which, to the best of our knowledge, we are the first that introduce in a peer review setting, called *core* [5]. In this context, this notion requires that given an assignment there is no subset of authors– who can

[1] https://cmt3.research.microsoft.com/
[2] https://github.com/openreview/openreview-matcher

also serve as reviewers– that can deviate as following: They can find an assignment of their submissions among themselves such that

- no author reviews her own submissions,
- each paper is reviewed by as many reviewers as in the given assignment,
- each reviewer reviews no more papers than in the given assignment, and
- the submissions of each author are assigned to better reviewers than in the given assignment.

Intuitively, this notion of fairness requires that any group of authors is treated in a way that it does not have any incentive to deviate from the given assignment and create its own assignment that meets the constraints of the peer review procedure. In other words, any sub-community in a big conference is treated in a way that it does not have any incentive to deviate from the conference and create its own smaller conference. Note that this definition provides fairness to *every* sub-community and not only to pre-defined ones, and as result it guarantees that even emerged interdisciplinary subcommunities, are treated fairly.

1.1 Our Contribution

In this work, we consider the case that each submission is authored by one agent that also serves as reviewer. A reviewing assignment is valid if each paper is reviewed by k_p reviewers, each reviewer reviews up to k_a papers and no agent reviews her own submissions. To ensure that a valid assignment always exists, we assume that the maximum number of papers that each agent can submit is at most $\lfloor k_a/k_p \rfloor$.

In Section 3, we present an efficient algorithm that always returns a valid assignment in the core under very minor assumptions regarding the preferences of the authors for different potential reviewers. In particular, we assume that each author holds an ordinal preference over the reviewers with respect to each of her submissions and the extension of these preferences to preferences over sets of reviewers that review her submissions follows some very natural properties.

In Section 4, we show that there are instances where no assignment in the core can provide an approximation better than $\Omega(n)$ with respect to the utilitarian social welfare and bounded approximation with respect to the egalitarian social welfare. Moreover, we show that it is NP-hard to find an assignment in the core with maximum utilitarian social welfare and an assignment in the core that provides bounded approximation to the best egalitarian welfare that can be achieved by any assignment in the core.

In Section 5, we conduct experiments with real data and observe that our algorithm achieves good utilitarian and egalitarian social welfare in the average case, while broadly applied methods fail to find assignments in the core, and as a result communities are incentivized to deviate.

1.2 Related Work

The reviewing assignment problem has been extensively studied [30]. Toronto Paper Matching System [1] which is a very broadly applied method focuses on maximizing the utilitarian welfare and this approach has been adopted by other popular conference management systems such as EasyChair[3] and HotCRP [4] [25]. O'Dell et al. [14] got a different approach where the goal is maximize the minimum total utility that a paper gets, and Stelmakh et al. [25] generalized this approach by maximizing the minimum paper score, then maximizing the next smallest paper score, etc. One of the key issues in reviewer assignment is to ensure that the assignment is fair and efficient for the reviewers as well as the papers/authors. Several papers have examined this issue in different respects (see, e.g., [4, 8, 10, 16, 21]). The core property we focus on can also be viewed as a fairness or efficiency requirement.

Assignment of papers to reviewers is essentially a matching problem and hence has connections with several classical problems in matching. Our model is related to exchange problems with endowments. Agents can be viewed as being endowed by their own papers which they wish to exchange with other agents. In contrast to classical exchange problem with endowments, our model has a distinctive requirement that agents need to give away *all* their items/papers as the papers need to be reviewed by the agent who gets the paper. The difference is crucial as explained next.

A basic exchange problem the Shapley-Scarf housing market in which each agent owns one house. Shapley and Scarf [22] showed that a simple yet elegant mechanism called *Gale's Top Trading Cycle (TTC)* finds an allocation which is in the core. TTC is based on multi-way exchanges of houses between agents. Since the basic assumption in the model is that agents have strict preferences over houses, TTC is also strict core selecting and therefore Pareto optimal. Our model involves agents getting multiple items. For problems with multiple endowments, Konishi et al. [9] showed that the core can be empty under additive valuations. Note our problem is different as individual rationality has no bite in our context and an agent is required to give away all of his 'resources' (own papers).

Our model also has connections with matching with two-sided preferences where agents have preferences over reviewers. In many to many matchings with two-sided preferences, several solution concepts have been used to identify desirable matchings. The classical concept of *pairwise stability* requires that there are no two agents who are not partners, but by becoming partners, possibly dissolving some of their partnerships to remain within their quotas and possibly keeping other ones, can both obtain a strictly preferred set of partners. A matching *corewise-stable* if there is no subset of agents who by forming all their partnerships only among themselves, can all obtain a strictly preferred set of partners. A matching μ will be called *setwise-stable (SW)* if there is no subset of agents who by forming new partnerships only among themselves, possibly dissolving some partnerships of μ to remain within their quotas and possibly keeping other ones, can all obtain a strictly preferred set of partners [23]. For a detailed taxonomy of stability concepts for matching for many to many matchings, see the paper by Klaus and Walzl [7].

2 MODEL

For $q \in \mathbb{N}$, we use $[q]$ to denote the set $\{1, \ldots, q\}$. There is a set of agents $N = [n]$ where each agent can serve as reviewer and

[3] https://easychair.org
[4] https://hotcrp.com

may also author some papers. Let $P_i = \{p_{i,1}, \dots, p_{i,m_i}\}$ be the set of submissions of agent i where $m_i \in \mathbb{N}$ and $P = (P_1, \dots, P_n)$. We call $p_{i,\ell}$ as the ℓ-th submission of agent i. Whenever all the agents have just one submission, we drop ℓ and simply write p_i. Let us define $m = \sum_{i \in N} m_i$, i.e. m denotes the total number of submissions.

Preferences. Each agent $i \in N$ has a preference ranking over the agents in $N \setminus \{i\}$ with respect to her ℓ-th submission, denoted by $\sigma_{i,\ell}$.[5] This preference can be based on a mixture of factors, such as how qualified the other agents are to review her ℓ-th submission and how likely they are to provide a positive review. Let $\sigma_{i,\ell}(i')$ be the position of agent $i' \in N \setminus \{i\}$ in the ranking. An agent i prefers her submissions $p_{i,\ell}$ to be reviewed by i' rather than i'', if $\sigma_{i,\ell}(i') < \sigma_{i,\ell}(i'')$. Again, when all the agents have just one submission, we drop ℓ and just write σ_i. Let $\vec{\sigma} = (\sigma_{1,1}, \dots, \sigma_{1,m_1}, \dots, \sigma_{n,1}, \dots, \sigma_{n,m_n})$.

Typically, each paper receives multiple reviews; hence, we need to define the preferences of agents over sets of reviewers. When agent i prefers (resp., weakly prefers) her ℓ-th submission to be reviewed by the set of agents S rather than the set of agents S', we denote it by $S \succ_{i,\ell} S'$ (resp., $S \succeq_{i,\ell} S'$). We assume that this extension from preferences over individual agents to preferences over sets of agents satisfies the following natural property.

Definition 2.1 (Order Separability). Let $S_1, S_2, S_3 \subseteq N$ with $|S_1| = |S_2|$. If for each $i' \in S_1$ and each $i'' \in S_2$, it holds that $\sigma_{i,\ell}(i') < \sigma_{i,\ell}(i'')$, then $S_1 \cup S_3 \succ_{i,\ell} S_2 \cup S_3$.

Assignment. A *review assignment* (sometimes simply called as assignment) $R \in \{0,1\}^{n \times m}$ is a binary matrix such that $R(i,j) = 1$, if agent i is assigned to review submission j. With a slight abuse of notation, we denote with $R_i^a = \{j \in [m] : R(i,j) = 1\}$, i.e. the submissions that agent i reviews and with $R_j^p = \{i \in [n] : R(i,j) = 1\}$, i.e. the agents that review submission j. We say that a review assignment is *valid* if:

- For each $j \in [m]$, $|R_j^p| = k_p$, i.e. each paper is reviewed by k_p agents.
- For each $i \in [n]$, $|R_i^a| \le k_a$, i.e. each agent reviews at most k_a submissions.
- For each $i \in [n]$ and $\ell \in [m_i]$, $R(i, p_{i,\ell}) = 0$, i.e. no agent is assigned to review her own submissions.

To ensure that a valid assignment always exists, we impose the constraint that $m_i \cdot k_p \le k_a$ for each $i \in N$, which implies that $m \cdot k_p \le n \cdot k_a$.

Given $N' \subseteq N$ and $P_i' \subseteq P_i$ for each $i \in N'$ with $P' = \cup_{i \in N'} P_i'$, an assignment $\hat{R} \in \{0,1\}^{|N'| \times |P'|}$ that is *restricted* over N' and P' is called valid if each submission in P' is reviewed by k_p agents from N' and each agent in N' is assigned to review at most k_a submissions from $P' \setminus P_i'$.

Hereinafter, when referring to an assignment or a restricted assignment, we will assume validity, unless specified otherwise.

Preferences over assignments. Recall that $\succ_{i,\ell}$ (resp., $\succeq_{i,\ell}$) denotes the preferences (resp., weak preferences) of agent i over possible sets of reviewers to her ℓ-th submission. To extend this to preferences (resp., weak preferences) of the agent over assignments,

Algorithm 1 PeerReview-TTC$(N, P, \vec{\sigma}, k_a, k_p)$

Let $R(i,j) \leftarrow 0$ for every $i \in N$ and $j \in P$ ▹ Initialize an empty assignment
Construct the preference graph G_R
while $\exists$ cycle in G_R **do**
  Eliminate the cycle
  Update $\bar{P}_i$-s by removing any completely assigned paper
  Update G_R
end while
$U \leftarrow \{i \in N : \bar{P}_i \neq \emptyset\}$
$L \leftarrow$ the last $k_p - |U| + 1$ agents in $N \setminus U$ to have all their submissions completely assigned
Return R, L, U

denoted by $\succ_i$ (resp., $\succeq_i$), we need to collate her preferences across all her submissions. We simply require that the collated preference extension satisfies the following natural property.

Definition 2.2 (Consistency). Let R be an assignment, $\hat{R}$ be an assignment restricted over $N' \subseteq N$ and $P' = \cup_{i \in N'} P_i'$, where $P_i' \subseteq P_i$ for each $i \in N'$, and $i \in N'$ be an agent. If $R^p_{p_{i,\ell}} \succeq_{i,\ell} \hat{R}^p_{p_{i,\ell}}$ for each $p_{i,\ell} \in P_i'$, then we must have $R \succeq_i \hat{R}$.

Core. In this work, we are interested in finding assignments such that no subset of agents has an incentive to deviate with any subset of their submissions and implement a restricted assignment that each deviating agent prefers. Formally:

Definition 2.3 (Core). An assignment R is in the core if there is no $N' \subseteq N$, $P_i' \subseteq P_i$ for each $i \in N'$, and assignment $\hat{R}$ restricted over N' and $P' = \cup_{i \in N'} P_i'$ such that $\hat{R} \succ_i R$ for each $i \in N'$.

Note that a core assignment R guarantees no subset of agents simultaneously finds an alternative assignment $\hat{R}$ restricted to them and a subset of their submissions strictly better according to *any preference extension* satisfying order separability and consistency. In other words, every deviating agent must find R weakly better than $\hat{R}$ for every single one of her submissions in the deviation simultaneously, which is a demanding guarantee for the assignment R to provide. Nonetheless, our main result is that an assignment with such a strong guarantee always exists.

3 ASSIGNMENT IN THE CORE

In this section, we show that when the preferences of the agents are order separable and consistent, an assignment in the core always exists and can be found in polynomial time. The main algorithm (Algorithm 3), which is called PRCore, uses Algorithm 1 and Algorithm 2. Algorithm 1 is quite similar to the Top-Trading-Cycles (TTC) mechanism. After the execution of Algorithm 1, it is possible that there are submissions that have not been assigned to k_p agents, but either no agent can review more submissions or all the agents that can review more submissions already review them. In this case, we call Algorithm 2 which ensures that each submission is reviewed by k_p agents, by making tweaks in the assignment. It is important to mention that assignments, that were established in the execution of Algorithm 1, may be removed in the execution of Algorithm 2.

[5] Our algorithms continue to work with weak orders; one can arbitrarily break ties to convert them into strict orders before feeding them into our algorithms.

Reviewers \ Rounds	1	2	3	4	5
1 :	p_3	p_2	p_4		
2 :	p_1	p_3		p_6	
3 :	p_2	p_1			p_4
4 :			p_1		p_5
5 :				p_2	p_3
6 :				p_5	

(a) Execution of PeerReview-TTC

Reviewers \ Rounds	1 (Phase 1)	1 (Phase 2)	2 (Phase 2)
1 :	p_3, p_2, p_4	$\cancel{p_3}, p_6\ p_2, p_4$	$p_6\ p_2, \cancel{p_4}, p_5$
2 :	p_1, p_3, p_6	p_1, p_3, p_6	p_1, p_3, p_6
3 :	p_2, p_1, p_4	p_2, p_1, p_4	p_2, p_1, p_4
4 :	p_1, p_5, p_6	p_1, p_5, p_6	p_1, p_5, p_6
5 :	p_2, p_3	p_2, p_3	p_2, p_3, p_4
6 :	p_5, p_4	p_5, p_4, p_3	p_5, p_4, p_3

(b) Execution of Filling-Gaps

Figure 1: Execution of PRCore when $n = 6$, $k_p = k_a = 1$, $\sigma_1 = 2 \succ 3 \succ 4 \succ \ldots$, $\sigma_2 = 3 \succ 1 \succ 5 \succ \ldots$, $\sigma_3 = 1 \succ 2 \succ 5 \succ \ldots$, $\sigma_4 = 1 \succ 3 \succ 5 \succ \ldots$, $\sigma_5 = 6 \succ 4 \succ \ldots$ and $\sigma_6 = 2 \succ \ldots$. On the left table, we see the assignments that are established in each round of PeerReview-TTC by eliminating cycles. After the execution of PeerReview-TTC, three papers, p_4, p_5, p_6 are not completely assigned. Thus, $U = \{4, 5, 6\}$ and $L = \{3\}$. On the right table, we see that there is a cycle in the greedy graph which is eliminated at the first round of Phase 1. In Phase 2, where $\vec{\rho} = (6, 5)$, at the first round, since p_3 is authored by an agent in $U \cup L \setminus \{6\}$, is not reviewed by 6 and is completely assigned, p_3 is assigned to 6 while it is removed form 1 in which p_6 is now assigned. At the second round, since p_4 is authored by an agent in $U \cup L \setminus \{5\}$, is not reviewed by 5 and is completely assigned, p_4 is assigned to 5 while it is removed form 1 in which p_5 is now assigned.

Algorithm 2 Filling-Gaps$(N, P, \vec{\sigma}, k_a, k_p, R, L, U)$

Phase 1
Construct the greedy graph G_R
while $\exists$ cycle **do**
  Eliminate the cycle
  Update $\bar{P}_i$-s by removing any completely assigned paper
  Update U and L by moving any agent i from U to L if $\bar{P}_i = \emptyset$
  Update G_R
end while

Phase 2
Construct an order $\vec{\rho}$ over the agents in U such that $\forall i \in U \setminus \{\rho(1), \ldots, \rho(t-1), \rho(t)\}$ and $\forall p_{i,\ell} \in \bar{P}_i$, $R(\rho(t), p_{i,\ell}) = 1$
for $t \in [|U|]$ **do**
  while $\bar{P}_{\rho(t)} \neq \emptyset$ **do**
    Pick arbitrary $p_{\rho(t),\ell} \in \bar{P}_{\rho(t)}$
    Find completely assigned $p_{i',\ell'}$ such that $R(\rho(t), p_{i',\ell'}) = 0$ for some $i' \in U \cup L \setminus \{\rho(t)\}$
    Find $i'' \neq \rho(t)$ such that $R(i'', p_{\rho(t),\ell}) = 0$ and $R(i'', p_{i',\ell'}) = 1$
    $R(i'', p_{\rho(t),\ell}) \leftarrow 1$
    $R(i'', p_{i',\ell'}) \leftarrow 0$
    $R(\rho(t), p_{i',\ell'}) \leftarrow 1$
    Remove $p_{\rho(t),\ell}$ from $\bar{P}_{\rho(t)}$ if it is completely assigned
  end while
end for
Return R

Before we describe the algorithms in detail, let us introduce some more notation. Let $m^* = \max_{i \in N} m_i$. For reasons that will become clear later, we want to ensure that $m_i = m^*$, for each $i \in N$. To achieve that, we add $m^* - m_i$ dummy submissions to agent i, denoted by $p_{i,m_i+1}, p_{i,m_i+2}, \ldots, p_{i,m^*}$, and the rankings over reviewers with

---

Algorithm 3 PRCore$(N, P, \vec{\sigma}, k_a, k_p)$

R, L, U =PeerReview-TTC$(N, P, \vec{\sigma}, k_a, k_p)$
if $|U| > 0$ **then**
  R =Filling-Gaps$(N, P, \vec{\sigma}, k_a, k_p, R, L, U)$
end if
Return R

---

respect to these submissions are arbitrary. An assignment is called *partial* if no agent reviews more than k_a submissions, but there are submissions that are reviewed by less than k_p agents. A submission that is reviewed by k_p agents under a partial assignment is called *completely assigned* and *incompletely assigned*, otherwise. We denote with $\bar{P}_i(\hat{R})$ the set of submissions of i that are incompletely assigned under a partial assignment $\hat{R}$, i.e. $\bar{P}_i(\hat{R}) = \{p_{i,\ell} \in P_i : |\hat{R}^P_{p_{i,\ell}}| < k_p\}$. Let $\bar{P}(\hat{R}) = (\bar{P}_1(\hat{R}), \ldots, \bar{P}_n(\hat{R}))$. We omit $\hat{R}$ from the notation when it is clear from context.

In order to define the PeerReview-TTC algorithm (Algorithm 1), we first need to introduce the notion of a *preference graph*. Suppose we have a partial assignment $\hat{R}$. Each agent i with $\bar{P}_i \neq \emptyset$ picks one of her incompletely assigned submissions arbitrarily. Without loss of generality, we assume that she picks her ℓ^*-th submission. We define the directed preference graph $G_{\hat{R}} = (N, E_{\hat{R}})$ where each agent is a node and for each i with $\bar{P}_i \neq \emptyset$, $(i, i') \in E_{\hat{R}}$ if and only if i' is ranked higher in σ_{i,ℓ^*} among the agents that don't review p_{i,ℓ^*} and review less than k_a submissions. In other words, each agent points her most preferred agent with respect to p_{i,ℓ^*} that does not already review it and can review at least one more submission. Moreover, for each $i \in N$ with $\bar{P}_i = \emptyset$, we add an edge from i to i', where i' is an arbitrary agent with $\bar{P}_{i'} \neq \emptyset$, i.e. each agent, whose all submissions are completely assigned, points an arbitrary agent whose some of her submissions are not completely assigned.

PeerReview-TTC starts with an empty assignment, constructs the preference graph and searches for a directed cycle in the graph.

If such a cycle exists, the algorithm eliminates it as following: For each (i, i') that is included in the cycle, it assigns submission p_{i,ℓ^*} to i' (if i's submissions are already completely assigned, it does nothing) and removes p_{i,ℓ^*} from $\bar{P}_i$, if it is now completely assigned. Then, the algorithm updates the preference graph and continues to eliminate cycles in the same way. When there are no left cycles in the preference graph, the algorithm terminates and returns a set U that contains all the agents that some of their submissions are incompletely assigned and a set L that contains the last $k_p - |U| + 1$ agents whose all submissions became completely assigned.

PRCore first calls PeerReview-TTC and if U is non-empty it also calls Algorithm 2, called Filling-Gaps, to ensure that the final assignment is valid. Before we describe the algorithm, we also need to introduce the notion of a *greedy graph*. Suppose that we have a partial assignment $\hat{R}$ which indicates a set U that contains all the agents whose at least one submission is incompletely assigned. We define the directed greedy graph $G_{\hat{R}} = (U, E_{\hat{R}})$ where $(i, i') \in E_{\hat{R}}$ if $\hat{R}(i', p_{i,\ell}) = 0$ for some $p_{i,\ell} \in \bar{P}_i$. In other words, while in the preference graph, agent i points only to her favourite potential reviewer with respect to one of her incomplete submissions, in the greedy graph agent i points to any agent in $U \setminus \{i\}$ that could review at least one of her submissions that is incompletely assigned.

Filling-Gaps consists of two phases. In the first phase, starting from the partial assignment R that was created in Algorithm 1, it constructs the greedy graph, searches for cycles and eliminates a cycle by assigning submission $p_{i,\ell}$ to agent i' for each (i, i') in the cycle that exists due to $p_{i,\ell}$ (when an edge exists due to multiple submissions, the algorithm chooses one of them arbitrary). Then, it updates $\bar{P}_i$ be removing any $p_{i,\ell}$ that became completely assigned and also updates U be moving any i to L if $\bar{P}_i$ became empty. It continues by updating the greedy graph and eliminating cycles in the same away. When no more cycles exist in the greedy graph, if U is empty, the algorithm terminates. Otherwise, the algorithm starts the second phase, where in $|U|$ rounds ensures that the incomplete submissions of each agent become completely assigned as following. It constructs an order $\vec{\rho}$ over the agents in U, that has some specific properties, and in round $t \in [|U|]$, for each incomplete assigned $p_{\rho(t),\ell}$, it finds a completely assigned submission j' that is authored by some agent in $U \cup L \setminus \{\rho(t)\}$ and is not reviewed by $\rho(t)$ and an agent i that reviews j' but not $p_{\rho(t),\ell}$, and then, it moves j' from i to $\rho(t)$ and assigns $p_{\rho(t),\ell}$ to i. Figure 1 shows the execution of PRCore in a small instance.

Theorem 3.1. *RPCore returns an assignment in the core, in polynomial time.*

Proof. First, in the next lemma, we show that the assignment that the algorithm returns is valid.

Lemma 3.2. *RPCore returns a valid assignment.*

Proof. First, note that in Algorithm 1, if an agent i with $\bar{P}_i \neq \emptyset$ is assigned one submission due to the elimination of a cycle, then we know that one of her submissions that is incompletely assigned is also assigned to an agent that does not review it already. Hence, we see that until $\bar{P}_i$ becomes empty, we have that

$$|R_i^a| = \sum_{j \in [m^*]} |R_{p_{i,j}}^p|. \tag{1}$$

Since $\sum_{j \in [m^*]} |R_{p_{i,j}}^p| = m^* \cdot k_p \leq k_a$, we get that i is not assigned more than k_a papers to review until the point where all of her submissions become completely-assigned. After that point, the agent may still participate in a cycle as long as she reviews strictly less than k_a submissions. Thus, we see that if Algorithm 1 terminates with an empty set U, i.e. all the submissions are completely assigned, the assignment that it returns is valid, as each submission is reviewed by k_p different agents, each agent reviews at most k_a submissions and no author reviews her own submissions.

Now, suppose that Algorithm 1 returns a non-empty set U. First, we show that $|U| \leq k_p$. For each $i \in U$, from Equation (1), we know that

$$|R_i^a| = \sum_{j \in [m^*]} |R_{p_{i,j}}^p| < m^* \cdot k_p \leq k_a \tag{2}$$

as there exists at least one submission of i that is assigned to less than k_p agents. Hence, we get that i can review more submissions. Now, suppose for contradiction that at the last iteration of the algorithm, each agent $i \in U$ has an outgoing edge in the preference graph. In this case, we claim that there exists a directed cycle in the preference graph which is a contradiction as Algorithm 1 would have not been terminated yet. To see that, note that each outgoing edge of an agent $i \in U$ either goes to another agent $i' \in U$, since i' can review more submissions, or goes to an agent $i' \notin U$ whose all submissions are completely-assigned. In the latter case, i' has an outgoing edge to an agent in U by the definition of the preference graph. Thus, starting from any agent in U, we conclude in an agent in U and eventually we would found a cycle. Therefore, we have that there exists an agent $i^* \in U$ that at the last iteration of the algorithm arbitrary picks her incomplete submission p_{i^*,ℓ^*} and does not have any outgoing edge to any other agent. This means that all the agents that can review more submissions, already review p_{i^*,ℓ^*}. Since all the agents in $U \setminus \{i^*\}$ can review more submissions, we get that all of them are assigned p_{i^*,ℓ^*}. But since p_{i^*,ℓ^*} is not completely assigned, we conclude that $|U \setminus \{i^*\}| < k_p$. Therefore, we have that $|L| \geq 1$ and from their definitions, we get that $|U \cup L| = k_p + 1$.

When U is non empty, PRCore calls Filling-Gaps. This algorithm first eliminates cycles in the greedy graph. With similar arguments as in the elimination of cycles in the preference graph, we conclude that during and after the first phase of Filling-Gaps, Equation (1) is still true for any $i \in N$ with $\bar{P}_i \neq \emptyset$. When no more cycles exist and U is still non empty, the algorithm constructs an order over the agents U, denoted by $\vec{\rho}$, such that $\rho(t)$ reviews all the incompletely assigned submissions of each $i \in U \setminus \{\rho(1), \ldots, \rho(t-1), \rho(t)\}$.

To see why such an order exists, first note that the greedy graph is a DAG, since it has no cycles. If we construct the topological ordering of the DAG over the nodes in U, denoted by $\vec{\rho}$, we get that no $i \in U \setminus \{\rho(1), \ldots, \rho(t-1), \rho(t)\}$ has an outgoing edge to $\rho(t)$. But since $\rho(t)$ can review more submissions and each such i has incomplete submissions, from the definition of the greedy graph, we get that $\rho(t)$ reviews all the incompletely assigned submissions of each i in $U \setminus \{\rho(1), \ldots, \rho(t-1), \rho(t)\}$.

After having created the order $\vec{\rho}$, Algorithm 2 runs $|U|$ rounds, where in round $t \in [|U|]$, it ensures that all the submissions of agent $\rho(t)$ become completely assigned as following. For each $p_{\rho(t),\ell} \in \bar{P}_{\rho(t)}$, it finds a completely assigned submission $p_{i',\ell'}$ of an agent $i' \in U \cup L \setminus \{\rho(t)\}$ that is not reviewed by $\rho(t)$. It also finds an agent i'' that reviews $p_{i',\ell'}$, but does not review $p_{\rho(t),\ell}$. Then, the algorithm assigns $p_{\rho(t),\ell}$ to i'', and moves $p_{i',\ell'}$ from i'' to $\rho(t)$.

Before, we show that there exist $p_{i',\ell'}$ and i'' with the desired properties, we show that it holds $|R^a_{\rho(t)}| = \sum_{j\in[m^*]} |R^p_{p_{\rho(t),j}}|$, before and during the execution of round t. We already know that this is true after the first phase of Filling-Gaps. In the second phase, note that until round t, if $\rho(t)$ is assigned a new submission to review, she is removed one of the old assigned submissions. Moreover, none of her incompletely assigned submissions is assigned to any agent. Hence, indeed before round t, we have the desired property. Now, note that when we execute step t, $\rho(t)$ is assigned one more submission to review and one of her incomplete submissions is assigned to a new agent. Thus, it is still true that $|R^a_{\rho(t)}| = \sum_{\ell\in[m^*]} |R^p_{p_{\rho(t),\ell}}|$.

Now, we show that as long as $\bar{P}_{\rho(t)}$ is non-empty, there exists a completely assigned submission $p_{i',\ell'}$ of an agent $i' \in U\cup L\setminus\{\rho(t)\}$ that is not reviewed by $\rho(t)$. Note that at iteration t, all the submissions of each agent in $i' \in L \cup \{\rho(1), \dots, \rho(t-1)\}$ are completely assigned. Thus, any incompletely assigned submission, that does not belong to $\rho(t)$, belongs to some agent $i \in U \setminus \{\rho(1), \dots, \rho(t-1), \rho(t)\}$. But, we already know that $\rho(t)$ reviews any such submission. Moreover, we note that $\rho(t)$ cannot review all the submissions of all the agents in $U \cup L \setminus \{\rho(t)\}$. Indeed, if we assume for contradiction that $\rho(t)$ reviews all the submissions of all the agents in $U \cup L \setminus \{\rho(t)\}$, then we have that

$$|R^a_{\rho(t)}| = \sum_{\ell\in[m^*]} |R^p_{p_{\rho(t),\ell}}| = k_p \cdot m^*,$$

since $|U \cup L \setminus \{\rho(t)\}| = k_p$ and each of them has m^* submissions, which would imply that all the submissions of $\rho(t)$ are completely assigned since $\rho(t)$ has m^* submissions and each of them should be assigned to k_p reviewers. Hence, we get that since $\rho(t)$ reviews all the incompletely assigned submissions but cannot review all the submissions of all agents in $i \in U \cup L \setminus \{\rho(t)\}$, there exists a completely assigned submission that belongs to some $i' \in U \cup L \setminus \{\rho(t)\}$ and is not reviewed by $\rho(t)$.

Next, we show that there exists i'' that reviews $p_{i',\ell'}$, but does not review $p_{\rho(t),\ell}$. Indeed, since $p_{i',\ell'}$ is reviewed by k_p agents and not from $\rho(t)$, while $p_{\rho(t),\ell}$ is reviewed by strictly less than k_p agents, it exists an agent that reviews the former submission but not the latter.

Note that after this assignment, any submission except for $p_{\rho(t),\ell}$ is assigned to the same number of reviewers as before this step and every agent except for $\rho(t)$ reviews the same number of submissions as before this step. Moreover, we see that after step t, it remains true that $|R^a_{\rho(t)}| = \sum_{\ell\in[m^*]} |R^p_{p_{\rho(t),\ell}}|$, as if $\rho(t)$ is assigned a new submission, an old assigned submission is removed, while if $p_{\rho(t),\ell}$ is assigned to a new reviewer, it is removed form another reviewer. Thus, we conclude that after the execution of Filling-Gaps, the assignment is valid. □

Now, we show that the final assignment R that PRCore returns is in the core. First, note that while it is probable that an assignment of a submission of an agent in $U \cup L$, that was established during the execution of Algorithm 1, to be removed in the execution of Algorithm 2, this never happens for submissions that belong to some agent in $N \setminus (U \cup L)$. Now, for the sake of contradiction, we assume that $N' \subseteq N$, with $P'_i \subseteq P_i$ for each $i \in N'$, deviate to a restricted assignment $\widetilde{R}$ over N' and $\cup_{i\in N'} P'_i$. Note that $\widetilde{R}$ is valid if and only if $|N'| > k_p$, as if $|N'| \leq k_p$, there is no way each submission in $\cup_{i\in N'} P'_i$ to be completely assigned, since no agent can review her own submissions.

Now, we distinguish into two cases.

Case I: $\exists i \in N' : i \notin L \cup U$. Let $i^* \in N'$ be the first agent in N' whose all submissions became completely assigned in the execution of PeerReview-TTC. Note that since there exists $i \notin U \cup L$, we get that $i^* \notin U \cup L$ from the definitions of U and L. Now, consider any $p_{i^*,\ell}$. Let $Q_1 = R^p_{p_{i^*,\ell}} \setminus (R^p_{p_{i^*,\ell}} \cap \widetilde{R}^p_{p_{i^*,\ell}})$ and $Q_2 = \widetilde{R}^p_{p_{i^*,\ell}} \setminus (R^p_{p_{i^*,\ell}} \cap \widetilde{R}^p_{p_{i^*,\ell}})$. If $Q_1 = \emptyset$, then we have that $R^p_{p_{i^*,\ell}} = \widetilde{R}^p_{p_{i^*,\ell}}$ which means that $R^p_{p_{i^*,\ell}} \succeq_{i^*,\ell} \widetilde{R}^p_{p_{i^*,\ell}}$. Otherwise, let

$$i' = \underset{i\in Q_1}{\operatorname{argmax}}\, \sigma_{i^*,\ell}(i),$$

i.e. i' is ranked at the lowest position in $\sigma_{i^*,\ell}$ among the agents that review $p_{i^*,\ell}$ under R but not under $\widetilde{R}$ and let

$$i'' = \underset{i\in Q_2}{\operatorname{argmin}}\, \sigma_{i^*,\ell}(i),$$

i.e. i'' is ranked at the highest position in $\sigma_{i^*,\ell}$ among the agents that review $p_{i^*,\ell}$ under $\widetilde{R}$ but not under R. We have $R(i', p_{i^*,\ell}) = 1$, if and only if i^* has an outgoing edge to i' at some round of PeerReview-TTC. At the same round, we get that i'' can review more submissions, since $i'' \in N'$ and if i^* has incompletely assigned submissions, then any $i \in N'$ has incompletely assigned submissions, and hence $|R^a_{i''}| < k_p \cdot m^* \leq k_a$. This means that if $\sigma_{i^*,\ell}(i') > \sigma_{i^*,\ell}(i'')$, then i^* would point i'' instead of i'. Hence, we conclude that $\sigma_{i^*,\ell}(i') < \sigma_{i^*,\ell}(i'')$. Then, from the definition of i' and i'' and from the order separability property we have that $R^p_{p_{i^*,\ell}} \succ_{i^*,\ell} \widetilde{R}^p_{p_{i^*,\ell}}$. Thus, either if Q_1 is empty or not, we have that for any $p_{i^*,\ell} \in P'_i$, it holds that $R^p_{p_{i^*,\ell}} \succeq_{i^*,\ell} \widetilde{R}^p_{p_{i^*,\ell}}$ and from consistency we get that $R \succeq_{i^*} \widetilde{R}$ which is a contradiction.

Case II: $\nexists i \in N' : i \notin L \cup U$. In this case we have that $N' = U \cup L$, as $|U \cup L| = k_p + 1$. This means that for each $i \in U \cup L$ and $\ell \in [m^*]$, $\widetilde{R}^p_{p_{i,\ell}} = (U \cup L) \setminus \{i\}$. Let $i^* \in L$ be the first agent in L whose all the submissions became completely assigned in the execution of PeerReview-TTC. Consider any $p_{i^*,\ell}$. Note that it is probable that while $p_{i^*,\ell}$ was assigned to some agent i in PeerReview-TTC, it was moved to another agent i' during the execution of Filling-Gaps. But, i' belongs to U and we can conclude that if $p_{i^*,\ell}$ is assigned to some $i \in N \setminus U$ at the output of PRCore, this assignment took place during the execution of PeerReview-TTC. Now, let $Q_1 = R^p_{p_{i^*,\ell}} \setminus (R^p_{p_{i^*,\ell}} \cap \widetilde{R}^p_{p_{i^*,\ell}})$ and $Q_2 = \widetilde{R}^p_{p_{i^*,\ell}} \setminus (R^p_{p_{i^*,\ell}} \cap \widetilde{R}^p_{p_{i^*,\ell}})$. If $Q_1 = \emptyset$, then we have that $R^p_{p_{i^*,\ell}} = \widetilde{R}^p_{p_{i^*,\ell}}$ which means that $R^p_{p_{i^*,\ell}} \succeq_{i^*,\ell} \widetilde{R}^p_{p_{i^*,\ell}}$. If $Q_1 \neq \emptyset$, then $Q_1 \subseteq N \setminus (U \cup L)$ and $Q_2 \subseteq U \cup L$ since $\widetilde{R}^p_{p_{i^*,\ell}} = U \cup L$. Let

$$i' = \underset{i\in Q_1}{\operatorname{argmax}}\, \sigma_{i^*,\ell}(i),$$

i.e. i' is ranked at the lowest position in $\sigma_{i^*,\ell}$ among the agents that review $p_{i^*,\ell}$ under R but not under $\widetilde{R}$ and let

$$i'' = \underset{i\in Q_2}{\operatorname{argmin}}\, \sigma_{i^*,\ell}(i),$$

i.e. i'' is ranked at the highest position in $\sigma_{i^*,\ell}$ among the agents that review $p_{i^*,\ell}$ under $\widetilde{R}$ but not under R. From above, we know that the assignment of $p_{i^*,\ell}$ to i' was implemented during the execution of PeerReview-TTC, since $i' \in N \setminus (U \cup L)$. Hence, with very similar arguments as in the previous case, we will conclude that $\sigma_{i^*,\ell}(i') < \sigma_{i^*,\ell}(i'')$. We have $R(i', p_{i^*,\ell}) = 1$ if and only if i^* has an outgoing edge to i' at some round of PeerReview-TTC. At this round, we know that i'' can review more submissions, since $i'' \in N'$ and if i^* has incompletely assigned submissions, then any $i \in N'$ has incompletely assigned submissions. This means that if $\sigma_{i^*,\ell}(i') > \sigma_{i^*,\ell}(i'')$, then i^* would point i'' instead of i'. Hence, we conclude that $\sigma_{i^*,\ell}(i') < \sigma_{i^*,\ell}(i'')$. Then, from the definition of i' and i'' and from the order separability property we have that $R^p_{p_{i^*,\ell}} \succ_{i^*,\ell} \widetilde{R}^p_{p_{i^*,\ell}}$. Thus, either if Q_1 is empty or not, we have that for any $p_{i^*,\ell} \in P'_i$, it holds that $R^p_{p_{i^*,\ell}} \succeq_{i^*,\ell} \widetilde{R}^p_{p_{i^*,\ell}}$ and from consistency we get that $R \succeq_{i^*} \widetilde{R}$ which is a contradiction. □

4 CORE AND OTHER OBJECTIVES

In the previous section, we show that an assignment in the core is guaranteed to exist under very minor assumptions regarding the preferences of the authors. As we mentioned in the introduction, existing works have focused on different objectives. To be able to compare our objective with the existing ones, from now on, we take the standard approach that for each paper j and each reviewer i, it is given a similarity score $S(i, j)$ which is calculated as a function of different parameters. Given an assignment R, we also assume that the utilities of the papers and the authors are additive, i.e. the utility of a paper j, denoted by u^p_j, is equal to $u^p_j = \sum_{i \in R^p_j} S(i, j)$ and the utility of an author i, denoted by u^a_i, is equal to $u^a_i = \sum_{j \in P_i} u^p_j$.

The most known objective, as it is used at the Toronto Paper Matching System (TPMS) [1] is the maximization of the utilitarian social welfare (USW), which is given by

$$USW(R) = \sum_{i \in N} \sum_{j \in P_i} u^p_j(R).$$

We denote the algorithm which computes such an assignment as TPMS.

A different objective that was introduced by Stelmakh et al. [25] is to maximize the egalitarian social welfare (ESW) which is given by

$$ESW(R) = \min_{j \in \cup_{i \in N} P_i} u^p_j(R).$$

Stelmakh et al. [25] considered the extended leximin version of this objective where subject to maximize the minimum utility of all papers, they aim to maximize the second minimum utility of all papers, and subject to that they aim to maximize the third minimum utility of all papers and so on. The algorithm that achieves this objective is called PeerReview4All (PR4A).

A reasonable question is whether the core is compatible with good approximations of USW and ESW. Below, we show that there are instances where any solution in the core does not achieve an approximation ratio better than $\Omega(n)$ with respect to USW and a finite approximation ratio with respect to ESW.

Theorem 4.1. *For any n, k_p and k_a, where k_a is divisible by k_p, when $n \geq 2 \cdot k_p \cdot ka + 1$, there exists an instance such that no assignment in the core achieves approximation ratio better than $\Omega(n/k_a)$ with respect to USW and a finite approximation with respect to ESW.*

Proof. Suppose that each agent submits k_a/k_p submissions. Let $N_1 = \{1, \ldots, \lfloor n/2 \rfloor\}$ and $N_2 = \{\lfloor n/2 \rfloor + 1, \ldots, n\}$ Consider an instance where the similarity scores are as following:

- For $i, i' \in N_1$, $s(i', p_{i,1}) = 0$
- For $i \in N_1$ and $i' \in N_2$, $s(i', p_{i,1}) = 1$
- For $i, i' \in N_2$, $s(i', p_{i,1}) = \epsilon_1$
- For $i \in N_2$ and $i' \in N_1$, $s(i', p_{i,1}) = \epsilon_2$
- For $i, i' \in N$, $s(i', p_{i,j}) = 0$, for each $j > 1$

where $\epsilon_1 > \epsilon_2$.

Now, suppose that there are at least k_p+1 agents in N_2 whose first submissions are not exclusively assigned to reviewers in N_2. Then, if they deviate and assign their submissions among themselves, this would lead in a valid assignment as each submission would be reviewed by k_p agents and they would strictly improve their utility. Thus, we conclude that an assignment is in the core if the first submission of at most k_p authors in N_2 are not exclusively assigned to authors in N_2. Hence, we get that there are at most $k_p \cdot k_a$ assignments of submissions that belong in authors in N_1 to reviewers in N_2. This means that the maximum utilitarian welfare of an assignment in the core is equal to $k_p \cdot k_a + n \cdot k_a \cdot \epsilon_1$ where as ϵ_1 goes to zero the welfare goes to $k_p \cdot k_a$. Moreover, for $\lfloor n/2 \rfloor > k_p \cdot k_a$, we have that at least one agent in N_1 should have zero utility under any assignment in the core. On the other hand, by assigning the submissions of the agents in N_1 to agents in N_2 and the submissions of the agents in N_2 (except for the submissions of the last agent when n is odd) to agents in N_1, we achieve utility at least equal to $k_p \cdot \lfloor n/2 \rfloor$. Thus, the approximation with respect to the optimal social welfare cannot be better than $\Omega(n/k_a)$. Moreover, by this way the minimum utility is equal to ϵ_2 and for $\epsilon_2 > 0$, and we get that the approximation with respect to ESW is unbounded. □

The condition of the theorem is quite realistic since in practice k_p and k_a are small constants. Thus, we can also get the following Corollary.

Corollary 4.2. *There are instances where no assignment in the core achieves approximation ratio better than $\Omega(n)$ with respect to USW.*

It is known from the literature that we can find an assignment with maximum USW in polynomial times using standard tools [26]. In the previous section, we also presented a polynomial time algorithm that finds an assignment in the core. Here, we surprisingly show that if $NP \neq P$, there is no polynomial time algorithm that finds an assignment in the core with maximum USW. Moreover, it is known that finding an assignment with maximum egalitarian social welfare is a NP-hard problem [2]. Here, we show that it is NP-hard to find an assignment in the core with bounded approximation to the maximum ESW that can be achieved by any assignment in the core.

Theorem 4.3. *Finding an assignment in the core with maximum social welfare is NP-hard. Moreover, finding an assignment in the core with bounded approximation with respect to the maximum egalitarian welfare achieved by assignments in the core is NP-hard.*

Proof. We begin by proving the theorem for the case that $k_a = k_p = 1$, and later we generalize it for any k_a.

We use a polynomial-time reduction from 2P2N-3SAT, the special case of 3SAT where every boolean variable appears twice as positive and twice as negative literal. Let ϕ be an instance of 2P2N-3SAT which consists of n boolean variables, $x_1, x_2, ..., x_n$ and m clauses $C_1, C_2, ..., C_m$ with $n = 3m/4$. We assume that n is divisible by 3. Given ϕ, we construct an instance for the assignment review problem such that:

- If ϕ is satisfiable, then there exists an assignment in the core with social welfare at least $4n/3$ and with minimum paper score $\epsilon > 0$.
- If ϕ is not satisfiable, then any assignment in the core has social welfare less than $4n/3 - 1/2$ and minimum paper score equal to zero.

The assignment review problem is as following: For each boolean variable x_i, we add agents $x_{i,1}$, $x_{i,2}$, $\bar{x}_{i,2}$ and $\bar{x}_{i,1}$, where the first two (respectively, the last two) agents corresponds to the two occurrences of the literal x_i (resp., $\neg x_i$). Moreover, for each clause C_j we add agent c_j. Each agent has exactly on submission and the expertises of the agents over the submissions are as follows: For each bolean variable x_i,

- $S(x_{i,1}, p_{\bar{x}_{i,1}}) = S(\bar{x}_{i,1}, p_{x_i}) = \epsilon_1$
- $S(x_{i,2}, p_{\bar{x}_{i,2}}) = S(\bar{x}_{i,2}, p_{x_{i,2}}) = \epsilon_1$
- $S(x_{i,1}, p_{\bar{x}_{i,2}}) = S(\bar{x}_{i,2}, p_{x_{i,1}}) = \epsilon_2$
- $S(x_{i,2}, p_{\bar{x}_{i,1}}) = S(\bar{x}_{i,1}, p_{x_{i,2}}) = \epsilon_2$

and for each $C_j = (\ell_1 \vee \ell_2 \vee \ell_3)$ where ℓ_1, ℓ_2 and ℓ_3 correspond to the t_1-th appearance of literal x_{i_1} (resp., of the literal $\neg x_{i_1}$), t_2-th appearance of literal x_{i_3} (resp., of the literal $\neg x_{i_3}$) and t_3-th appearance of literal x_{i_3} (resp., of the literal $\neg x_{i_3}$), respectively, with $t_1, t_2, t_3 \in \{1, 2\}$,

- $S(x_{i_1,t_1}, p_{c_j}) = S(x_{i_2,t_2}, p_{c_j}) = S(x_{i_3,t_3}, p_{c_j}) = 1$
- $S(i, p_{c_j})$=0, for any agent $i \notin \{x_{i_1,t_1}, x_{i_2,t_1 2}, x_{i_3,t_3}\}$

and all the remaining scores are equal to ϵ_3 with $1/(8n) \geq \epsilon_1 > \epsilon_2 > \epsilon_3 > 0$.

If ϕ is satisfiable, then we use a truth assignment to find an assignment in the core with social welfare at least $4n/3$ and minimum paper score equal to $\epsilon_2 > 0$ as follows: For every true variable x_i, we assign $p_{x_{i,t}}$ to $\bar{x}_{i,t}$, for each $t \in \{1, 2\}$. Respectively, for every false variable x_i, we assign $p_{\bar{x}_{i,t}}$ to $x_{i,t}$, for each $t \in \{1, 2\}$. For every clause C_j, we arbitrary select one of the true literals of the clause, and assign p_{c_j} to the corresponding agent. All the remaining assignments are arbitrary. First, we see that under this assignment each agent c_j has utility 1, as p_{c_j} is assigned to a reviewer with similarity score equal to 1. Hence, the social welfare is at least $4n/3$. Moreover, the minimum score utility is equal to ϵ_2. Now, we show that it is also in the core. First, no agent c_j has incentives to deviate as her submission is assigned to one of the best possible agents for her submission. Now, consider an agent $x_{i,t}$, for $t \in \{1, 2\}$, when variable x_i is true. From the construction, $p_{x_{i,t}}$ is assigned to $\bar{x}_{i,t}$ which has the highest similarity score for it and hence $x_{i,t}$ does not have any incentives to deviate. With similar, arguments we can show that $\bar{x}_{i,t}$, for $t \in \{1, 2\}$, does not have incentives to deviate when x_i is false. Next, consider an agent $x_{i,t}$, for $t \in \{1, 2\}$ when variable x_i is false. While $\bar{x}_{i,1}$ and $\bar{x}_{i,2}$ are the two agents that have higher similarity score for $p_{x_{i,t}}$, than its current reviewer, we know from above that $\bar{x}_{i,1}$ and $\bar{x}_{i,2}$ do not have any incentives to deviate. Hence, $x_{i,t}$ does not have any incentive to deviate with any other agents when x_i is false, and similarity we can show that $\bar{x}_{i,t}$ does not have any incentive to deviate if x_i is true. Thus, in any case there is no deviating coalition and the assignment is in the core.

Now, we show that if the social welfare is at least $4n/3$, then ϕ is satisfiable. Moreover, if the lower utility of each paper is positive, then ϕ is satisfiable. First, assume for the sake of contradiction that some p_{c_j} is not assigned to an agent that corresponds to one of the literals of C_j. Then, since any other agent has similarity score equal to 0 for p_{c_j}, the sum of utilities of agents c_j for $j \in [m]$ is at most $4n/3 - 1$. But since for any $p_{x_{i,t}}$ and $p_{\bar{x}_{i,t}}$ the similarity score of any agent is less than $1/8n$ and there are $4n$ such submissions, we have that the overall sum of utilities is at most $4n/3 - 1/2$ and we reach a contradiction. Moreover, we see that in this case the minimum utility of a paper is equal to zero. Hence, we conclude that every p_{c_j} is assigned to an agent that corresponds to one of the literals of C_j. Suppose that p_{c_j} is assigned to $x_{i,t}$, where x_i appears as positive literary to C_j, and without loss of generality assume that $t = 1$. Then, we see that $p_{x_{i,1}}$ is assigned to $\bar{x}_{i,1}$, as otherwise $x_{i,1}$ and $\bar{x}_{i,1}$ could deviate to their own coalition. Moreover, we notice that the assignment is in the core if and only if at least one of $x_{i,2}$ and $p_{\bar{x}_{i,2}}$ reviews the other's submission. For the sake of contradiction, assume that the assignment meets this requirement by assigning $p_{\bar{x}_{i,2}}$ to $x_{i,2}$, but not $p_{x_{i,2}}$ to $\bar{x}_{i,2}$. Then, $\bar{x}_{i,1}$ and $x_{i,2}$ as none of them reviews each other's submission, they could deviate to their own coalition. So, it should be the case that $\bar{x}_{i,2}$ reviews $p_{x_{i,2}}$. Thus, we have that if p_{c_j} is assigned to $x_{i,1}$, then $p_{x_{i,1}}$ and $p_{x_{i,2}}$ are assigned to $\bar{x}_{i,1}$ and $\bar{x}_{i,2}$, respectively. Hence, no $p_{c_{j'}}$ can be assigned to $\bar{x}_{i,1}$ or $\bar{x}_{i,2}$. With similar arguments, we can show that if p_{c_j} is assigned to $\bar{x}_{i,t}$, then there is no $p_{c_{j'}}$ that is assigned to $x_{i,1}$ or $x_{i,2}$. Now, we see that by setting the variable x_i to true if some p_{c_j} is reviewed by $x_{i,1}$ or $x_{i,2}$, and to false if some p_{c_j} is reviewed by $\bar{x}_{i,1}$ or $\bar{x}_{i,2}$ (these cannot happen concurrently), we get an assignment that satisfies all clauses of ϕ.

Now, when $k_a > 1$, then for each agent $x_{i,t}$ (resp. $\bar{x}_{i,t}$) in the above construction, we assume that there are $k_a - 1$ other agents, denoted by $x^1_{i,t}, \ldots, x^{k_a}_{i,t}$ such that each of them have similarity score equal to ϵ_4 for each other's submission with $1/(8n) > \epsilon_4 > \epsilon_1$. The remaining scores are set as for the case of $x_{i,t}$. If there are at least two agents among $x^1_{i,t}, \ldots, x^{k_a}_{i,t}$ that their submissions are not reviewed by another agent in $\{x^1_{i,t}, \ldots, x^{k_a}_{i,t}\}$, then they have incentives to deviate. Hence, we see that at most one of them can review a submission that does not belong to some agent in $\{x^1_{i,t}, \ldots, x^{k_a}_{i,t}\}$ and at most one submission of some agent in $\{x^1_{i,t}, \ldots, x^{k_a}_{i,t}\}$ may not reviewed from some agent in the same set. Thus, by interpreting this agent as $x_{i,t}$ and this submission as $p_{x_{i,t}}$, with arguments as above the statement follows. □

5 EXPERIMENTS

In this section, we empirically compare PRCore with TPMS, which is widely used, and PR4A which was used in ICML 2020 [24]. While the latter does not explicitly take into account conflicts between reviewers and submissions, when a reviewer is the author of a

Alg.	USW	ESW	α-Core	Dev Pr
PRCore	0.145 ± 0.01	0.037 ± 0.01	1	0%
TPMS	0.161 ± 0.01	0.055 ± 0.01	1.029 ± 0.037	65%
PR4A	0.156 ± 0.01	0.082 ± 0.01	1.053 ± 0.053	86%

Table 1: Results on ICLR 2018

Alg.	USW	ESW	α-Core	Dev Pr
PRCore	0.985 ± 0.04	0.075 ± 0.04	1	0%
TPMS	1.229 ± 0.04	0.075 ± 0.04	1.984 ± 0.32	100%
PR4A	1.059 ± 0.04	0.075 ± 0.04	1.456 ± 0.02	100%

Table 2: Results on CVPR

Alg.	USW	ESW	α-Core	Dev Pr
PRCore	2.62 ± 0.04	0.075 ± 0.04	1	0%
TPMS	2.986 ± 0.04	0.125 ± 0.04	1.085 ± 0.03	95%
PR4A	2.928 ± 0.04	0.847 ± 0.04	1.182 ± 0.05	100%

Table 3: Results on CVPR 2018

submission, we set the corresponding score to be equal to a large negative number. FairIR and FairFlow algorithms [8] also did not take conflicts into account, but the same trick did not work since the latter does not work with negative scores, while when we implemented and incorporated negative scores in the former, it could not find an optimal solution. For this reason, we do not compare our algorithm with these algorithms.

We use three conference datasets: Conference on Computer Vision and Pattern Recognition (CVPR) and the 2018 iteration of CVPR which both were used by [8], and International Conference on Learning Representations (ICLR) 2018, which was used by [31]. In all the experiments, we set $k_a = k_p = 3$. In ICLR'18, the similarity matrix and the conflict matrix are available where the entry in row i and column j indicates the similarity score and a conflict, respectively, between reviewer i and submission j. As Xu et al. [31], we deploy the conflict matrix as the authorship matrix. We disregard any reviewer that does not author any submissions, but note that the addition of more reviewers can only improve the results of our algorithm since these additional reviewers will have no incentives to deviate. Since in our model each submission is authored by exactly one author, and no author can submit more than $\lfloor k_a/k_p \rfloor = 1$ papers, we found a maximum matching on the conflict matrix, and use this as the authorship matrix for our experiments, as Dhull et al. [3] also did. By this way, 883 out of the 911 papers were matched. In CVPR and CVPR'18, the similarity matrix was available, but not the conflict matrix. In both datasets, there are less reviewers than papers. We constructed an artificial author matrix, by matching a paper to the reviewer that has the highest score for it and is not assigned as an author to any other paper so far. By this way, 1373 out of 2623 papers from CVPR and 2840 out of 5062 papers from CVPR'18 were matched.

To measure the performance of different algorithms with respect to the core, we consider multiplicative approximations. In particular, we say that an assignment is in the α-core, if there is no deviating coalition such that all the authors improve their utility by a multiplicative factor of α. For each experiment, we report USW, ESW and the value of α. Because the calculation of the core approximation requires much time, we subsample 50 papers from each database and report means and standard deviation over 100 runs. Note, that if there is an instance where all the authors with initial score equal to 0, deviate to an assignment where all have strictly positive score, then the value of α would be infinite. To avoid situations where the existence of such instance would explode the mean value of α to infinite, we make sure that the similarity matrices do not contain zero values (something that it is very common in CVPR and CVPR'18), by adding 0.0005 to each cell of a similarity matrix. We also report the probability that a deviating coalition exists. Following Kobren et al. [8] and Stelmakh et al. [25], we run 4 iterations of PR4A (they actually run only one), which ensures that the four minimum scores are maximized.

In Table 1, we see the results of ICLR'18. As expected we see that TPMS achieves the highest USW while PR4A achieves the highest ESW. We see that PRCore achieves a multiplicalte approximation better than 1.2 with respet to USW and better than 2.31 with respect to ESW. Both TMPS and PR4A violate core very often, but it seems that the improvement of the scores is not very significant. In Table 2, we see the results of CVPR. Here, again we see that the mutliplicate approximation of PRCore with respect to USW is around 1.3, but it seems to achieve a much better approximation with respect to ESW. On the other hand, TMPS and PR4A violate core with certainty and in this case the value of α is more than 1.4. Lastly, in Table 3, we see the results of CVPR'18. In this case, we notice that PRCore achieves a better approximation with respect to USW, but worse with respect to ESW. Again, TMPS and PR4A violate core almost with certainty and the value of α is non negligible. Overall, we see that in contrast with the worst case, PRCore seems to achieve good approximations with respect to both USW and ESW in the average case. Moreover, methods that are widely used in practise, violate core very often and significantly incentivize communities to deviate.

6 DISCUSSION

This work introduces a novel notion of group fairness, called core, in a peer review setting which asks that each group is treated in a way that does not have any incentive to deviate and make its own conference. We show that a review assignment in the core always exists when each submission is authored by one agent and each agent serves as a reviewer. While, in the worst case, our algorithm achieves bad approximations with respect to often desirable desideratums, using real data, we show that in the average case, the approximations are quite good. On the other hand, famous reviewing assignment systems fail to satisfy group fairness very often and therefore incentivize communities to deviate from the current peer review process.

There are many directions for future work. First, our algorithm cannot be applied in the case that each submission is authored by multiple authors. The main difference is that when a submission is authored by multiple authors, it is probable that PeerReview-TTC returns U, with $|U| > k_p$. While one can see that PeerReview-TTC returns an assignment in the core even when a submission is authored by many agents, the tweaks that are made in FillingGaps may result in an assignment that is not in the core when $|U| > k_p$. Another limitation of our algorithm is that it assumes that each paper is assigned to the same number of reviewers and each reviewer can review at most the same number of papers. The extension of our algorithm for the multi-author case, and for the cases that papers are assigned to different numbers of reviewers and reviewers review different number of papers is an important open problem. Moreover, strategic issues in conference peer review have been examined in detail in recent years (see the surveys [15, 20]). In Appendix A, we show that our algorithm is not strategyproof. The design of an algorithm that returns an assignment in the core and is also strategyproof is another interesting problem. Lastly, even if a community has incentives to deviate, the cost of deviating, may outweigh the benefits of doing it. A more general model that also includes the cost for a community to break away from a large conference is another interesting direction.

REFERENCES

[1] Laurent Charlin and Richard Zemel. 2013. The Toronto paper matching system: an automated paper-reviewer assignment system. In *Proceedings of the ICML Workshop on Peer Reviewing and Publishing Models*.

[2] Stephen Demko and Theodore P. Hill. 1988. Equitable distribution of indivisible objects. *Mathematical Social Sciences* 16, 2 (1988), 145–158.

[3] Komal Dhull, Steven Jecmen, Pravesh Kothari, and Nihar B. Shah. 2022. Strategyproofing peer assessment via partitioning: The price in terms of evaluators' expertise. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, Vol. 10. 53–63.

[4] Naveen Garg, Telikepalli Kavitha, Amit Kumar, Kurt Mehlhorn, and Julián . Mestre. 2010. Assigning Papers to Referees. *Algorithmica* 58, 1 (2010), 119–136.

[5] Donald Bruce Gillies. 1953. *Some theorems on n-person games.* Princeton University.

[6] Joseph Y. Halpern and David C. Parkes. 2011. Journals for certification, conferences for rapid dissemination. *Commun. ACM* 54, 8 (2011), 36–38.

[7] Bettina-Elisabeth Klaus and Markus Walzl. 2009. Stable many-to-many matchings with contracts. *Journal of Mathematical Economics* 45, 7–8 (2009), 422–434.

[8] Ari Kobren, Barna Saha, and Andrew McCallum. 2019. Paper matching with local fairness constraints. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1247–1257.

[9] Hideo Konishi, Thomas Quint, and Jun Wako. 2001. On the Shapley-Scarf economy: the case of multiple types of indivisible goods. *Journal of Mathematical Economics* 35, 1 (2001), 1–15.

[10] Jing Wu Lian, Nicholas Mattei, Renee Noble, and Toby Walsh. 2018. The Conference Paper Assignment Problem: Using Order Weighted Averages to Assign Indivisible Goods. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, Sheila A. McIlraith and Kilian Q. Weinberger (Eds.). AAAI Press, 1138–1145.

[11] Xiang Liu, Torsten Suel, and Nasir Memon. 2014. A robust model for paper reviewer assignment. In *Proceedings of the 8th ACM Conference on Recommender systems*. 25–32.

[12] David Mimno and Andrew McCallum. 2007. Expertise modeling for matching papers with reviewers. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. 500–509.

[13] Raymond S. Nickerson. 2005. What Authors Want From Journal Reviewers and Editors. (2005).

[14] Regina O'Dell, Mirjam Wattenhofer, and Roger Wattenhofer. 2005. The paper assignment problem. *Technical Report/ETH Zurich, Department of Computer Science* 491 (2005).

[15] Matthew Olckers and Toby Walsh. 2022. Manipulation and Peer Mechanisms: A Survey. *CoRR* abs/2210.01984 (2022). arXiv:2210.01984

[16] Justin Payan and Yair Zick. 2022. I Will Have Order! Optimizing Orders for Fair Reviewer Assignment. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*. 440–446.

[17] Andi Peng, Jessica Zosa Forde, Yonadav Shavit, and Jonathan Frankle. 2022. Strengthening Subcommunities: Towards Sustainable Growth in AI Research. *arXiv preprint arXiv:2204.08377* (2022).

[18] Marko A. Rodriguez and Johan Bollen. 2008. An algorithm to determine peer-reviewers. In *Proceedings of the 17th ACM conference on Information and knowledge management*. 319–328.

[19] Nihar B. Shah. 2019. Principled methods to improve peer review. *Technical Report* (2019).

[20] Nihar B. Shah. 2022. Challenges, experiments, and computational solutions in peer review. *Commun. ACM* 65, 6 (2022), 76–87.

[21] Nihar B. Shah. 2022. Challenges, experiments, and computational solutions in peer review. *Commun. ACM* 65, 6 (2022), 76–87.

[22] Lloyd Shapley and Herbert Scarf. 1974. On cores and indivisibility. *Journal of Mathematical Economics* 1, 1 (1974), 23–37.

[23] Marilda Sotomayor. 1999. Three remarks on the many-to-many stable matching problem. *Mathematical Social Sciences* 38, 1 (1999), 55–70.

[24] Ivan Stelmakh. 2021. Towards Fair, Equitable, and Efficient Peer Review. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 15736–15737.

[25] Ivan Stelmakh, Nihar B. Shah, and Aarti Singh. 2019. PeerReview4All: Fair and accurate reviewer assignment in peer review. In *Algorithmic Learning Theory*. PMLR, 828–856.

[26] Camillo J. Taylor. 2008. On the optimal assignment of conference papers to reviewers. (2008).

[27] Hong Diep Tran, Guillaume Cabanac, and Gilles Hubert. 2017. Expert suggestion for conference program committees. In *2017 11th International Conference on Research Challenges in Information Science (RCIS)*. IEEE, 221–232.

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

[29] George Vrettas and Mark Sanderson. 2015. Conferences versus journals in computer science. *Journal of the Association for Information Science and Technology* 66, 12 (2015), 2674–2684.

[30] Fan Wang, Ben Chen, and Zhaowei Miao. 2008. A survey on reviewer assignment problem. In *International conference on industrial, engineering and other applications of applied intelligent systems*. 718–727.

[31] Yichong Xu, Xiaofei Zhao, Han Shi, and Nihar B. Shah. 2019. On strategyproof conference peer review. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*. 616–622.

A STRATEGYPROOFNESS

THEOREM A.1. *PRCore is not strategyproof.*

PROOF. Consider an instance with $n = 4$ and $k_p = k_a = 1$. Suppose that $\sigma_1 = 2 \succ 3 \succ 4$, $\sigma_2 = 3 \succ 1 \succ 4$ and $\sigma_3 = 1 \succ 2 \succ 4$. We see that if we run PeerReview-TTC, then the partial assignment that is constructed is that 1 is assigned p_3, 2 is assigned p_1 and 3 is assigned p_2. Then, Filling-Gaps should be called as p_4 is not completely assigned and there are no more cycles in the preference graph. Since, the submissions of the first three agents became complete assigned at the same round, L can include any of them. Due to symmetry, without loss of generality, assume that $L = \{1\}$. Then, Filling-Gaps to ensure that the assignment is valid, moves p_1 from 2 to 4 and assigns p_4 to 2. Hence, at the final assignment, p_1 is assigned to 4. If 1 misreports $\sigma_1' = 3 \succ 2 \succ 4$, then when we run PRCore, in PeerReview-TTC, p_1 is assigned to 3 and p_3 is assigned to 1, and then p_2 is assigned to 4 and p_4 is assigned to 2. The algorithm does not call Filling-Gaps as the assignment is already valid. So, under this misreport p_1 is assigned to 3 which 1 strictly prefers than 4. So, 1 has incentives to misreport and we conclude that PRCore is not strategyproof. □