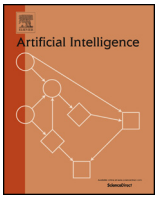
Artificial Intelligence 316 (2023) 103843

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint



# PeerNomination: A novel peer selection algorithm to handle strategic and noisy assessments

Omer Lev [a], Nicholas Mattei [b], Paolo Turrini [c], Stanislav Zhydkov [d,*]

[a] Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Israel
[b] Department of Computer Science, Tulane University, USA
[c] Department of Computer Science, University of Warwick, United Kingdom
[d] Mathematics Institute, University of Warwick, United Kingdom

ARTICLE INFO

*Article history:*
Received 30 November 2021
Received in revised form 12 June 2022
Accepted 21 December 2022
Available online 29 December 2022

*Keywords:*
Peer selection
Strategyproofness
Optimality
Noisy opinions
Reweighting

ABSTRACT

In peer selection a group of agents must choose a subset of themselves, as winners for, e.g., peer-reviewed grants or prizes. We take a Condorcet view of this aggregation problem, assuming that there is an objective ground-truth ordering over the agents. We study agents that have a noisy perception of this ground truth and give assessments that, even when truthful, can be inaccurate. Our goal is to select the best set of agents according to the underlying ground truth by looking at the potentially unreliable assessments of the peers. Besides being potentially unreliable, we also allow agents to be self-interested, attempting to influence the outcome of the decision in their favour. Hence, we are focused on tackling the problem of *impartial (or strategyproof) peer selection* – how do we prevent agents from manipulating their reviews while still selecting the most deserving individuals, all in the presence of noisy evaluations? We propose a novel impartial peer selection algorithm, PeerNomination, that aims to fulfil the above desiderata. We provide a comprehensive theoretical analysis of the recall of PeerNomination and prove various properties, including impartiality and monotonicity. We also provide empirical results based on computer simulations to show its effectiveness compared to the state-of-the-art impartial peer selection algorithms. We then investigate the robustness of PeerNomination to various levels of noise in the reviews. In order to maintain good performance under such conditions, we extend PeerNomination by using *weights* for reviewers which, informally, capture some notion of reliability of the reviewer. We show, theoretically, that the new algorithm preserves strategyproofness and, empirically, that the weights help identify the noisy reviewers and hence to increase selection performance.[1]



* Corresponding author.
*E-mail addresses:* omerlev@bgu.ac.il (O. Lev), nsmattei@tulane.edu (N. Mattei), p.turrini@warwick.ac.uk (P. Turrini), s.zhydkov@warwick.ac.uk (S. Zhydkov).

[1] This paper is a significant extension of our IJCAI 2020 contribution [29], which was limited to the study of PeerNomination without noise. Besides exploring the role of weights in peer selection in the presence of noise, we also extend the unweighted version with novel theoretical and experimental results.

https://doi.org/10.1016/j.artint.2022.103843

## 1. Introduction

Peer evaluation and selection, where agents rate others and then choose a subset of themselves for an award or a prize, is one of the pillars for quality assessment in scientific contexts and beyond. While many of the current methods rely on expert panels, ideally impartial to the selection process [7,42], there is increasing need for alternative mechanisms that keep the procedure both reliable and cheap. An important approach to achieve this goal is that of using the agents that have submitted proposals for review as the set of reviewers themselves. This is particularly relevant in open online courses [37], where hiring professional graders is prohibitively expensive. Indeed, even large AI venues such as IJCAI and NeurIPS have been implementing a portion of this system, requiring authors who submit papers to agree to be the reviewers of other papers.

The importance of improving peer reviewing procedures has been brought to light by the 2014 NeurIPS experiment [24,43]: of all papers submitted to NeurIPS 2014, 10% were reviewed twice by two independent committees which, astonishingly, agreed on less than half of the accepted papers. Whether the outcome was due to bias, incompetence, or rather well-thought disagreement is still unclear. What had been made clear, however, is that the current solutions seem to suffer from undesirable features. The exploding number of papers at AI and general computer science venues has spurred interest in improving many aspects of the peer review process, including: assignment biases [32,25,19], review quality [48], reviewer training [46], and even the quality of reviewers' discussions (see overview by Shah [42]). Other studies of bias in evaluative processes have also brought to the fore the extent and impact of inaccurate assessments in peer reviewing, for example [49,45]. Finding high quality mechanisms for peer review is a critical step in helping the review process in large conferences [4], grant reviewing [33], online courses [47], and other domains.

Researchers in algorithmic game theory and computational social choice worked on the peer selection problem for at least the past decade, focusing on accurate and strategyproof algorithms, including Partition [1], Credible Subset [23] and ExactDollarPartition (EDP) [4]; we provide an overview of these algorithms and more in Section 3. All of these algorithms take a Condorcet view on this aggregation problem, i.e., that there exists an *a-priori* ground-truth ranking of the agents, and we wish to select as many of the top ranked agents as possible, though given only access to the agents' own noisy reports [52]. While this raises obvious philosophical challenges – e.g., what does this ground truth represent if we cannot have direct access to it? – we follow this view as it allows for quantitative analysis of the performance of peer selection algorithms, and hence their objective comparison.

Many of the existing algorithms we survey in Section 3 highlight the trade-offs forced by the pursuit of the dual goal of impartiality and optimality. Some require the set of reviewing agents to be partitioned into clusters that do not review each other [4]; while others sacrifice *exactness* – the ability to select a given number of agents consistently [23]. With PeerNomination, the algorithm presented in this paper, we also sacrifice exactness, but we are able to achieve a new state-of-the-art performance. Additionally, none of the existing algorithms seek to alleviate the problem of noisy inputs in a unified, strategyproof mechanism. When earlier work did engage with noisy reports, it was limited to empirical testing with relatively low noise, e.g., a Mallows model with $\varphi = 0.5$ [4], which yields fairly minor changes in agents' reports (as will be shown in Section 2.3). We are instead concerned with algorithms that can handle a significant level of noise, while maintaining strategyproofness and high quality of selection, an important missing aspect in the literature.

Ideally, we would like an algorithm that is capable of identifying inaccurate reviewers and reducing their influence on the final selection, using only the agents' reports themselves as a guide. We could, for example, try and downgrade those reviewers that differ too much from others. However, there are two problems with this approach: first, the noise may be such that it is difficult to establish what the consensus actually is; and second, that this meta-level reweighting can be exploited strategically. Simple reweighting is not strategyproof: consider, for example, an agent $a$ that is harshly reviewing agent $b$, with both $a$ and $b$ reviewing a third agent $c$. Agent $b$ could benefit by reviewing agent $c$ in a way that would present agent $a$ as an unreliable agent, lowering the impact of the report of agent $a$ for agent $b$ if weights are computed based on correlations to the evaluations of others, e.g., as done by Merrifield and Saari [33]. On the other hand, if a mechanism is able to identify agent $b$ as a source of noise, it can increase the overall quality of the selection. While one can reweight agents without maintaining strategyproofness [47,51], we wish to achieve increased selection quality *and* strategyproofness. The algorithm we present in this paper is able to achieve both of these tasks with state of the art performance.

### 1.1. Contribution

We present PeerNomination, an impartial (or strategyproof) peer selection method for scenarios where $n$ agents review and are each reviewed by $m$ others, with the goal of selecting $k$ of them. Each proposal,[2] which we identify with the proposing agent, is considered independently and it is selected only if it falls in the top $\frac{k}{n}m$ of the majority of its reviewers' (partial) rankings, using a probabilistic completion if such number is not an integer. Performing the selection independently relaxes the exactness requirement, hence our algorithm is not guaranteed to select exactly $k$ agents every time. However, under some mild assumptions, the algorithm does select exactly $k$ agents in expectation. Unlike other well-known

[2] For the sake of clarity, when an agent is referred to as a *reviewer* we will always mean in the context of reviewing others and we will use the word *proposal* when referring to an agent that is being reviewed.

peer reviewing methods, e.g., ExactDollarPartition (EDP), PeerNomination does not rely on clustering nor on reviewers submitting complete rankings, allowing more flexibility in where and when it may be deployed.

We compare the performance of PeerNomination against an underlying ground truth ranking, when agent rankings are drawn according to a Mallows Model [28,52], deriving its expected recall analytically.[3] Furthermore, we extend PeerNomination to make use of *reviewer weights* in order to handle noisy and inaccurate agents. To do so, we explicitly formulate (reliability) weights on reviewers in a way that does not violate strategyproofness, and use this information to reweight their scores. PeerNomination with weights is able to handle high levels of noise, even when reviewers act adversarially. We show analytically that weighting schemes can improve the overall quality of the selection significantly.

Finally, we empirically compare our method against other peer selection mechanisms, for which analytic performance bounds are unknown, using a number of well-known classification measures. Our results show that PeerNomination improves on the current best performance in terms of recall known from the literature and relies on milder assumptions on the underlying reviewer graph. This suggests that relaxing the exactness requirement in peer selection outcomes can give us an improved quality of the accepted set. Moreover, we show empirically that PeerNomination with weights is able to significantly improve the quality of peer selection of PeerNomination without weights, under a variety of noise parameters.

*Paper outline.* In Section 2, we provide formal definitions of the problem and all the concepts required to describe the algorithm precisely, as well as a description of the noise model used for empirical testing. In Section 3 we detail a number of previously proposed algorithms for peer selection and what sets our method apart. Section 4 introduces PeerNomination in detail together with its different weighting schemes and an assignment procedure. We then use Section 5 to derive analytic results about PeerNomination such as strategyproofness and expected recall. Finally, in Section 6, we put PeerNomination up to the test against a state-of-the-art strategyproof peer selection algorithm, EDP [4], to measure its performance in a realistic setting.

## 2. Preliminaries

*Agents and ground truth.* In the peer selection problem, agents are represented by the set of positive integers $\mathcal{N} = \{1, 2, ..., n\}$. As is common in the peer reviewing literature and consistent with a Condorcet theory of voting [52], we assume that there is a *ground truth* that all agents share, which we define as a linear order over $\mathcal{N}$. In other words, we make the simplifying assumption that, if agents were to assess each other accurately, they would report the same ranking.[4] To provide a more general and realistic setup, we use a noise model that gives each agent a distorted view of this ground truth. Assuming noisy reviewers requires a more nuanced notion of truthfulness, i.e., each agent being true to their own potentially faulty perception.

In general, the peer review process consists of three steps: (1) the assignment of proposals to be reviewed by each agent; (2) the submission of reviews by the reviewing agents; (3) the aggregation of the submitted reviews. We formalise each of these steps next, focusing on the notions needed to study our PeerNomination algorithm. Adopting the academic peer review terminology, we will refer to agents as *reviewers* in the context of them giving reviews and as *proposals* in the context of being reviewed.

*Review assignment.* In peer reviewing, agents are assigned to review each others' work. A desirable stipulation for the review assignment is that no agent should review themselves. We also typically expect each agent to review a similar number of proposals and all proposals to receive a similar number of reviews.

Formally, a *review assignment* is a function $A : \mathcal{N} \to 2^{\mathcal{N}}$ such that for any $i \in \mathcal{N}$, $i \notin A(i)$. This gives each reviewer $i$ a set of proposals to evaluate, $A(i)$, which we call a reviewer's *bundle*. For our peer selection procedure we often need to refer to the set of agents assigned to review a particular proposal $j$. Slightly abusing notation, we denote all reviewers of proposal $j$ by $A^{-1}(j) = \{i \in \mathcal{N} \mid j \in A(i)\}$; we call this a proposal's *panel*. Given an integer $m$, an assignment is called *m-regular* if for any $i \in \mathcal{N}$, $|A(i)| = |A^{-1}(i)| = m$.

In practice, $m$ tends to be small and constant with respect to $n$, representing the assumption that each reviewer has limited reviewing capacity. This makes $m$-regular assignments desirable, as they distribute the workload evenly. For this reason and to simplify theoretical analysis, we will assume all our assignments are $m$-regular for the rest of the paper. Note that this assumption is not needed for our algorithm to work.

In some settings, such as conference peer review, we may wish to view the assignment in light of agent bids [25], similarity scores [12,32], or some other measure of assignment quality as done by Xu et al. [53]. In our work, much like is done at the US National Science Foundation [33] or would be done in a peer review classroom setting [16], we assume

[3] As we explain in detail in Section 6.1, in this paper we use the binary classification definition of recall to measure performance. Recall is calculated as the proportion of all positives selected by the algorithm. This is the same measure as used in Aziz et al. [4] and other peer selection literature, where it is usually referred to as *accuracy*. We decide to use the term recall to avoid confusion since accuracy has a distinct definition as a classification measure.

[4] This assumption is often too simplistic and does not account for the diversity of views which are common in scientific debates. It should be thought of as an idealisation of those scientific communities where methodological debates or epistemic views are not at the heart of the discussion and where participants agree on the objective value of a proposal, subject to its careful examination. From the technical point view this is a standard assumption in Condorcet views of voting [52] and allows for easy theoretical and empirical analysis of the performance of peer selection algorithms.

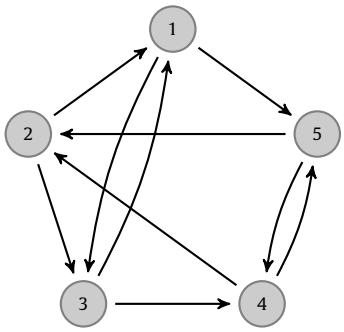



**Fig. 1.** A peer review assignment. The ground truth ranking over the agents is given by the agent number with agent 1 being the best and agent 5 being the worst, i.e., $1 \succ 2 \succ 3 \succ 4 \succ 5$.

that the assignment is independent of the ground truth ranking. An interesting direction for future work would incorporate PeerNomination into a larger framework with more assumptions over the information available to the assignment algorithm. However, in this paper we do not make such assumptions.

*Agents' reviews.* We assume that the reviewers' evaluations are represented by a rank ordering over their bundle. Given an $m$-regular review assignment $A$, agent $i$ reports a *ranking* as a bijection $\sigma_i : A(i) \to \{1, ..., m\}$. A ranking is called *truthful* if it is consistent with the (perceived) ground truth. In a setting without noise, a truthful ranking is therefore a ranking that is consistent with the ground truth, which is accessed by everyone. In a setting with noise, more generally, a truthful ranking is a ranking that is consistent with the agent's individual perception of it given by the noise model. The collection of all rankings is called a *profile* is denoted by $\sigma = (\sigma_1, ..., \sigma_n)$. A *truthful profile* is then a profile of truthful rankings. The set of all possible profiles, truthful or not, is denoted by $\Sigma$.

Most peer selection algorithms use the rankings provided by the reviewers directly, e.g., ExactDollarPartition. However, coming up with full rankings typically poses a higher cognitive load on reviewers and increases the chance of inaccurate assessments. Additionally, in some settings, such as in student peer evaluation of assignments as discussed by De Alfaro and Shavlovsky [16], agents may outright refuse to provide partial or complete rankings, preferring instead to indicate that work is acceptable or not only.

In contrast, our algorithm takes inspiration from approval voting, and does not require the full ranking to be submitted, but rather a set of *approved* agents. As common with approval voting [9], voters simply give a "yes" or a "no" to each candidate, potentially subject to a quota. We call these approvals *nominations*. We expand on this idea by allowing non-integer quotas with the non-integer part representing *partial nomination*.

Given an $m$-regular review assignment $A$ and a quota $q > 0$, a nomination vector from agent $i$ is a function $\overline{\sigma}_i : A(i) \to \{0, 1, q - \lfloor q \rfloor\}$ such that $|\overline{\sigma}_i^{-1}(1)| = \lfloor q \rfloor$ and $|\overline{\sigma}_i^{-1}(q - \lfloor q \rfloor)| = 1$. In other words, reviewers simply submit enough nominations to fill up their quota and choose one additional nomination to be partial. Under our proposed algorithm, this partial nomination will be resolved probabilistically.

**Example 1.** The directed graph in Fig. 1 represents a 2-regular assignment on $n = 5$ agents with no noise, where the ground truth ranking over the agents is: $1 \succ 2 \succ 3 \succ 4 \succ 5$. For example, agent 1 is reviewing agents 3 and 5, their review bundle, and their ranking is consistent with the ground truth that 3 is above 5. If the nomination quota was 1, the truthful strategy of agent 1 would nominate 3 and leave 5 out. If, instead, the nomination quota was increased to 1.2, agent 1 will have to employ partial nominations. Their truthful strategy is then to nominate agent 3 fully and agent 5 partially.

Example 1 shows the use of partial nominations, which will constitute a backbone of our main algorithm. Under PeerNomination, a partial nomination quota would directly translate to a probability of being nominated.

*Aggregation.* The final step of the process is to aggregate the rankings and select a set of winners. Typically, there is a specified number of winners, $k$. However, some mechanisms forgo this requirement and return either a possibly smaller set of winners [3] or possibly no winners [23]. Formally, given a review assignment $A$, a profile $\sigma \in \Sigma$ and an integer $k \leq n$, a *peer selection mechanism* is a function $f : \Sigma \to 2^{\mathcal{N}}$. The mechanism is called *exact* if for every $\sigma \in \Sigma$, $|f(\sigma)| = k$.

**Example 2.** Let us now go back to Example 1 and focus on a nomination quota of 1. If every agent submits a truthful ranking, then the proposals would receive $2, 2, 1, 0, 0$ nominations, respectively. Assume that the target number of winners is $k = 2$. If all agents were truthful, we would select agent 1 and 2. However, note that agent 3 can untruthfully nominate agent 4 instead of agent 1, giving the following nomination distribution: $1, 2, 1, 1, 0$; which makes agent 3 tie for second place with agents 1 and 4. If we were to select uniformly at random, agent 3 would increase their chance of selection from 0 to $\frac{1}{3}$ by manipulating their review, thus keeping $k = 2$ agents as the winners but violating strategyproofness (as defined in Section 2.2).

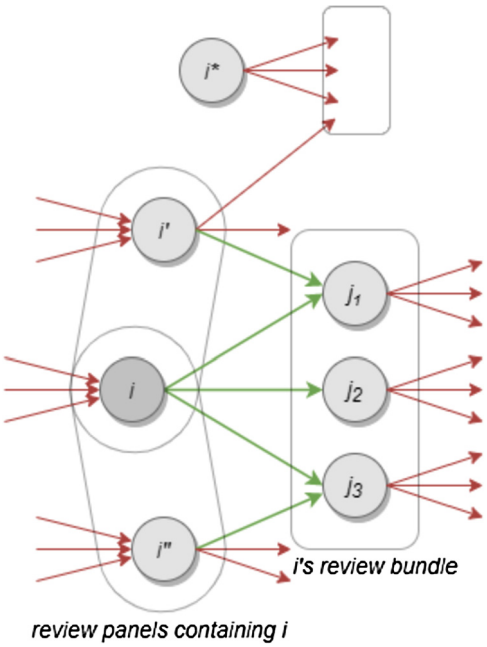



**Fig. 2.** When using first-order weights, the weight of reviewer $i$ is only affected by all rankings received by their bundle, $j_1, j_2, j_3$ (green arrows), but not by any other rankings in the system (red arrows). Hence, reviewers that are part of the same panels as $i$, such as $i'$ and $i''$, can affect their weight. On the contrary, reviewers outside of $i$'s panels, such as $i^*$, bear no effect on $w_i$. (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

An alternative approach would be to select all agents that have, say, one nomination. This way agent 3 is accepted regardless of who they nominate (if we required 2 nominations, agent 3 would not be selected independent of their review). This approach is strategyproof, however it comes at the cost of exactness – we select three agents instead of two.

Example 2 shows that aggregation protocols may not satisfy all desirable properties, in particular strategyproofness and exactness. In this paper we develop an algorithm that relaxes exactness to achieve strategyproofness. However, we also show that, under mild assumptions, exactness is guaranteed in expectation. We shall see that, in noisier scenarios, the employment of weighting schemes will be helpful to discriminate inaccurate reviewers and empirically achieve the higher recall.

### 2.1. Weighting schemes

Peer selection often takes place in the presence of noise of various types, e.g., reviewer bias and/or submitting short or hastily prepared reviews [42], but also in the presence of legitimately different evaluations of the quality of proposals. Hence, we desire algorithms that are able to work in situations of high noise, where agents may have an inaccurate assessment of the other agents but the algorithms are not so strict as to stifle dissenting viewpoints. In order to still provide a reasonable outcome, we will employ weighting schemes to mitigate as much noise as possible. These schemes assign weights to every agent based on the reviews they give to reflect how much (or little) they agree with other reviewers.

A *weighting scheme* is a function $w\colon \Sigma \to [0,1]^n$. When the review profile $\sigma$ is obvious from the context, we denote the reviewer weight of agent $i$ by $w_i$. A weighting scheme is *first-order* if for every $i$, $w_i$ depends only on the comparative rankings of the panels of which $i$ is a part. More formally, $w_i$ only depends on the rankings $\bigcup_{j \in A(i)} \{\sigma_k \mid j \in A(k)\}$. Thus, reviewer $i$ can only influence the weights of their *co-panelists* – those reviewers that are on the same panel as $i$ – as illustrated in Fig. 2. We also require our weighting schemes to be deterministic to preserve a form of *neutrality*: if the input to the weighting scheme is the same for two reviewers, their weights should be the same.

Notice how Google PageRank [36,22], for example, uses weights that are not first-order, as the influence of a ranking is propagated through the system indefinitely. This would be detrimental to our algorithm since a change in a single ranking, e.g., an attempt at strategising, would propagate through most, if not all, reviewer weights, thereby making it very hard to ensure strategyproofness. For this reason, we use only first-order weighting schemes in PeerNomination.

### 2.2. Properties

Since each agent wants their own proposal to be selected, the incentives of self-interested agents in the peer selection problem do not always align with the socially optimal outcome – the selection of the best $k$ agents according to the ground

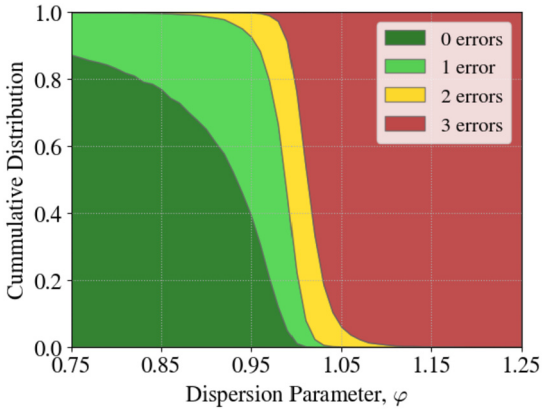



**Fig. 3.** Typical number of errors, i.e., nominations of proposals not actually in the top $\frac{k}{n}m$ of the ground truth ordering, committed by a reviewer who has to nominate 3 out of 9 proposals as a function of the dispersion parameter.

truth. A peer selection algorithm $f$ is called *strategyproof* or *impartial* if for every agent $i$, whenever $i \in f(\sigma)$ for some profile $\sigma$, then also $i \in f((\sigma_1, ..., \sigma_i^*, ..., \sigma_n))$ where $\sigma_i^*$ is truthful. In other words, deviating from truthful reporting can never be beneficial for any agent. In the probabilistic context, we require agents not being able to increase their probability of being selected.

In addition to strategyproofness, we will show that our mechanism, like most, maintains the properties of *anonymity*, i.e., permuting agents makes no difference; *non-imposition*, i.e., any set of $k$ accepted papers is a possible output; and *monotonicity*, i.e., receiving better scores does not decrease the probability of selection.

### 2.3. Noise model

To model the inaccuracies in reviewers' assessments, we assume that each agent is associated with a noisy observation of the ground truth according to a Mallows model [28]. Mallows models have been widely used to compare the performance of peer selection algorithms empirically [29,4], but so far only studied for very mild levels of noise which do not significantly affect the reported rankings.

The Mallows model is parameterised by a dispersion parameter $\varphi \in [0, 1]$ and a reference linear ranking $R$. Given $R$ and $\varphi$, the model induces a probability distribution over all permutations of $R$ such that the probability of the linear order $R'$ is $\pi_{R,\phi}(R') \propto \varphi^{KT(R,R')}$, where $KT(R, R')$ is the Kendall-$\tau$ distance between $R$ and $R'$. The Kendall-$\tau$ distance counts the number of pairwise disagreements between two rankings [21]. Hence, the probability of an agent reporting an additional pairwise disagreement from the reference ranking decreases exponentially. Note that, as we vary the dispersion parameter $\varphi$ from 0 to 1, the probability distribution over all linear rankings moves from being concentrated at $R$ to being uniform over all possible rankings. In our simulations (Section 6), we take the ground truth as the reference ranking and sample a noisy ranking for each agent using the $\varphi$ specified. An important feature of the Mallows model is that it can be sampled efficiently [27,52,30], which allows us to generate a unique reviewer profile for each experiment.

In addition, we test our weighting schemes in settings where some reviewers are not just random, but are actively contrarian to the ground truth. Since the Mallows model only produces random rankings in the worst case (at $\varphi = 1$), we introduce a simple extension in which agents may tend against the ground truth. Formally, given a reference ranking $R$ and a dispersion parameter $\varphi \in (1, 2]$, the *extended Mallows model* samples a ranking $R'$ with probability $\pi_{R^{-1},(2-\phi)}(R') \propto (2-\varphi)^{KT(R^{-1},R')}$, where $R^{-1}$ is the reverse of the linear order $R$.

For example, if we set $\varphi = 1.2$, we assume the agent has the reverse ground truth as the reference ranking and samples a ranking using Mallows model with $\varphi = 0.8$. Thus, the distribution moves smoothly from being concentrated at the ground truth towards the reverse ground truth, while still being uniform around 1. It is worth noting that the Mallows model behaves non-linearly with respect to the number of errors committed by the reviewer. In our setup, the number of errors, i.e., the proposals nominated that fall outside of the top $\frac{k}{m}n$ of the ground truth, is illustrated by Fig. 3. Unless the dispersion parameter is close to 1, reviewers commit very few errors on average. Moreover, a significant probability to get all 3 nominations wrong only arises when we consider $\varphi > 1$ following our contrarian extension.

## 3. Related work

Using the evaluations of peers to rank and select winners is a problem of broad interest beyond CS and AI, including numerous practical domains, e.g., conference, journal, and grant reviewing; large scale course grading, and group decision making. Brought to the fore by Merrifield and Saari [33] to allocate telescope time for the US National Science Foundation, the problem is deeply rooted in economics, with extensive work on the continuous case (see de Clippel et al. [14]), in which

agents allocate fractions of rewards (discrete variants of which – Dollar Raffle and Dollar Partition were suggested by Aziz et al. [4]). In the discrete case there has been interest in the AI community from the work of Alon et al. [1] on partition and onwards. Later, the Credible Subset method [23], where the mechanism examines the possibility of manipulations and accounts for it, was suggested. Despite strategyproofness, the system in inexact (can return no selection), and it was shown that this can happen in a significant number of cases [3]. A prominent recent algorithm is ExactDollarPartition (EDP) [4], which provides exactness at the cost of some randomness, while remaining strategyproof, and improving on the main earlier algorithms.

A study by Caragiannis et al. [11] provides optimal (non-impartial) algorithms for ordinal peer ranking in a setting that is close to the one considered in this paper. In particular, they show that the simple Borda mechanism is optimal in the setting with no noise; they also provide a way to construct an optimal algorithm for a specific Mallows-like noise model. This provides a good benchmark for testing impartial peer selection algorithms. For example, the results from Mattei et al. [29] show that unweighted PeerNomination approaches the recall of Borda in some settings.

Similar algorithms from the multi-agent systems communities include voting rules to aggregate ranks, e.g., $k$-Partite [20], the Committee Rule [20], and Divide-and-Rank [53] algorithms. Others focus on proving bounds on the quality of a given rank aggregation scheme under noisy and partial observations [10]. Yet other methods are approval-based but focus on single agent selection: Permutation [17] and Slicing [8].

A key application area for peer evaluation mechanisms is education, where the problems of reviewer reliability and bias have been extensively studied [38]. We are motivated by evidence from fielded peer evaluation mechanisms showing that students are often unwilling to strictly rank assignments [16] and would rather rely on scores or pass/fail marks (approvals). Within the conference and journal reviewing ecosystem there is also growing interest in detecting strategic behaviour on the part of the reviewers [45,32] as well as de-biasing and calibrating differences in the scores of reviewers [48,25]. We go beyond calibration and de-biasing, identifying suboptimal behaviour in agents' populations and looking at the effect of rescaling on the system as a whole.

Outside peer selection, there is extensive work in the machine learning, information retrieval, and preference learning communities on the *learning to rank* problem: inferring the most likely ranking from possibly noisy observations [26]. These works include learning noise models, e.g., the parameters of a Mallows model, for use in inferring latent preferences of agents [26,52]. This is of great practical interest in information retrieval, where one wishes to rank, e.g., web-pages based on user clicks [41] and in combining labelling from multiple sources for the construction of datasets [51]. However, all of these systems do not concern themselves with strategyproofness, a key focus of our study.

The notion of weights is used elsewhere in computer systems applications, for example in the field of recommender systems, where "reviewers", i.e., the customers, might have an incentive to submit untruthful ratings [39]; a similar approach is also taken in reputation system [40] and other platforms such as Google Search in the form of the PageRank algorithm [36,22].

## 4. PeerNomination

In this section we formally present PeerNomination, including the design and implementation of our reweighting mechanism. We then discuss the trade-offs that arise in the mechanism due to the introduction of these weights. Finally, we present specific examples of some weighting schemes that may be used in PeerNomination. The complete PeerNomination algorithm is given as Algorithm 1.

### 4.1. The PeerNomination algorithm

A usual requirement for a peer selection mechanisms is that it must return a set exactly of size $k$ [4,1,20]. Some approaches investigated relaxing this assumption [3,23], most notably, Bjelde et al. [6] show that this relaxation can lead to better approximation of the optimal selection of winners. We use the intuition that relaxing the exactness requirement can improve recall in PeerNomination, which returns a winning set of size approximately $k$ *in expectation*.

PeerNomination works as follows: suppose every agent reviews and is reviewed by $m$ other agents. If an agent is in the true top $k$, i.e., the *a-priori* ground truth, of the overall $n$ agents, we expect them to be ranked in the top $k$ proportion, i.e., top $\frac{k}{n}m$, of their review bundle by the majority of agents that review the proposal, if the reviewing agents were to report their rankings perfectly. We say that an agent is *nominated* by a reviewer if they are in the top $\frac{k}{n}$ proportion of the reviewer's declared ranking, i.e., their review bundle. We hence refer to $\frac{k}{n}m$ as the *nomination quota*.

As $\frac{k}{n}m$ is unlikely to be an integer, we consider a proposal *nominated for certain* if it is among the top $\lfloor\frac{k}{n}m\rfloor$ proposals in a particular review bundle, where $\lfloor x\rfloor$ denotes the whole part of a positive real number $x$. If a proposal is in the next position, i.e., $\lfloor\frac{k}{n}m\rfloor+1$, we consider the proposal nominated with probability $\frac{k}{n}m-\lfloor\frac{k}{n}m\rfloor$, that is, the decimal part of the nomination quota. For an illustration, see Fig. 4.

We now use the reviewers' nominations induced by their rankings to select the winners. As discussed above, we first use a weighting scheme to compute *reviewer weights* for each of the reviewers, with the aim to detect inaccurate reviewers and assign them a lower weight. We do this by measuring how much the reviewer disagrees with their co-panellists, i.e., how consistent a particular reviewer's ranking is with the agents reviewing the same proposal. We then compare them to

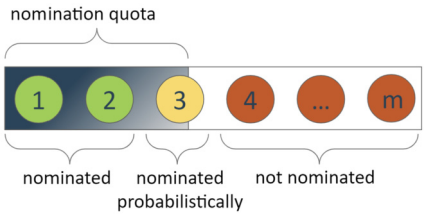



**Fig. 4.** Each reviewer nominates their quota of agents.

**Algorithm 1** PeerNomination.

**Input:** Assignment $A$, review profile $\sigma$, target quota $k$, slack parameter $\varepsilon$, reviewer weights $\{w_1, ..., w_n\}$
**Output:** Accepting set $S$

Set $nomQuota := \frac{k}{n}m + \varepsilon$
**for all** $j$ in $\mathcal{N}$ **do**
  Initialise $nomCount := 0$
  **for all** $i \in A^{-1}(j)$ **do** ▷ Count how many nominations $j$ received
    **if** $\sigma_i(j) \leq \lfloor nomQuota \rfloor$ **then**
      increment $nomCount$ by $w_i$
    **else if** $\sigma_i(j) = \lfloor nomQuota \rfloor + 1$ **then**
      increment $nomCount$ by $w_i$ with probability $nomQuota - \lfloor nomQuota \rfloor$
    **end if**
  **end for**
  **if** $nomCount \geq (\sum_{i \in A^{-1}(j)} w_i)/2$ **then**
    $S \leftarrow j$ ▷ Select $j$ if they have a weighted majority
  **end if**
**end for**
**return** $S$


**Fig. 5.** A non-strategyproof assignment (Left) and a strategyproof one (Right). Algorithm 2 only returns the latter type.

the *Unit* weighing scheme in which $w_i = 1$ for all agents, i.e., all agents have identical weights. The weighting schemes are discussed in detail in Section 4.3.

The final stage of PeerNomination consists in selecting a proposal as a winner if it achieves a weighted majority of nominations. In the Unit case, this translates to a majority of reviewers nominating it. A crucial observation is that, since each proposal is considered independently for selection, the algorithm is not guaranteed to return exactly $k$ agents. However, we will show that the algorithm will select a set of size approximately $k$ if the reviewers submit reviews that are close enough to the ground truth. We will show, moreover, that with PeerNomination truth-telling is an equilibrium outcome, i.e., PeerNomination is strategyproof.

The full PeerNomination algorithm is presented in Algorithm 1. Note that in the algorithm we introduce the *slack parameter*, $\varepsilon$, as part of the input, which extends the nomination quota and serves to fine-tune the algorithm performance. As we will discuss in Section 5.2, this is necessary in some settings to achieve the expected size of $k$ of the winning set.

### 4.2. Review assignment

One of our goals will be to show that PeerNomination is strategyproof (or impartial), which we do in Theorem 1. However, while this is fairly straightforward for the non-weighted (Unit) case, the introduction of weighting schemes requires some care, as illustrated by the following example.

**Example 3.** Consider the example in Fig. 5 (Left), where agents $a, b, c$ are reviewing each other, with the arrows directed from the reviewer to the reviewed proposal. Note how agent $a$ and $b$ are both reviewing agent $c$. In this scenario, agent $b$ may impact the weight given to agent $a$ by manipulating their evaluation of agent $c$. For example, if agent $b$ expects agent $a$ to nominate $c$, they can choose not to nominate $c$, "discrediting" $a$. Hence, $b$ may influence agent $a$'s role in determining if agent $b$ themself is selected.

**Algorithm 2** EULER-BASED ASSIGNMENT. For simplicity, we present the algorithm for an even $n$. This is done without loss of generality. If $n$ is odd, we can add a dummy agent and run the algorithm on $n+1$ agents and $m+1$ assignments. We can then remove the dummy agent, leaving each degree between $m$ and $m+1$.

**Input:** Set of $n$ agents, review number $m \leq n/4$
**Output:** Anti-transitive $m$-regular assignment $A$
Initialise $G = (V, E)$ with $V := [n]$, $E := \emptyset$
Partition $V$ into $X$ and $Y$ such that $|X| = |Y| = n/2$
**for all** $x$ in $X$ **do** ▷ Make a $2m$-regular bipartite graph G(V,E)
  **for all** $i$ in $1, ..., 2m$ **do**
    $y^* \leftarrow \arg\min_y \{\deg(y) \mid y \in Y\}$
    $E \leftarrow E \cup \{\{x, y^*\}\}$
  **end for**
**end for**
Set eulerCycle := *HierholzersAlgorithm*(G) ▷ Use Hierholzer's alg. to find an Euler cycle
Set $A := ([n], E_A)$, $E_A := \emptyset$
**for all** $(v_i, v_{i+1})$ in eulerCycle **do**
  $E_A \leftarrow E_A \cup \{(v_i, v_{i+1})\}$ ▷ Orient the edges in the direction of the Euler
**end for**
**return** A

The example shows that there are cases where the mechanism is not strategyproof. We will show below that some review assignments can avoid this. We henceforth refer to a review assignment that avoids the construction in Fig. 5 *anti-transitive*. Formally, a review assignment is anti-transitive if for any agents $a, b, c$, if $a$ reviews $b$ and $b$ reviews $c$, then $a$ does *not* review $c$.

We now present an algorithm for generating random anti-transitive review assignments and show in Section 5.1 that this algorithm is correct (Proposition 1) and that anti-transitivity makes PEERNOMINATION strategyproof (Theorem 1).

Algorithm 2 works as follows: it first randomly partitions all agents into 2 equally sized sets and creates a $2m$-regular bipartite graph based on this partition. It then orients the edges by traversing an Euler tour through the graph (Fig. 5 (Right)), which yields an $m$-regular directed graph. Notice that Algorithm 2 and Algorithm 1 do not depend on one another and can be studied in isolation.

### 4.3. Weighting schemes

In this section we present three weighting schemes, in addition to the Unit weighting scheme, that can be used to evaluate the reliability of the reviewers. Each of these weighting schemes satisfies the *first order* requirement described in Section 2.1. Informally, since each of these weighting schemes only propagates information through one link of the review graph, they each will ensure that PEERNOMINATION remains strategyproof under the assignment generated using Algorithm 2. The first two contain an "aggressiveness" parameter, which allows us to fine-tune by how much we wish to lower the weights of reviewers the scheme identifies as problematic.

While there are other weighting schemes we may consider, such as ones based on expectation maximisation (EM) algorithms including GLAD [50], Dawid-Skene [15], and even PageRank [36,22], all of these methods do not satisfy the first-order requirement and would render PEERNOMINATION not strategyproof when using those weighting schemes. For this reason, we do not consider these methods in our paper as our focus is on strategyproof methodologies, though exploring the loss of recall for strategyproof versus non-strategyproof methods is an interesting direction for future work.

**Distance** Distance weights are the distance between an agent's review and those of other reviewers. This distance is calculated by averaging the individual differences between the reviewers – when one nominated a proposal and another did not. Formally, the average distance of reviewer $i$ to other reviewers is $d_i = \frac{1}{m^2} \sum_{j \in A(i)} \sum_{l \in A^{-1}(j)} |\overline{\sigma}(j) - \overline{\sigma}_l(j)|$. Then the distance weight is $w_i^{\text{dist}} = (1 - d_i)^\gamma$, where $\gamma$ is the aggressiveness parameter that exaggerates the weights for better discrimination between the agents. This can be understood as an equivalent of Hamming distance in our framework, which is computed between the nominations given by a reviewer and those given by their co-panelists.

**Majority Errors** Let a nomination of a proposal be an "error" by a reviewer if it is a minority opinion, i.e., if the reviewer nominates a proposal which does not have a majority in their panel or does not nominate a proposal with a majority, rounding partial nominations to the closest integer. More formally, let

$$\text{maj}_j^\sigma = \begin{cases} 1, & \text{if } \sum_{i \in A^{-1}(j)} \overline{\sigma}(j) \geq m/2 \\ 0, & \text{otherwise} \end{cases}$$

be the majority for agent $j$ in profile $\sigma$. Then define the number of errors of reviewer $i$ to be $\text{err}_i^\sigma = \sum_{j \in A(i)} \mathbb{1}_{\overline{\sigma}(j) \neq \text{maj}_j^\sigma}$. The weight is defined as $w_i^{\text{majerr}} = 1 - \delta(\text{err}_i^\sigma / m)$, where $\delta$ is the aggressiveness parameter.

**Step** Step applies a step function to the error rate $\text{err}_i^\sigma/m$ defined in the Majority scheme, above. We choose two thresholds, $t_1$ and $t_2$ such that if the error rate reaches $t_1$, we reduce the weight of the reviewer to 0.5; if the error rate reaches $t_2$, we reduce the weight to 0. Additionally, we scale each threshold by the nomination quota as it plays a bigger role in error detection than just the size of the review bundle, $m$. Formally,

$$w_i = \begin{cases} 1, & \text{if } \text{err}_i^\sigma / \frac{n}{k} < t_1 \\ 0.5, & \text{if } t_1 \leq \text{err}_i^\sigma / \frac{n}{k} < t_2 \\ 0, & \text{otherwise.} \end{cases}$$

**Unit** We refer to the version of PeerNomination where weights are ignored (i.e., each $w_i = 1$) as *Unit* PeerNomination.

## 5. Theoretical analysis

In this section we provide the theoretical analysis of PeerNomination. We first prove that PeerNomination satisfies important axiomatic properties, notably strategyproofness. We then derive analytic expressions for the expected recall and output size of PeerNomination in the case of Unit weights. Lastly, we provide a motivation for the effectiveness of weighting schemes by introducing a simplified model of the peer reviewing setting and showing that detecting inaccurate reviewers can improve the recall of peer selection.

### 5.1. Axiomatic properties

Assigning weights to reviewers based on their reviews introduces further complexity and potential for manipulation, as illustrated by Example 3. Thankfully, since we only consider first-order weighting schemes, we only need to introduce a simple condition on the review assignment to maintain strategyproofness of PeerNomination. We first show that this condition is indeed sufficient and then prove that Algorithm 2 guarantees it. Of course, any other assignment-generating algorithm that guarantees this condition will also guarantee strategyproofness of PeerNomination.

**Theorem 1.** PeerNomination *is strategyproof if the review assignment is anti-transitive and the weighing scheme is first-order.*

**Proof.** First, we consider the Unit case, in which each weight is set to 1, independently of the reviews. Under PeerNomination, whether an agent is selected depends solely on the reviews they receive from their reviewers; there is no interaction with the reviews of others. Since no agent reviews themself, they thus cannot affect the chances of their selection.

Now consider the general case, where reviewers are assigned weights based on their nominations. The weights introduce an interaction between the agents, and to guarantee strategyproofness we wish to show that one cannot affect the weight of the reviewers of their own proposal, thus improving their chances of selection. Assume that the review assignment is anti-transitive and the weighing scheme is first-order. Say agent $i$ has a proposal. Since the weighting scheme is first-order, agent $i$ can only influence the weights of agent $i$'s co-panelists and no others. That is, it can only influence the weights of an agent $j$ for which there is a proposal $\ell$ such that both $i$ and $j$ are both reviewing $\ell$. But because assignment is anti-transitive, no such agent $j$ is also the reviewer of agent $i$'s proposal. Therefore, no change implemented by $i$ in their nominations will influence their chances of being selected. □

We observe that Algorithm 2 is *an* instance of an assignment that guarantees an anti-transitive assignment and others may exist. Indeed, as discussed in Section 2 we assume that we do not have any notions of assignment quality in our work. However, an interesting direction for future work would be to investigate how the anti-transitive assignment requirement may interact with assignment quality [53].

**Proposition 1.** *Algorithm 2 produces an anti-transitive review assignment.*

**Proof.** We need to show that for all agents $i, j, k$ in a review assignment, if $i$ reviews $j$ and $j$ reviews $k$, then $i$ does not review $k$.

Consider a review assignment produced by Algorithm 2 and suppose we have agents $i, j, k$ such that $i$ reviews $j$ and $j$ reviews $k$. All agents are partitioned into a balanced bipartite graph $(X, Y)$ at the beginning of the algorithm so, without loss of generality, assume $i \in X$. Since the graph remains bipartite throughout the algorithm and the final assignment simply orients the edges, we must have $j \in Y$ and then $k \in X$.

Hence, both $i$ and $k$ are both in the same part of the graph (namely, $X$) and so there is no edge between them. In other words, $i$ cannot review $k$. □

We also want the algorithm to be *monotonic*, that is, having better reviews does not hurt the chances of selection.

**Table 1**
Review profiles that lead to no agent being selected when the nomination quota is increased.

| Reviewer | ✓ | ✗ | ✗ | Weight |
|---|---|---|---|---|
| **1** | 2 | 3 | 4 | 0 |
| **2** | 1 | 3 | 4 | 1 |
| **3** | 1 | 4 | 2 | 1 |
| **4** | 1 | 2 | 3 | 1 |

(a)

| Reviewer | ✓ | ✓ | ✗ | Weight |
|---|---|---|---|---|
| **1** | 2 | 3 | 4 | 0 |
| **2** | 1 | 3 | 4 | 0 |
| **3** | 1 | 4 | 2 | 0 |
| **4** | 1 | 2 | 3 | 0 |

(b)

**Proposition 2.** PeerNomination *is monotonic.*

**Proof.** Suppose $j$ is reviewed by $i$ and compare the probability of selecting $j$ given the original review of $i$ vs. a modified one where $j$ is ranked higher by $i$. Note that for every $i$, $w_i \geq 0$, hence a nomination always has a non-negative impact on the proposal. There are three cases:

(i) $j$ was already inside the integer part of the nomination quota in the original ranking or $j$ is still (after modification) completely outside of the nomination quota in the modified review. In both cases $j$ was already certain to be nominated or not nominated, respectively, by $i$, hence their probability does not change.
(ii) $j$ moves from being a fractional nominee to being a full nominee increasing the chances of nomination (by $1 - (k_q - \lfloor k_q \rfloor)$), hence increasing their chances of selection.
(iii) $j$ moves from being not nominated to being fractionally nominated increasing the chance of nomination (by $k_q - \lfloor k_q \rfloor$), hence increasing the chances of selection.

In all cases $j$'s chances of selection do not decrease, completing the proof. □

Unit PeerNomination is also committee monotonic, that is, increasing the target quota $k$ does not hurt the chances of selection for an agent. However, this is no longer true with other weighting schemes.

**Proposition 3.** PeerNomination *with Unit weights is committee monotonic.*

**Proof.** Fix a review assignment and profile, and suppose we increase the target number of agents to select, $k$. This increases the nomination quota of each reviewer and, since the review profile is fixed, each agent's sum of nominations does not decrease. So any previously selected proposal is still selected. □

**Proposition 4.** PeerNomination *is not committee monotonic.*

**Proof.** As a counter-example, we present an instance of a peer-review problem and a weighting scheme. Consider an instance given by Table 1, with the objective underlying ground truth $1 \succ 2 \succ 3 \succ 4$. Here we have a set of 4 agents, each of whom reviews everyone else (i.e., $n = 4$ and $m = 3$). We augment PeerNomination with a simple weighting scheme: if the reviewer's nominations are also nominated by all other reviewers, we set their weight to 1; otherwise we set their weight to 0. In other words, if there is any disagreement between the reviewers, we discredit all reviewers.

Now suppose $k = \frac{4}{3}$, giving the nomination quota of 1, and that the review profile is given in Table 1a. Reviews are accurate, except agent 3 placed agent 4 over 2. As shown in the table, before the reweighting, agent 1 is nominated 3 times (in a complete agreement) while agent 2 is nominated once. However, since agent 1 is the only one who nominated agent 2, their weight is set to 0, effectively nullifying their nomination. Hence, only agent 1 is selected.

Now suppose we extend $k$ to $\frac{8}{3}$, giving the nomination quota of 2, as shown in Table 1b. Now, every reviewer nominates 2 proposals and, for each reviewer, there is at least 1 proposal that is not nominated by someone else. Hence, according to the weighting schemes, each reviewer receives the weight of 0, resulting in no nominations and hence no selection.[5] □

In addition, PeerNomination is, trivially, also *anonymous* (permuting agents does not matter) and satisfies *non-imposition*, i.e., any $k$ proposals can be selected given an appropriate nomination.

### 5.2. Expected size and slack parameter

In order to understand the interplay between the number of selected agents and the slack parameter, we now derive the expected size of the winning set returned by PeerNomination as a function of $n, m$ and $k$, assuming Unit weights and no

[5] Technically, PeerNomination uses non-strict majority for selection and every agent achieves the selection threshold of 0. However, in the practical implementation, we do not select agents who receive 0 nominations.

noise. The expression reached by the proposition below is not, on its face, very insightful. But it will allow us to draw out the various parameters and see their effects.

**Proposition 5.** *Assume Unit* PeerNomination *is run on a peer review instance with parameters $n, m, k$ under the truthful profile and no noise. Then the probability of selection for an agent in the ground truth position $r$ is given by*

$$\mathbb{P}[accept \mid R = r] = \sum_{i=\lceil m/2 \rceil}^{m} \binom{m}{i} q_r^i (1 - q_r)^{m-i}, \tag{1}$$

*where $q_r(n, m, k)$ is the agent's probability to be nominated by one reviewer:*

$$q_r := \sum_{y=1}^{\lfloor k_q \rfloor} \mathbb{P}[Y = y \mid R = r] + \left(k_q - \lfloor k_q \rfloor\right) \mathbb{P}[Y = \lfloor k_q \rfloor + 1 \mid R = r]. \tag{2}$$

**Proof.** Recall that the algorithm is run on an $m$-regular assignment and we assume reviews are truthful. We also assume the assignment is sampled uniformly, and thus each review bundle is equally likely to be assigned to any reviewer. First, consider the probability of being in position $y$ in the sample of size $m$, given position $r$ in the ground truth ranking. When drawing the sample, we need to choose $y - 1$ individuals out of $r - 1$ that are above agent $r$ in the ground truth, and then choose $m - y$ out of $n - r$ that are worse. In total, as expected, we are choosing $m - 1$ other agents out of $n - 1$. Hence:

$$\mathbb{P}[Y = y \mid R = r] = \binom{r-1}{y-1}\binom{n-r}{m-y} \Big/ \binom{n-1}{m-1},$$

where $Y$ is a random variable representing the position in the review bundle and $R$ is a random variable representing the ground truth position. In order to proceed with the analysis, we need to make a simplifying assumption that the probability above is independent for each bundle. This is not true in general,[6] however our empirical data shows that its effect is negligible for large $n$.

Denote now the nomination quota by $k_q := \frac{k}{n} m$ and recall that in any given review bundle, top $\lfloor k_q \rfloor$ agents are nominated for certain and the next position is nominated with the probability of $k_q - \lfloor k_q \rfloor$. Hence, the probability of being nominated in any bundle from position $r$ in the ranking is, independently:

$$q_r := \sum_{y=1}^{\lfloor k_q \rfloor} \mathbb{P}[Y = y \mid R = r] + \left(k_q - \lfloor k_q \rfloor\right) \mathbb{P}[Y = \lfloor k_q \rfloor + 1 \mid R = r].$$

Since each review bundle can be regarded as a Bernoulli trial with probability $q_r$ and to be accepted an agent has to be nominated $\lceil m/2 \rceil$ times, the probability of being accepted from position $r$ is given by the cumulative Binomial distribution:

$$\mathbb{P}[\text{accept} \mid R = r] = \sum_{i=\lceil m/2 \rceil}^{m} \binom{m}{i} q_r^i (1 - q_r)^{m-i}. \quad \square$$

An illustration of acceptance probabilities as a function of the ground truth position is shown in Fig. 6. We can see that agents that are well inside the top $k$ are almost certain to be accepted while those well outside of the top $k$ are almost certain to be rejected. The width of the interval around the top $k$ for which the probability is away from the extremes is dictated by $m$. Higher $m$ reduces uncertainty by providing more "trials" for each agent and so narrows the interval.

We can now use the derived probability of acceptance to calculate the expected size of the accepting set. Since every individual is accepted independently with probability $\mathbb{P}[\text{accept} \mid R = r]$ and contributes 1 to the size if they are accepted, the expected size is given by:

$$\mathbb{E}[\text{accepting size}] = \sum_{r=1}^{n} \mathbb{P}[\text{accept} \mid R = r].$$

The complexity of this expression makes it difficult to analyse it explicitly. However, Fig. 7a shows a typical behaviour of the expected size as a function of $m$.[7] We observe that this approaches $k$ as $m$ increases. However, for small values of $m$ the

[6] Suppose we have a peer selection instance with $n = 5$ and $m = 3$, where agents are labelled by their ground truth position. There are 5 review bundles in total, one for each agent. Agent 1 must be in 3 of them and must be placed first in each according to the accurate reviewers. Now consider agent 2. The probability that they are placed first, according to the calculation above, is $\frac{1}{2}$, however, at least one of their bundles must contain agent 1, hence the probability of agent 2 placing first in all bundles is 0.

[7] Note the figure's y-axis begins at 26 – the variations are much milder than a cursory look implies.

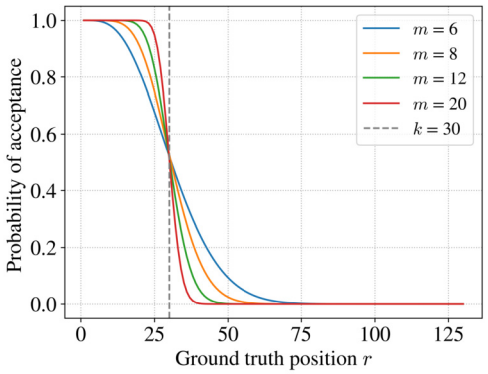
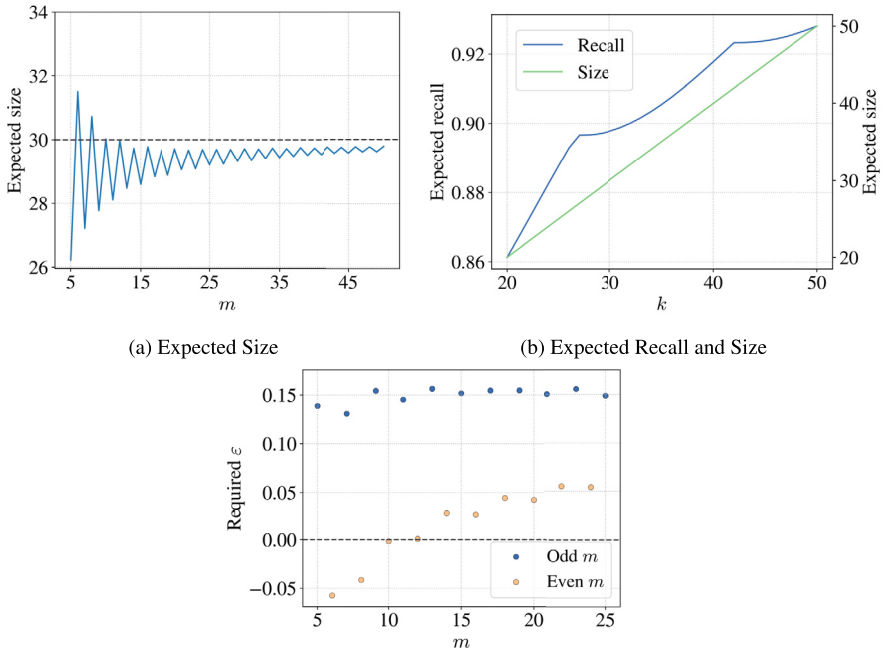



**Fig. 6.** Probability of being accepted by the algorithm given the position in the ranking when $n = 130$ and $k = 30$.




(a) Expected Size

(b) Expected Recall and Size

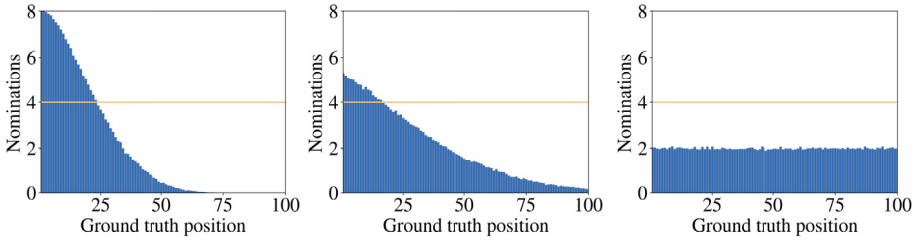
(c) Required Slack Parameter

**Fig. 7.** (a) Expected size of the accepting set returned by the algorithm when $n = 130, k = 30$ and varying $m$. (b) Expected recall and accepting size when $n = 130$, $m = 9$, $\varepsilon = 0.15$ and varying $k$. The green line shows the expected accepting size and the blue line shows the recall. (c) The slack parameter $\varepsilon$ required to achieve the expected accepting size of $k$. We computed $\varepsilon$ using the method outlined in Section 5.4, with $n = 130, k = 30$ and varying $m$.

expected size can vary significantly from $k$, especially when $m$ is odd, recall that agents need to get a clear majority in this case, making selection more difficult.

To tackle these issues, we introduce an additional parameter $\varepsilon$, that we call the slack parameter, which allows us to control the size of the accepting set more finely. If $\varepsilon$ is set to a non-zero value (usually a positive one), we extend the nomination quota in each review bundle by this amount. Usually this increment simply contributes to the probability that the "fractional nominee" is nominated. For example, in the setting $n = 130, m = 9$ and $k = 30$, Fig. 7a shows the expected size slightly above 27 while our aim is 30. Setting $\varepsilon = 0.13$ yields the expected size very close to 30. For most practical applications $\varepsilon \in [-0.05, 0.15]$, as shown in Fig. 7c, meaning the original algorithm is rather well-behaved. Note that this is in contrast to other inexact mechanisms in the literature: Credible Subset must return no solutions with positive probability [23], while the Dollar Partition method may return as many additional agents as the number of clusters [3].

The above analysis assumes reviewers are accurate and all weights set to 1. If these assumptions fail, we cannot provide any guarantees even for the expected size of the accepting set. It is straightforward to construct marginal cases in which everyone or no one is selected in the worst case scenario, as we see in the next example.




(a) $\varphi = 0$. With no noise in the reviewers' rankings, the number of nominations discriminates between good and bad agents well. Also, around $k$ agents achieve the selection threshold resulting in the correct output size.

(b) $\varphi = 0.95$. With some noise among reviewers, the nominations are much more spread out resulting in fewer agents being selected.

(c) $\varphi = 1$. With completely random reviewers everyone gets the same number of nominations and so no agent is selected.

**Fig. 8.** $n = 100, k = 25, m = 8$. Each plot shows the sum of nominations for each agent in the respective ground truth position, averaged over 1000 simulations. The red line shows the selection threshold of $\frac{m}{2}$.

**Example 4.** Consider the setting with 3 agents with everyone reviewing each other and suppose we want to select one individual (i.e., $n = 3, m = 2$ and $k = 1$). Suppose agent 1 reviews 2 above 3, agent 2 reviews 3 above 1 and agent 3 reviews 1 above 2. The nomination quota with $\varepsilon = 0$ is $\frac{2}{3}$ and every agent is ranked in the first place once. Hence, each agent is selected with probability $\frac{2}{3}$ independently and so there exists a realisation where no one is selected as well as one where everyone is selected.

To ensure that the algorithm returns a reasonable number of agents in expectation, we need to put some assumptions on the population of agents. Suppose we have a large number of agents, $n$, and a common acceptance rate of 20% (i.e., $k = 0.2n$). If all reviewers are random, we expect the nominations to be spread evenly between the agents, with each agent receiving $0.2m$ nominations in expectation. Hence very few agents will achieve the threshold of $\frac{m}{2}$ and be selected. However, in a more realistic scenario where the reviewers are able to discriminate between agents, it is likely that a good number of agents will reach the required acceptance threshold (see Fig. 8). The unfavourable performance in the noisy settings is the motivation for the reviewer weights, as will be seen below (in particular, in the empirical part, Section 6).

Also note that in the definition of the algorithm we stipulate that $\varepsilon$ is part of the input. One could be tempted to calculate $\varepsilon$ after collecting the reviews in order to adjust the output size to be exactly $k$, however this is undesirable for several reasons. First, the run of the algorithm is non-deterministic, hence it might be impossible to find a value of $\varepsilon$ that *guarantees* such output size on every run. Second, and most importantly, this would eliminate strategyproofness since now an agent could estimate that reporting a particular untruthful review force the mechanism to increase $\varepsilon$ and so increase their chances of selection.

In the next section we derive the expected recall of our algorithm and the in Section 5.4 we give practical guidance on setting the slack parameter.

### 5.3. Expected recall

In Section 5.2 we derived the probability of acceptance given a position in the ground truth ranking assuming no noise in the reviewer's reported rankings. In this section, we modify the expression in Proposition 5 to include $\varepsilon$, the slack parameter. To do this, we update the nomination quota when computing $q_r$ in Equation (2). Hence, let $k_q^\varepsilon := k_q + \varepsilon$ and

$$q_r^\varepsilon := \sum_{y=1}^{\lfloor k_q^\varepsilon \rfloor} \mathbb{P}[Y = y \mid R = r] + \left(k_q^\varepsilon - \lfloor k_q^\varepsilon \rfloor\right) \mathbb{P}[Y = \lfloor k_q^\varepsilon + 1 \rfloor \mid R = r]. \tag{3}$$

This gives us $\mathbb{P}[\varepsilon\text{-accept} \mid R = r]$ for each ground truth position by simply replacing $q_r$ in Equation (1) by $q_r^\varepsilon$. The expected size is again given by a similar expression:

$$\mathbb{E}[\text{accepting size}] = \sum_{r=1}^{n} \mathbb{P}[\varepsilon\text{-accept} \mid R = r]. \tag{4}$$

In principle, we can now derive the expected performance of the algorithm. However, since the algorithm's output is inexact, there are multiple performance measures to consider, as is often the case for classification algorithms [5]. For example, we might care about how many agents of the top $k$ according to the ground truth we have selected (*recall*) or we

may want to not select too many agents from outside of the top $k$ (*false positive rate*). We focus on the former, referred to as *accuracy* by Aziz et al. [4]. The connection with classification metrics and exact definitions will be further explored in Section 6.1. In the following theorem, we provide the analytic expression for expected recall of PeerNomination.

**Theorem 2.** *The expected recall of Unit* PeerNomination *in the setting n, m, k is*

$$\mathbb{E}[recall] = \frac{1}{k}\sum_{r=1}^{k}\mathbb{P}[\varepsilon\text{-}accept \mid R = r],$$

*where* $\mathbb{P}[\varepsilon\text{-}accept \mid R = r]$ *is the probability of acceptance for the agent in the ground truth position r given by (3).*

**Proof.** Consider a single run of PeerNomination. Let $X_i$ be a random variable such that $X_i = 1$ if agent $i$ is accepted by the algorithm and $X_i = 0$ otherwise. Since an agent in the top $k$ positions contributes 1 to recall and 0 otherwise, recall is equal to $\frac{1}{k}\sum_{i=1}^{n}\mathbb{1}_{i\leq k}X_i = \frac{1}{k}\sum_{i=1}^{k}X_i$. Now, since $X_i$ is just a Bernoulli random variable,[8] $\mathbb{E}[X_i] = \mathbb{P}(X_i = 1)$. Finally, we have

$$\mathbb{E}[\text{recall}] = \frac{1}{k}\sum_{r=1}^{k}\mathbb{P}[X_r = 1] = \frac{1}{k}\sum_{r=1}^{k}\mathbb{P}[\varepsilon\text{-accept} \mid R = r],$$

as required. □

Again, the complexity of these expressions hinders theoretical analysis but Fig. 7b shows a typical output for different values of $k$. While its performance appears good in isolation, it is important to compare PeerNomination with other peer selection mechanisms which we do in Section 6.

### 5.4. Using the slack parameter in practice

The analytic expression for the expected accepting size of PeerNomination, given in Equation (4) allows us to derive a practical way to estimate the slack parameter $\varepsilon$. Given the setting $(n, m, k)$, let $f(\varepsilon) = \mathbb{E}[\text{accepting size}]$ as given in Equation (4). Since we want $f(\varepsilon) = k$, to estimate the required slack parameter we simply need to find a root of the function $g(\varepsilon) = f(\varepsilon) - k$. Since $f$ is highly non-linear (and not even continuous), an analytic solution for the root is unlikely to be obtainable. However, since $f$ is easy to compute, a good approximation for the root can be obtained quickly using a root-finding algorithm, e.g., using Brent's method. Indeed, in Section 6, we use this method to estimate $\varepsilon$ when running PeerNomination.

### 5.5. Effect of weights

The constructive use of weighting schemes in Algorithm 1 depends on the ability of identifying accurate and inaccurate reviewers, and using this identification to reweight their reviews. Not knowing the ground truth means any identification of accurate/inaccurate agents has to depend on comparing different agents' submitted rankings and nominations. If all agents were accurate no reweighting of agents would be needed, but as the proportion of accurate agents drops the problem becomes more difficult. Still, when a large majority of agents are accurate, the correct opinion is usually the majority if inaccurate agents are providing random rankings. However, if the number of accurate agents is very low, or other agents are actively malicious, identification becomes impossible, as there is no metric to evaluate the agents against.

To provide some intuition to the conceptual underpinnings of our algorithm we now present a simplified setting, and show how our algorithm – even with a very simple, conservative, weighting scheme – is still able to improve over PeerNomination. We start with an $m$-regular assignment, where each agent has one of two types: $\mathcal{A}$, meaning the agent is an accurate reviewer; or $\overline{\mathcal{A}}$, meaning the agent is inaccurate. Recall that in PeerNomination a proposal is selected if a majority of their reviewers approve. We show that a very simple dynamic weighting scheme, only relying on knowing how many times an agent has been in a minority, has a good chance of flipping a decision made by $\overline{\mathcal{A}}$ agents to one made by $\mathcal{A}$ agents, improving on PeerNomination.

Let an $\overline{\mathcal{A}}$-agent be *identified as inaccurate* if they hold the minority opinion in at least $j$ panels. We want to find the probability of the following event: (1) an agent's panel consists of an $\overline{\mathcal{A}}$-majority and (2) enough of the $\overline{\mathcal{A}}$-agents of that majority are *identified as inaccurate*.

We now wish to use this simplified model to derive the mathematical expressions of the probability of surely identifying a "bad" reviewer, and seeing if it can be a worthwhile improvement to the algorithm. While the mathematical expressions we reach are not, on their own, easily analyzable, examining them empirically shows that, indeed, even in this simple model we reach meaningful values.

[8] As explained in Section 5.2, $X_i$ are not exactly independent but this simplifying assumption is reasonable for large $n$.

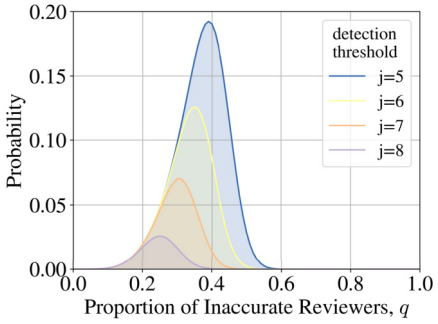



**Fig. 9.** The probability of identifying an inaccurate agent, when $m = 9$, and the threshold for identification is $j$.

Given the noise model, let $q$ be the probability for an agent to be of type $\overline{\mathcal{A}}$. Then the probability that an agent is reviewed by a majority of $\overline{\mathcal{A}}$ agents and that majority is of size $k$ is:

$$q_{B,k} = \mathbb{P}[\overline{\mathcal{A}}\text{-majority of size } k] = \binom{m}{\lfloor m/2 \rfloor + k} q^{\lfloor m/2 \rfloor + k} (1-q)^{m-(\lfloor m/2 \rfloor + k)}.$$

In such a case we would like to identify at least $k$ of the $\overline{\mathcal{A}}$ agents as inaccurate in order to nullify their votes.

We can also find the probability that $\overline{\mathcal{A}}$-agents have a majority of any size:

$$q_B := \mathbb{P}[\overline{\mathcal{A}}\text{-majority}] = \sum_{k=1}^{\lceil m/2 \rceil} q_{b,k} = \sum_{i=0}^{\lfloor m/2 \rfloor} \binom{m}{i} (1-q)^i q^{m-i}.$$

Our simple weighting algorithm labels an agent as an $\overline{\mathcal{A}}$-agent if they are in minority for at least $j$ of their other panels (and are a majority on at least one panel, where decreasing their weight will make a change). The probability of this event, $q_{\text{det}}$, is given by the cumulative binomial probability, keeping in mind that the probability of $\mathcal{A}$-majority on a panel is conditioned on the fact that they contain at least one $\overline{\mathcal{A}}$-agent:

$$\begin{aligned} q'_A &:= \mathbb{P}[\mathcal{A}\text{-majority} \mid \text{there is at least one } \overline{\mathcal{A}}] \\ &= \sum_{i=0}^{\lfloor m/2 \rfloor} \binom{m-1}{i} (1-q)^i q^{m-1-i} \\ \Rightarrow q_{\text{det}} &= \sum_{i=j}^{m-1} \binom{m-1}{i} (q'_A)^i (1-q'_A)^{m-1-i}. \end{aligned}$$

Notice that $\overline{\mathcal{A}}$ may be not just inaccurate but adversarial, in which case we could flip their nomination and only need to do so for $k$ of them. However, we take the safer approach here, which means we need to detect at least $2k$ $\overline{\mathcal{A}}$-agents to correct the decision. The probability of this correcting event is given by the following expression:

$$\mathbb{P}[\text{correction event}] = \sum_{k=1}^{\lceil m/2 \rceil} q_{b,k} \cdot \left( \sum_{i=2k}^{\lfloor m/2 \rfloor + k} \binom{\lfloor m/2 \rfloor + k}{i} q_{\text{det}}^i (1-q_{\text{det}})^{\lfloor m/2 \rfloor + k - i} \right).$$

As desired, we get a significant probability of the correction event, illustrated by Fig. 9. As can be seen, for a wide variety of $q$ and $j$, even our very conservative weighting scheme produces a reasonably high probability of improving some reviews. As we shall see with our designed weighting schemes in Section 4.3, examined by our simulations, even better results can be achieved.

It should be noted that one could produce analogous probabilities of the weighting scheme incorrectly identifying $\mathcal{A}$-agents as $\overline{\mathcal{A}}$-agents. However, for a large enough $j$ (say, $j \geq m/2$), and a majority of $\mathcal{A}$ agents (i.e., $q < n/2$), this number will always be smaller, i.e., the benefit from the reweighting will be positive.

## 6. Empirical analysis

In this section we use an experimental framework to demonstrate that PeerNomination outperforms other mechanisms proposed. In doing so we draw a novel connection between inexact peer selection and the literature on classification in machine learning [5].

### 6.1. Classification measures

The usual and intuitive way to measure the "accuracy" of an exact peer-selection mechanism is counting how many agents from the top $k$ positions in the ground truth have been selected, as a proportion of all $k$ agents selected. This allows us to compare exact peer-selection mechanisms as was done by Aziz et al. [4]. However, comparison with (and between) inexact mechanisms is less obviously done. Since the accepting set is not guaranteed to be exactly of size $k$, any output with more than $k$ agents may artificially increase the performance of the inexact mechanism and the opposite for any smaller output. One option is to measure the performance as a proportion of the output size, however, this approach will overrate outputs that are accurate but much smaller than $k$. Inexactness allows us to view peer selection as a classification problem in which selection means positive classification. We can then view the selected agents from the true top $k$ as true positives and the non-selected agents from outside the true top $k$ as true negatives. We apply the standard classification performance measures [5] such as recall and precision to PeerNomination to analyse its performance.

More formally, let $S$ be the set of agents selected by the algorithm and $S^+ = \{r \in S \mid \text{rank}(r) \leq k\}$ the set of selected agents that are in the true top $k$, i.e., true positives (TP). Similarly, we can use $S^- = \{r \in S \mid \text{rank}(r) > k\}$ for false positives (FP). Hence we can define: $\text{TP} = |S^+|$, $\text{FP} = |S^-| = |S| - \text{TP}$, true negatives $\text{TN} = |\{r \notin S \mid \text{rank}(r) > k\}| = n - k - \text{FP}$, and false negatives $\text{FN} = |\{r \notin S \mid \text{rank}(r) \leq k\}| = n - |S| - \text{TN}$.

We can now look at some of the typical performance metrics: Positive Predictive Value (PPV) (aka Precision), True Positive Rate (TPR) (aka Recall) and False Positive Rate (FPR), defined as follows:

$$\text{PPV} := \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad \text{TPR} := \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad \text{FPR} := \frac{\text{FP}}{\text{TN} + \text{FP}}$$

To see how the nomination quota affects these parameters in PeerNomination we use the Precision-Recall (PR) and Receiver-Operator Characteristic (ROC) curves. These curves show the trade off between sensitivity (TRP) and inclusivity (FPR). We use the slack parameter $\varepsilon$ as the sensitivity threshold akin to the probability threshold in the machine learning literature (see e.g., [18]). So we vary $\varepsilon$ such that the nomination quota varies between 0 and $m$ and measure the Precision, Recall and False Positive Rate at each value. An example is presented in Fig. 11.

The curves show the trade off between sensitivity (TRP) and inclusivity (FPR): As we follow the ROC curve, which corresponds to gradually increasing the nomination quota, the (TPR) increases quickly. That is, by adding few extra agents we already improve significantly the selection of the true top $k$ proposals. On the other hand, we can still achieve TPR of around 0.8 with the FPR very close to 0. This shows that we can select around 80% of the proposals in the true top $k$ if we concentrate on not selecting the "undeserving" individuals, i.e., those that fall outside the true top $k$. While the curves are interesting on their own, we want to be able to compare them to other peer-selection mechanisms, so an important direction is finding a generalizable way of constructing curves for other peer-selection mechanisms.

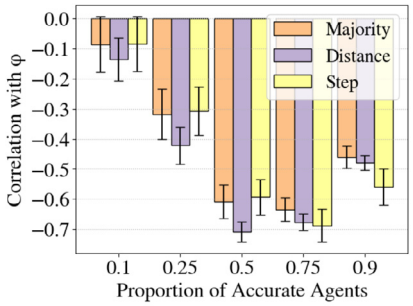
### 6.2. Experimental setup

We extend the testing framework developed by Aziz et al. [4] and using methods from PrefLib [31]. As in Aziz et al. [4], we set $n = 120$ and tested the algorithm on various values of $k$ and $m$. The test values for $k$ were $15, 20, 25, 30, 35$ and the test values for $m$ were 5 to 11.[9] For each setting, we tested algorithms across various noise levels. For the comparison, we included PeerNomination paired with the various weighting schemes introduced in Section 4.3 (including Unit) and also ExactDollarPartition as the state-of-the-art strategyproof peer selection algorithm. Mattei et al. [29] provides comparison of Unit PeerNomination against other peer selection algorithms in a low-noise setting.

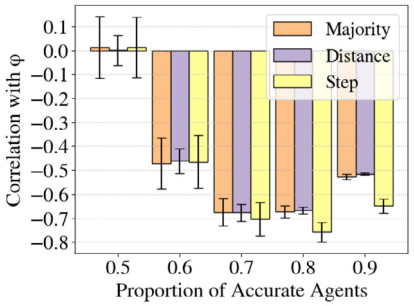
Both ExactDollarPartition and PeerNomination (using Algorithm 2) rely on partition-based assignments. However, Algorithm 2 partitions agents into 2 clusters, while ExactDollarPartition tends to perform best when the number of clusters is 4. For this reason, we decided to generate our $m$-regular assignments with $l = 4$ clusters using the algorithm by Aziz et al. [4]. This might make weighted PeerNomination non-strategyproof but ensures that the performance of ExactDollarPartition is not crippled to ensure a fair comparison. In practice, the performance of PeerNomination does not depend on the type of assignment. We include the performance of PeerNomination paired with Algorithm 2 separately in Fig. 14. In fact, we observe that PeerNomination tends to perform slightly better when the assignment is generated by Algorithm 2.

We use two different settings to model the noise among the reviewers – one with random reviewers and one with contrarian (or adversarial) reviewers. In both settings, we partition the population into accurate and inaccurate (random or contrarian) reviewers and generate individual noisy rankings using Mallows noise [28], as discussed in Section 2.3. In the random case, the dispersion parameter of accurate reviewers is $\varphi = 0.5$, while the random reviewers have $\varphi = 1$. In the contrarian case, the values are 0.8 and 1.2, respectively. In our experiments, we gradually increase the proportion of inaccurate reviewers to test the robustness of the algorithms to noise. Note that while the reviewers' individual rankings are noisy, we assume that their reporting is truthful with respect to their beliefs. Since the tested algorithms are strategyproof, the truthful profile is an equilibrium one.

[9] In Figs. 12, 13 and 14, we only present the results for $k = 25$ and $m = 9$, however these are representative of other settings.




(a) Agents produced from Mallows distribution with $\varphi = 0.5$ for the share shown, the rest with $\varphi = 1$.




(b) Agents produced from Mallows distribution with $\varphi = 0.8$ for the share shown, the rest with $\varphi = 1.2$.

**Fig. 10.** Spearman correlation of the weights from the different weighting schemes with the underlying $\varphi$ of each agent. The bars represent the mean and the standard deviation over 1000 simulations of the weights.

To summarise, a single simulation consists of the following steps:

1. Generate a random $m$-regular assignment matching reviewers to proposals.
2. Determine the type, i.e., accurate or inaccurate, as well as the rankings of each reviewer for their bundle of proposals using a Mallows model.
3. Run each algorithm on the generated instance and measure their performance, e.g., precision and recall.

The experiment was repeated 1000 times for each setting, after which the average recall was calculated giving us high confidence in our results. For PeerNomination, we used theoretical estimates of $\varepsilon$ to achieve the right expected size of the accepting set. The error bars in Figs. 12 and 13 represent 1 standard deviation.

We observe that in our test, ExactDollarPartition was given access to the partial (noisy) rankings of the Mallows model, while PeerNomination only used the simple rule that reviewers approve the top half of their reported order. Hence, our results show that PeerNomination is capable of performing as well or better than ExactDollarPartition even in the presence of less information. An interesting direction for future work would be a complete analysis of the possible reporting spaces, e.g., partial rankings, full rankings, utilities, approvals, and the impact of those reports on overall algorithm performance.

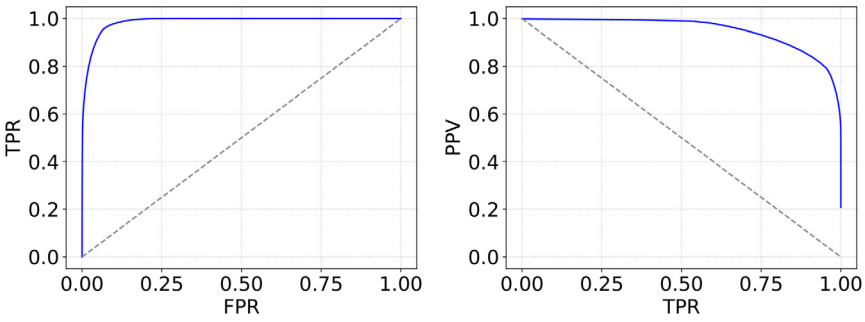
In another testing setup we adopted a slightly different procedure in order to ensure a fair comparison. In each simulation, we generate a random instance, run PeerNomination using the target $k$ as an input, measure the actual size of the output and then run EDP using this actual winning set size as the input target size $k$ for EDP. This ensures that during each simulation both algorithms return the same number of winning proposals. The results of this comparison are presented in Fig. 13.

### 6.3. Results

#### 6.3.1. Random reviewers

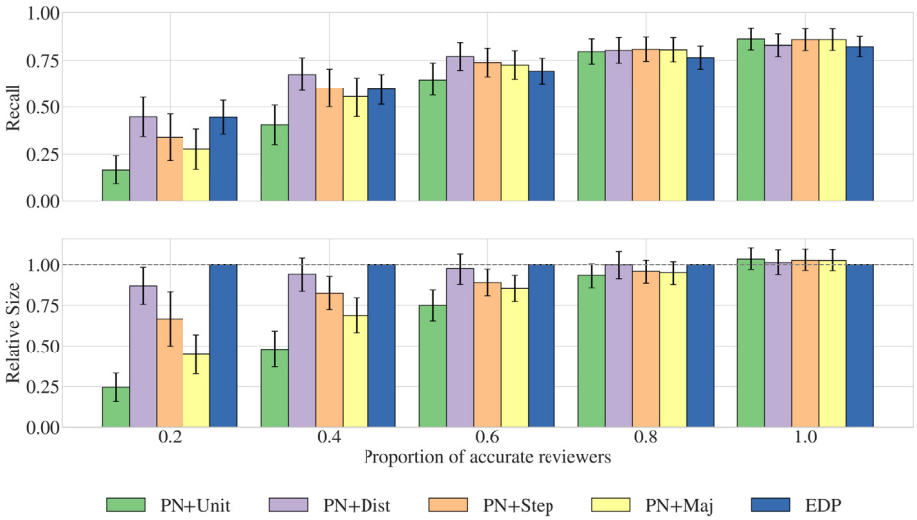
Fig. 12a compares the performance of PeerNomination with selected weighting schemes – as presented in Section 4.3 – and ExactDollarPartition. It can be observed that when the proportion of accurate reviewers is high, i.e., in the 0.8 and 1.0 range, the tested weighting schemes show practically no improvement over Unit. This is a setting with barely any noise, where the weighting schemes behave as desired, without overfit. It can also be observed that all weighting schemes outperform ExactDollarPartition. When the proportion of random reviewers rises, i.e., in the 0.4 and 0.6 range, Unit PeerNomination is underperforming compared to all other weighting schemes. At 0.2, the imbalance decreases further. For instance, PeerNomination with Distance is 2.28 times more accurate than PeerNomination with Unit. We can see that the advantage of weighting schemes over EDP is also evident in the 0.4 setting where despite the lower output size, PeerNomination with Distance achieves higher recall. The ability of the weighting schemes to discriminate between reviewers is further supported by Fig. 10, which demonstrates that our metrics strongly correlate with the underlying $\varphi$. This means that our metrics are able to identify inaccurate reviewer with reasonable certainty.

In general, weighting schemes are much better than Unit at keeping the output size close to the desired $k$, with Distance keeping the output size consistent across all levels of noise. At the same time, Unit's output size is reduced drastically, by a factor of 4, which is explained by Fig. 8: noisier reviewers tend to agree on nominations less, meaning that nominations are more spread out and so few proposals reach the selection threshold. In addition, the much greater recall of Distance and other weighting schemes indicates the additional selected agents are usually the deserving ones. Between the weighting schemes, Distance manages to gain more and more advantage as the noise levels increase as it is the most fine-grained and aggressive one, i.e., it decreases the weight of inaccurate reviewers most severely compared to the other schemes. This allows Distance to both identify the inaccurate reviewers and maintain consistent output size. It has to be noted that, when

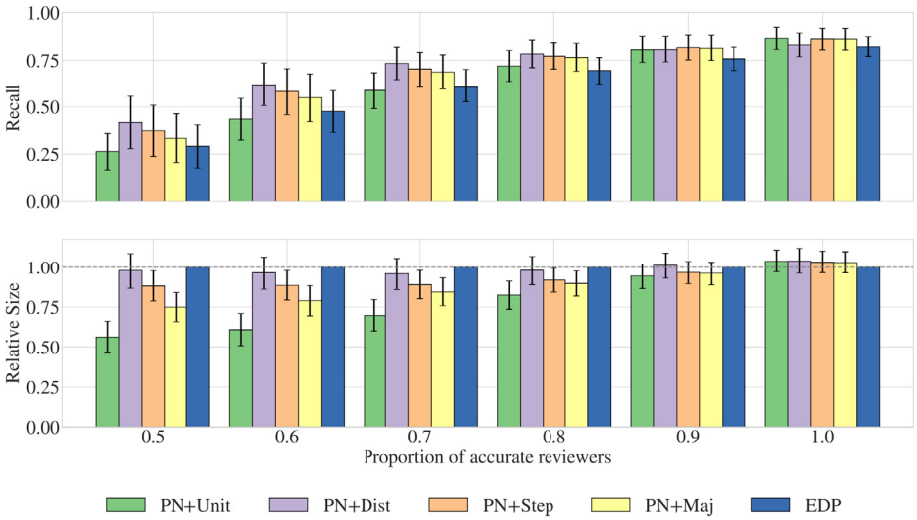



(a) ROC Curve (b) PR Curve

**Fig. 11.** ROC and PR curves for PeerNomination. They were computed empirically with $n = 120, m = 8, k = 25$.

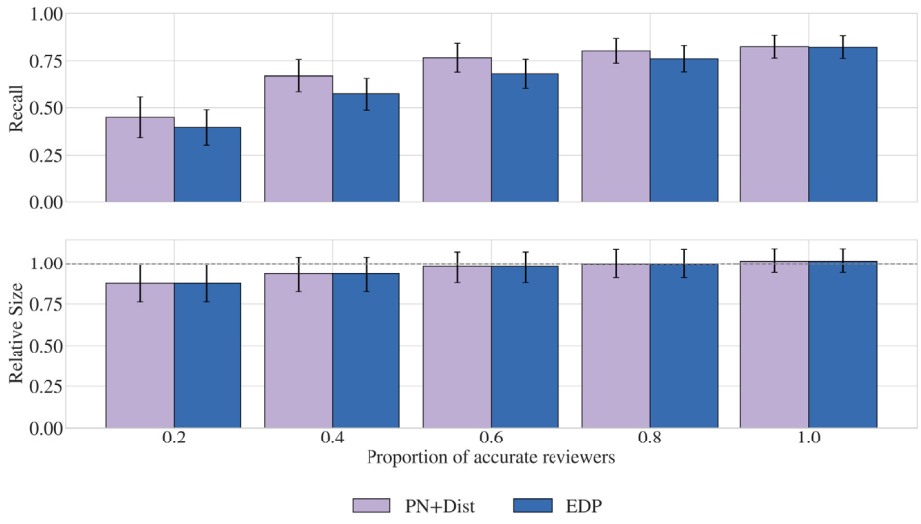



(a) Performance comparison with random reviewers.

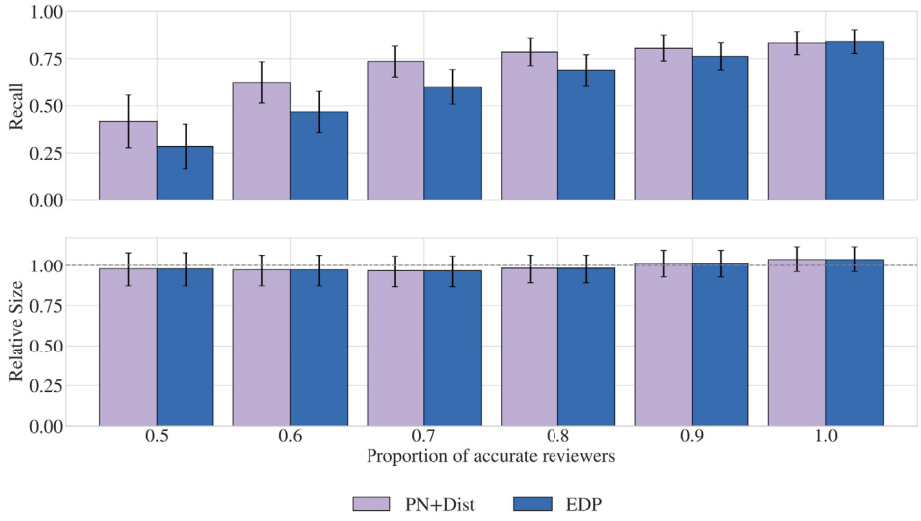



(b) Performance comparison with contrarian reviewers.

**Fig. 12.** Performance comparison for $n = 120, k = 25, m = 9$.




(a) Performance comparison with random reviewers.




(b) Performance comparison with contrarian reviewers.

**Fig. 13.** Results of the forced size experiment, where PeerNomination is run first to set EDP's target size, so that both are always guaranteed to return the same number of agents. The parameters are set to $n = 120, k = 25, m = 9$.

noise is low, Distance tends to slightly worsen its performance. For instance, at 1, even if staying above EDP, PeerNomination with Distance achieves 4% lower recall than PeerNomination with Unit.

*6.3.2. Contrarian reviewers*

When we work with settings with contrarian reviewers, the weighting schemes are even more effective. The results of our study are shown in Fig. 12b. Notice how we analyse the proportion of accurate reviewers from at 0.5, keeping the contrarians as a minority. Again, at low noise levels, i.e., 1, the results match, as expected, the observations made in the previous paragraph. As the proportion of contrarian reviewers rises, e.g., at the 0.7 point, we can observe that all schemes outperform EDP, with Distance reaching a 20% performance increase when compared to EDP. Interestingly, at moderate levels of noise, e.g., 0.7 and 0.8, Distance shows similar performance to the case with random reviewers. Even though the reviewers are, on average, more diverging, they are also easier to detect. Even when half of the population is contrarian, Distance gets impressively close to the theoretical maximum of 50% as shown in Fig. 12b. Beyond this point, the contrarian point of view becomes a majority, and there is no way to retrieve the original ground truth.

It is finally worth noting that the bottom graphs in Figs. 12a and 12b show that PeerNomination tends to return a slightly larger than $k$ set, on average, in the noiseless setting, usually $< 1$ additional agent. This may give the impression that the results might inflate the performance of PeerNomination compared to an exact algorithm, such as

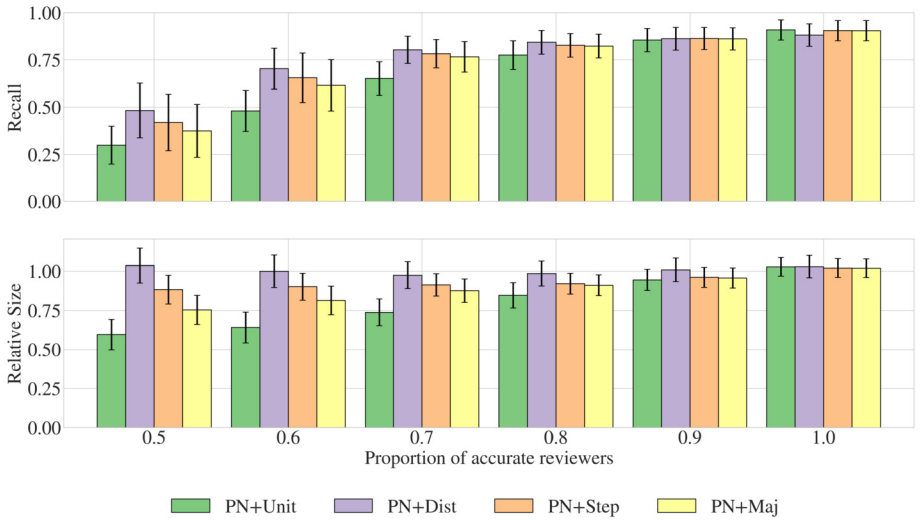



**Fig. 14.** Performance comparison for $n = 120, k = 25, m = 9$ in a contrarian setting between the PeerNomination weighting schemes. Here the assignment for each iteration was generated using Algorithm 2, ensuring that the mechanisms are strategyproof.

ExactDollarPartition, provided that the extra agents are chosen correctly. Section 6.3.3, where we perform a fair comparison between the two, shows that this makes a negligible difference and only in some settings.

*6.3.3. Fair tests*

The fair testing setup described above only allows us to test an *inexact* algorithm, which does not always output the target number of agents. Hence, we chose to compare the best performing weighting scheme of PeerNomination, Distance, against ExactDollarPartition. The results of the fair test, Fig. 13, align with the previous findings. In the noiseless setting, ExactDollarPartition does gain an advantage over PeerNomination paired with Distance, likely due to over-fitting of the weighting scheme. However, as the level of noise increases, Distance gets a clear advantage over ExactDollarPartition. For instance, when 40% of reviewers are contrarian, PeerNomination sees a 34% increase in recall over ExactDollarPartition. The advantage is particularly stark in the contrarian setting, where Distance benefits greatly from the ability to identify and reweight the inaccurate reviewers.

## 7. Discussion and conclusions

We have proposed a novel strategyproof peer selection algorithm – PeerNomination, which weighs reviewers based on their perceived accuracy. The basis for this reweighting is the observation that in most cases, one's reviews are correlated in quality, and we can use the correlation to improve the recall of the overall algorithm. We develop several weighting methods, showing that even straightforward ones can reach high quality outcomes, even under high levels of noise in the reported rankings of the reviewers. Hence, we have shown that PeerNomination achieves state-of-the-art performance on the problem of peer selection.

Given that PeerNomination is constructed in a modular way, there are a variety of weighting and evaluation methods that can be developed for particular settings or noise models. This modularity allows for multiple directions of future development. One possible direction for future work is exploring whether the weighting schemes we have developed so far are optimal. As we have already seen from our results, different schemes perform best under different conditions. For example, a more aggressive approach might come out on top when the majority of reviewers are inaccurate but lose out to a more forgiving weighting when there is not much noise in the reported rankings. Indeed, while Distance (with a rather aggressive approach) seemed as the most promising weighting scheme when the share of good reviewers was not very high, both Majority and Step are much simpler and easier to calculate and administer.

While we have examined the weighting schemes under various variants of the Mallows model, other weighing schemes may behave differently under different distributions. In particular, Random Utility Models (RUMs) [2], which have been used extensively in social choice, provide an interesting alternative to consider. Moreover, each particular peer review setting has its own assumptions and sources of noise, and it might be possible to develop a weighing scheme that adapts well to its particular situation. Even among the proposed weighting schemes, each has free parameters that can be optimised. Reviewers' assessment patterns can also be learnt over time and we can use this information as an input for PeerNomination. The use of convolutional neural networks to infer peer assessment patterns is currently under study [34].

On the other hand, the weighting schemes can be easily decoupled form PeerNomination and adapted to be used with other strategyproof peer selection mechanisms. For example, considering the already good performance of ExactDollarPartition under noisy conditions, it would be interesting to see whether it benefits from reweighting in a sim-

ilar way to PeerNomination. Note that we did not run this test explicitly in this paper as adapting ExactDollarPartition to use reweighting schemes would require us to completely redesign the ExactDollarPartition clustering and assignment mechanisms. Observe that while for PeerNomination Algorithm 2 ensures a particular structure to the allocation of papers, this algorithm does not work for ExactDollarPartition. Specifically, using the $k$-partition balanced assignment procedure that is part of ExactDollarPartition with 3 or more partitions may result in the case illustrated in Fig. 5, causing none of the proposed weighting schemes to be impartial for ExactDollarPartition.

There are many avenues for additional theoretical work on summarising and quantifying the effects of strategyproofness of peer review mechanisms, as well as the effects of weighting schemes. We have seen different trade-offs, e.g., relaxing exactness or imposing constraints on the review assignment, employed to ensure strategyproofness. It is important to specify these assumptions precisely and, if possible, quantify their effect on the performance of peer selection theoretically. For example, a direction for future work lies in evaluating non-strategyproof reweighting methods such as ones based on expectation maximisation (EM) algorithms including GLAD [50], Dawid-Skene [15], and even PageRank [36,22]. While none of these methods maintain the first-order requirement for strategyproofness in our setting, it would be interesting to evaluate recall of these methods as compared to strategyproof methods.

In the context of peer reviewing, we see strategyproofness as a non-negotiable desideratum, which we strove for when designing our algorithm even in the presence of noise. Our assignment and weights are selected in such a way that there is no incentive for reviewers to gain an advantage by discrediting other reviewers or submitting insincere reports, as other peer selection mechanisms used in practice may allow [33]. We only use non-strategyproof systems as a benchmark, to compare the performance of our algorithm to an ideal optimum, however we do not advocate the use of such systems in practice. Although beyond the scope of this paper, we believe that the study of non-strategyproof systems can lead to important discoveries in terms of "cost of strategyproofness", i.e., what we are sacrificing in terms of optimality in order to deploy systems where reviewers do not have an incentive to lie. This may even lead to "acceptable weakenings" of strategyproofness, if optimality gains are proved to be significant. Currently, we do not have theoretical guarantees that strategyproof peer selection algorithms produce close to optimal results, within the respective constraints. Likewise, we do not know that a relaxation of strategyproofness is the only way to obtain significant gains in terms of recall. Hence, it is unclear whether the focus should lie in improving the recall of the current mechanisms or in developing new mechanisms that rely on weaker assumptions. We believe this to be an important future direction for research in peer selection.

If exactness, rather than strategyproofness, is the objective, there may be little reason to go beyond a Borda-like mechanism and using either pairwise Shah and Wainwright [44] or complete rankings [13]. Additionally, past work including that of Mattei et al. [29], Aziz et al. [4], and Kahng et al. [20] provide some empirical comparison of Borda and various strategyproof mechanisms, which sheds some light on the trade-off between exactness and strategyproofness. Theoretical quantification of the various trade-offs between, e.g., exactness, impartiality and optimality, is indeed an exciting direction for future work. Additionally, when we do not use strategyproofness mechanisms there is the question of what exactly are agents incentivised to report [45]. An interesting future direction is to analyse these mechanisms from a mechanism design viewpoint [35] where we align the incentives of the agents in a way that might not be strategyproof.

Finally, there has been a lack of real-world data analysis in the field of peer selection. The challenge lies in that in the domains we considered, such as academic peer review, the best available ground truth can only be acquired from subjective opinions. For instance, to evaluate the submissions to a major conference, an independent expert panel would have to evaluate and agree on the ranking of thousands of papers – and even that would not guarantee the best possible approximation of the ground truth. Nevertheless, any validation with real-world data would give a much better idea of the true performance of the current mechanisms as well as help us create more realistic artificial models.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

[1] N. Alon, F. Fischer, A. Procaccia, M. Tennenholtz, Sum of us: strategyproof selection from the selectors, in: Proceedings of the 13th Conference on Theoretical Aspects of Rationality and Knowledge (TARK), 2011, pp. 101–110.
[2] H. Azari, D. Parks, L. Xia, Random utility theory for social choice, Adv. Neural Inf. Process. Syst. (2012) 25.
[3] H. Aziz, O. Lev, N. Mattei, J.S. Rosenschein, T. Walsh, Strategyproof peer selection: mechanisms, analyses, and experiments, in: D. Schuurmans, M.P. Wellman (Eds.), AAAI, AAAI Press, 2016, pp. 397–403.
[4] H. Aziz, O. Lev, N. Mattei, J.S. Rosenschein, T. Walsh, Strategyproof peer selection using randomization, partitioning, and apportionment, Artif. Intell. 275 (2019) 295–309, https://doi.org/10.1016/j.artint.2019.06.004.
[5] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
[6] A. Bjelde, F. Fischer, M. Klimm, Impartial selection and the power of up to two choices, ACM Trans. Econ. Comput. 5 (4) (2017) 1–20, https://doi.org/10.1145/3107922.
[7] J. Bohannon, Who's afraid of peer review?, Science 342 (6154) (2013) 60–65.
[8] N. Bousquet, S. Norin, A. Vetta, A near-optimal mechanism for impartial selection, in: Proceedings of the 10th International Workshop on Internet and Network Economics (WINE), in: Lecture Notes in Computer Science (LNCS), 2014, pp. 133–146.
[9] S. Brams, P. Fishburn, Approval voting, Am. Polit. Sci. Rev. 72 (1978) 831–847.

[10] I. Caragiannis, G.A. Krimpas, A.A. Voudouris, Aggregating partial rankings with applications to peer grading in massive online open courses, in: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS, ACM, 2015, pp. 675–683.
[11] I. Caragiannis, G.A. Krimpas, A.A. Voudouris, How effective can simple ordinal peer grading be?, ACM Trans. Econ. Comput. 8 (3) (2020), https://doi.org/10.1145/3412347.
[12] L. Charlin, R.S. Zemel, C. Boutilier, A framework for optimizing paper matching, in: UAI 2011, Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2011, pp. 86–95.
[13] W. Chen, R. Zhou, C. Tian, C. Shen, On top-$k$ selection from $m$-wise partial rankings via Borda counting, IEEE Trans. Signal Process. 70 (2022) 2031–2045.
[14] G. de Clippel, H. Moulin, N. Tideman, Impartial division of a dollar, J. Econ. Theory 139 (2008) 176–191.
[15] A.P. Dawid, A.M. Skene, Maximum likelihood estimation of observer error-rates using the em algorithm, J. R. Stat. Soc., Ser. C, Appl. Stat. 28 (1) (1979) 20–28.
[16] L. De Alfaro, M. Shavlovsky, Crowdgrader: a tool for crowdsourcing the evaluation of homework assignments, in: Proceedings of the 45th ACM Technical Symposium on Computer Science Education (ACM:CACM), 2014, pp. 415–420.
[17] F. Fischer, M. Klimm, Optimal impartial selection, in: Proceedings of the 15th ACM Conference on Economics and Computation (ACM-EC), 2014, pp. 803–820.
[18] P.A. Flach, Machine Learning - The Art and Science of Algorithms That Make Sense of Data, Cambridge University Press, 2012, http://www.cambridge.org/de/academic/subjects/computer-science/pattern-recognition-and-machine-learning/machine-learning-art-and-science-algorithms-make-sense-data.
[19] S. Jecmen, H. Zhang, R. Liu, N.B. Shah, V. Conitzer, F. Fang, Mitigating manipulation in peer review via randomized reviewer assignments, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Annual Conference on Neural Information Processing Systems 2020, NeurIPS, 2020.
[20] A. Kahng, Y. Kotturi, C. Kulkarni, D. Kurokawa, A. Procaccia, Ranking wily people who rank each other, in: Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI), 2018.
[21] M.G. Kendall, A new measure of rank correlation, Biometrika 30 (1/2) (1938) 81–93.
[22] J.M. Kleinberg, Authoritative sources in a hyperlinked environment, J. ACM 46 (5) (1999) 604–632.
[23] D. Kurokawa, O. Lev, J. Morgenstern, A.D. Procaccia, Impartial peer review, in: Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI'15, AAAI Press, 2015, pp. 582–588, http://dl.acm.org/citation.cfm?id=2832249.2832330.
[24] J. Langford, The NIPS experiment, https://cacm.acm.org/blogs/blog-cacm/181996-the-nips-experiment/fulltext, 2015.
[25] J.W. Lian, N. Mattei, R. Noble, T. Walsh, The conference paper assignment problem: using order weighted averages to assign indivisible goods, in: Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI), 2018, pp. 1138–1145.
[26] T.Y. Liu, Learning to Rank for Information Retrieval, Springer Science & Business Media, 2011.
[27] T. Lu, C. Boutilier, Learning mallows models with pairwise preferences, in: Proceedings of the 28th International Conference on International Conference on Machine Learning (ICML), 2011, pp. 145–152.
[28] C.L. Mallows, Non-null ranking models. I, Biometrika 44 (1–2) (1957) 114–130.
[29] N. Mattei, P. Turrini, S. Zhydkov, Peernomination: relaxing exactness for increased accuracy in peer selection, in: Proc. International Joint Conference on Artificial Intelligence (IJCAI), 2020, pp. 393–399, ijcai.org.
[30] N. Mattei, T. Walsh, PrefLib: a library for preferences, in: Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT), 2013, http://www.preflib.org.
[31] N. Mattei, T. Walsh, A PrefLib.Org retrospective: lessons learned and new directions, in: U. Endriss (Ed.), Trends in Computational Social Choice, AI Access Foundation, 2017, pp. 289–309.
[32] R. Meir, J. Lang, J. Lesca, N. Kaminski, N. Mattei, A market-inspired bidding scheme for peer review paper assignment, in: Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI), 2021.
[33] M. Merrifield, D. Saari, Telescope time without tears: a distributed approach to peer review, Astron. Geophys. 50 (4) (2009) 4.16–4.20, https://doi.org/10.1111/j.1468-4004.2009.50416.x.
[34] A.A. Namanloo, J. Thorpe, A. Salehi-Abari, Improving peer assessment with graph convolutional networks, https://arxiv.org/abs/2111.04466, 2021.
[35] N. Nisan, T. Roughgarden, E. Tardos, V.V. Vazirani, Algorithmic Game Theory, Cambridge University Press, Cambridge, 2007.
[36] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank Citation Ranking: Bringing Order to the Web, Technical Report 1999-66, Stanford InfoLab, 1999, http://ilpubs.stanford.edu:8090/422/, previous number = SIDL-WP-1999-0120.
[37] C. Piech, J. Huang, Z. Chen, C. Do, A. Ng, D. Koller, Tuned models of peer assessment in moocs, arXiv preprint, arXiv:1307.2579, 2013.
[38] C. Piech, J. Huang, Z. Chen, C.B. Do, A.Y. Ng, D. Koller, Tuned models of peer assessment in moocs, in: Proceedings of the 6th International Conference on Educational Data Mining (EDM), 2013, pp. 153–160.
[39] P. Resnick, R. Sami, The influence limiter: provably manipulation-resistant recommender systems, in: Proceedings of the 2007 ACM Conference on Recommender Systems, 2007, pp. 25–32.
[40] J. Sabater, C. Sierra, Review on computational trust and reputation models, Artif. Intell. Rev. 24 (1) (2005) 33–60.
[41] T. Schnabel, A. Swaminathan, P.I. Frazier, T. Joachims, Unbiased comparative evaluation of ranking functions, in: Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval, 2016, pp. 109–118.
[42] N.B. Shah, KDD 2021 tutorial on systemic challenges and solutions on bias and unfairness in peer review, in: F. Zhu, B.C. Ooi, C. Miao (Eds.), KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021, ACM, 2021, pp. 4066–4067.
[43] N.B. Shah, B. Tabibian, K. Muandet, I. Guyon, U. Von Luxburg, Design and analysis of the NIPS 2016 review process, J. Mach. Learn. Res. 19 (1) (2018) 1913–1946.
[44] N.B. Shah, M.J. Wainwright, Simple, robust and optimal ranking from pairwise comparisons, J. Mach. Learn. Res. 18 (1) (2017) 7246–7283.
[45] I. Stelmakh, N.B. Shah, A. Singh, Catch me if I can: detecting strategic behaviour in peer assessment, arXiv preprint, arXiv:2010.04041, 2020.
[46] I. Stelmakh, N.B. Shah, A. Singh, H. Daumé III, A novice-reviewer experiment to address scarcity of qualified reviewers in large conferences, in: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI Press, 2021, pp. 4785–4793.
[47] T. Walsh, The PeerRank method for peer assessment, in: Proceedings of the 21st European Conference on Artificial Intelligence (ECAI), Prague, Czech Republic, 2014, pp. 909–914.
[48] J. Wang, N.B. Shah, Your 2 is my 1, your 3 is my 9: handling arbitrary miscalibrations in ratings, in: E. Elkind, M. Veloso, N. Agmon, M.E. Taylor (Eds.), Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems AAMAS, IFAAMAS, 2019, pp. 864–872.
[49] J. Wang, I. Stelmakh, Y. Wei, N.B. Shah, Debiasing evaluations that are biased by evaluations, arXiv preprint, arXiv:2012.00714, 2020.
[50] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, J. Movellan, Whose vote should count more: optimal integration of labels from labelers of unknown expertise, in: Proceedings of the 22nd International Conference on Neural Information Processing Systems, 2009, pp. 2035–2043.
[51] J. Whitehill, T.f. Wu, J. Bergsma, J. Movellan, P. Ruvolo, Whose vote should count more: optimal integration of labels from labelers of unknown expertise, in: Advances in Neural Information Processing Systems, 2009, pp. 2035–2043.
[52] L. Xia, Learning and Decision-Making from Rank Data. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan and Claypool, 2019.
[53] Y. Xu, H. Zhao, X. Shi, N.B. Shah, On strategyproof conference peer review, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI), Macau, 2019, pp. 616–622.