PEERNOMINATION: Relaxing Exactness for Increased Accuracy in Peer Selection

Nicholas Mattei[1] , **Paolo Turrini**[2] and **Stanislav Zhydkov**[2]
[1]Department of Computer Science, Tulane University
[2]Department of Computer Science, University of Warwick
nsmattei@tulane.edu, {p.turrini, s.zhydkov}@warwick.ac.uk

Abstract

In peer selection agents must choose a subset of themselves for an award or a prize. As agents are self-interested, we want to design algorithms that are impartial, so that an individual agent cannot affect their own chance of being selected. This problem has broad application in resource allocation and mechanism design and has received substantial attention in the artificial intelligence literature. Here, we present a novel algorithm for impartial peer selection, PEERNOMINATION, and provide a theoretical analysis of its accuracy. Our algorithm possesses various desirable features. In particular, it does not require an explicit partitioning of the agents, as previous algorithms in the literature. We show empirically that it achieves higher accuracy than the exiting algorithms over several metrics.

1 Introduction

Peer selection, where agents must 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 current methods rely on expert panels, there is increasing attention to how to design trustworthy mechanisms that improve the accuracy and reliability of the outcome, keeping the procedure simple and cheap. The latter is particularly relevant in open online courses [Piech *et al.*, 2013], where hiring professional graders is prohibitively expensive. Indeed, even IJCAI 2020 is implementing a portion of this system, requiring authors who submit papers to agree to be reviewers themselves.

The importance of having an "objective" assessment in conference reviewing has been brought to light by the famous NIPS experiment [Langford, 2015; Shah *et al.*, 2018]: of all papers submitted to NIPS 2014, 10% were reviewed twice by two independent committees which, astonishingly, agreed on less than half of the accepted papers in their pool. Whether the outcome was due to bias, incompetence or simply well-thought disagreement is still unclear. What is clear though is that the current solutions show undesirable properties.

Methods for *impartial* peer selection, where self-interested individuals assess one another in such a way that none of them has an incentive to misrepresent their evaluation, have a long standing tradition in economics, e.g., [Douceur, 2009; Holzman and Moulin, 2013; de Clippel *et al.*, 2008], which has in turn encouraged several groups in artificial intelligence and computer science more broadly to investigate these problems, e.g., [Kurokawa *et al.*, 2015; Alon *et al.*, 2011; Xu *et al.*, 2019; Aziz *et al.*, 2019].

The interest in such methods has culminated in a pilot scheme by the US National Science Foundation (NSF) [Naghizadeh and Liu, 2013], called for by Merrifield and Saari [2009], in which each principal investigator (PI) was asked to rank 7 proposals from other PIs. The rankings were then combined using the Borda score with the additional truth-telling incentive of receiving a bonus the closer one gets to the average of the other reviewers' marks. Though this method is not impartial, and leads to a Keynesian beauty contest [Keynes, 1936], the results were encouraging.

Research in artificial intelligence and economics has led to a number of proposals for algorithms choosing a set of k agents from amongst themselves, commonly known as the peer selection problem. We review some of the most prominent ones here to which we will compare our proposal.

State of the Art. In Credible Subset (CS) [Kurokawa *et al.*, 2015], reviewers assign scores to their allocated proposals and the potential manipulators, i.e., the reviewers that could be within the k funded ones, are also selected to be funded, with a given probability. While the system is strategy-proof, it will yield an empty set of funded proposals in a number of cases [Aziz *et al.*, 2016]. The Dollar Raffle method (DR) [de Clippel *et al.*, 2008] is a well-known peer reviewing protocol consisting of reviewers distributing a score in the interval [0,1] to their reviews rather than independently allocating them as in (CS). DR showed poor accuracy when compared to partition-based methods [Aziz *et al.*, 2019]. In Exact Dollar Partition (EDP) [Aziz *et al.*, 2019] reviewers are clustered at random and rank peers in different clusters. Using a randomized rounding scheme based on the the shares computed with the method of de Clippel *et al.* [2008], the top proposals of each cluster are selected, depending on their clusters' importance. Dollar Partition is strategy-proof and has been shown to be the most accurate available method [Aziz *et al.*, 2019].

We also compare our algorithm against two more basic procedures: Vanilla, which selects the k agents with the highest total Borda score based on the reviews received; and Partition, which, instead, divides the agents into a set of clusters and selects a predetermined number of them from each (typically k divided by the number of clusters) as rated by the agents from

the other clusters. Notice that, unlike Partition, Vanilla is not impartial but is commonly used as a baseline for comparison.

Relevant recent developments with a different focus use voting rules to aggregate ranks (e.g., k-Partite [Kahng *et al.*, 2018], including the Committee [Kahng *et al.*, 2018] and Divide-and-Rank [Xu *et al.*, 2019]) algorithms. Other methods are approval-based but only focus on single agent selection: Permutation [Fischer and Klimm, 2014] and Slicing [Bousquet *et al.*, 2014]. Additional work in this area also focuses on assignment and calibration issues [Wang and Shah, 2019; Lian *et al.*, 2018].

Our Contribution. We present PEERNOMINATION, an impartial peer selection method for scenarios where n agents review and are reviewed by m others, with the goal of selecting k of them. Each proposal 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. This way we relax the exactness requirement, in the sense that our algorithm is not guaranteed to select exactly k proposals every time. However, under some mild rationality assumptions, the algorithm does so in expectation. Unlike other well-known peer reviewing methods, e.g., Exact Dollar Partition (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 [Mallows, 1957; Xia, 2019], exactly deriving its expected accuracy analytically. Moreover, 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 accuracy 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 performance with respect to the accuracy of the accepted set.

Paper Structure. In Section 2 we set up the basic terminology and notation. Section 3 presents our algorithm and its theoretical properties. Section 4 compares its accuracy against the main existing alternatives, under various metrics.

2 Preliminaries

We work with a set of agents $\mathcal{N} = \{1, 2, ..., n\}$ and an order over them, induced by their index, which represents the final ranking the agents would have, if they were to be assessed objectively. We refer to this order as the *ground truth*. Each agent is assigned m other agents to review and is in turn reviewed by m others. We represent such m-regular assignment as a function $A : \mathcal{N} \to 2^{\mathcal{N}}$ and denote i's review pool as $A(i)$, while $A^{-1}(i)$ denotes i's reviewers. It is worth noting that while generating a random m-regular assignment is easy for small m (by generating an m-regular bipartite graph), sampling one uniformly is non-trivial and is an active area of study (e.g., see [Berger and Müller-Hannemann, 2010]). In this paper, we assume uniform sampling to make our theoretical analysis tractable in Section 3 but not for experiments in Section 4. In practice, we observed negligible effect on the performance of algorithms when using different assignment-generating procedures. In real-world settings, agents can only review a limited number of proposals or papers so m is typically small and constant, given n.

Each reviewer i submits a ranking of their review pool $A(i)$, which we represent as a strategy $\sigma_i : A(i) \to \{1, ..., m\}$, where $\sigma_i(j)$ gives the rank of j given by i in i's review pool. A collection of all declared strategies is called a *profile* and is denoted by σ. The unique profile which is consistent with the ground truth is called *truthful*. After the individual preferences are declared, they are aggregated to select k individuals. We call a peer selection mechanism *impartial* or *strategyproof* if no agent can affect their chances of selection in any assignment using any strategy.

3 PEERNOMINATION

In this section we present PEERNOMINATION and describe its performance analytically.

3.1 The Algorithm

A usual requirement for peer selection mechanisms is that it must return an accepting set exactly of size k [Aziz *et al.*, 2019; Alon *et al.*, 2011; Kahng *et al.*, 2018]. Though some approaches investigated relaxing this assumption [Aziz *et al.*, 2016; Kurokawa *et al.*, 2015], most notably the results by Bjelde *et al.* [2017] show that this relaxation can lead to better optimality approximation. We use this intuition in designing the following algorithm that returns an accepting set of size 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, we expect them to be ranked in the top k proportion (i.e., top $\frac{k}{n}m$) of their review pool by the majority of agents that review them, if these were to report their accurate rankings. We say that an agent is *nominated* by a reviewer if they are in the top k proportion of the reviewer's declared ranking, i.e., their review pool. Likewise, we refer to $\frac{k}{n}m$ as the *nomination quota*. Hence, for every agent j, we look at all reviewers $i_1, ..., i_m$ reviewing j and select j only if they are nominated by the majority of these reviewers.

As $\frac{k}{n}m$ is unlikely to be an integer, we consider an agent *nominated for certain* if they are among the first $\lfloor \frac{k}{n}m \rfloor$ agents in the review pool, where $\lfloor x \rfloor$ denotes the whole part of a positive real number x. If they are in the next position (i.e., $\lfloor \frac{k}{n}m \rfloor + 1$), we randomly consider them nominated with probability $\frac{k}{n}m - \lfloor \frac{k}{n}m \rfloor$, that is, the decimal part of the nomination quota. Lastly, if the number of review pools an agent is part of is even, we require them to be nominated by just half of the review pools, not a strict majority.

A crucial observation is that, since each agent is considered independently for selection, the algorithm is not guaranteed to return exactly k agents. However, we will show that the algorithm is close enough to such number if the reviewers submit reviews that are close enough to the ground truth and, moreover, that truth-telling is an equilibrium outcome, i.e., PEERNOMINATION is impartial.

Algorithm 1 PEERNOMINATION

Input: Assignment A, review profile σ, target quota k, slack parameter ε
Output: Accepting set S

```
Set nomQuota := (k/n)m + ε
for all j in N do
    Initialise nomCount := 0
    for all i ∈ A^{-1}(j) do
        if σ_i(j) ≤ ⌊nomQuota⌋ then
            increment nomCount by 1
        else if σ_i(j) = ⌊nomQuota⌋ + 1 then
            increment nomCount by 1 with probability
            nomQuota − ⌊nomQuota⌋
        end if
    end for
    if nomCount ≥ ⌈m/2⌉ then
        S ← j
    end if
end for
return S
```

The PEERNOMINATION algorithm is presented in Algorithm 1. Note that in the algorithm we introduce the *slack parameter* ε, which extends the nomination quota accordingly. As we show next, this is necessary in some settings to achieve the right expected size of the accepting set.

3.2 Expected Size and Slack Parameter

We now derive the expected size of the accepting set returned by PEERNOMINATION as a function of n, m and k. Since each agent is considered independently, we just need to derive the probability of selection for an agent given their ground truth position. Assume the algorithm is run on an m-regular assignment and the reviews are truthful. Note that we assume such assignment is sampled uniformly and so each review pool is equally likely to be assigned to any reviewer. Firstly, consider the probability of obtaining position y in the sample of size m, given position r in the underlying 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:

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

where Y is a random variable representing the position in the review pool and R is a random variable representing the ground truth position.

Denote now the nomination quota by $k_q := \frac{k}{n}m$ and recall that in any given review pool, 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 pool from position r in the ranking is, independently:

$$\begin{aligned} q_r := &\sum_{y=1}^{\lfloor k_q \rfloor} \mathbb{P}[Y = y | R = r] \\ &+ (k_q - \lfloor k_q \rfloor)\, \mathbb{P}[Y = \lfloor k_q \rfloor + 1 | R = r] \end{aligned} \tag{1}$$

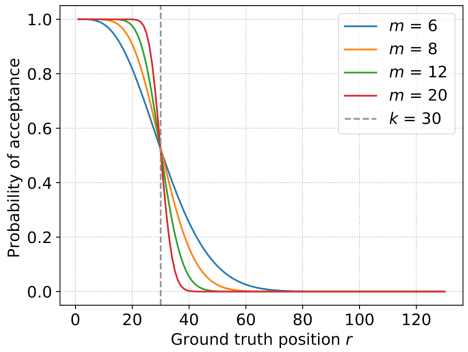


Figure 1: Probability of being accepted by the algorithm given the position in the ranking when $n = 130$ and $k = 30$.

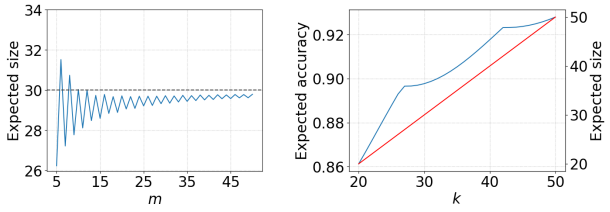
Since each review pool can be regarded as a Bernoulli trial with probability q_r and to be accepted an agent has to be nominated $\lceil m/2 \rceil$ times, the probability of being accepted from position r is given by the cumulative Binomial distribution:

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

An illustration of acceptance probabilities as a function of the ground truth position is shown in Figure 1. We can see that agents that are well inside top k are almost certain to be accepted while those well outside of top k are almost certain to be rejected. The width of the interval around 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 $\mathbb{P}[\text{accept} | R = r]$ and contributes 1 to the size if they are accepted, the expectation is simply $\sum_{r=1}^{n} \mathbb{P}[\text{accept} | R = r]$. The complexity of this expression makes it difficult to analyse it explicitly. However, Figure 2a shows a typical behaviour of the expected size as a function of m. We observe that this approaches k as m increases. However, for small values of m the 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 ε that allows us to control the size of the accepting set more finely. If ε is set to a non-zero value (usually a positive one), we extend the nomination quota in each review pool 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 $k = 30$, Figure 2a 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 $\varepsilon \in [-0.05, 0.15]$, 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 [Kurokawa *et al.*, 2015],



(a) Expected Size (b) Expected Accuracy

Figure 2: (a) Expected size of the accepting set returned by the algorithm when $n = 130, k = 30$ and varying m. (b) Expected accuracy and accepting size for different values of k. $n = 130$, $m = 9$ and $\varepsilon = 0.15$ were used for this figure. The red line shows the expected accepting size and the blue line shows the accuracy.

while the Dollar Partition method may return as many additional agents as the number of clusters [Aziz *et al.*, 2016].

Recall that the above analysis assumes reviewers to be accurate. If this assumption fails, we cannot provide any guarantees even for the expected size of the accepting set. It is also easy to construct marginal cases in which everyone or no one is selected in the worst case scenario.

Example 1. 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.

In Section 4 we consider a realistic setting that includes a noise model for the reviews and discuss the accepting size and the performance of PEERNOMINATION.

3.3 Expected Size and Accuracy

Above we derived the probability of acceptance given a position in the ground truth, assuming no noise, before introducing the parameter ε. It is easy to adapt this expression to include ε: simply update the nomination quota when computing q_r in Equation 1. Hence, let $k_q^\varepsilon = k_q + \varepsilon$ and

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

This gives us $\mathbb{P}[\varepsilon\text{-accept}|R = r]$ for each ground truth position by simply replacing q_r in Equation 2 by q_r^ε. The expected size is again given by a similar expression:

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

It is now in principle easy to derive the expected accuracy of the algorithm. However, since the algorithm's output is inexact, there are multiple accuracy measures to consider, as is often the case for classification algorithms [Bishop, 2006]. For example, we might care about how many agents of the true top k we have selected (recall) or that we do not select too many agents from outside of it (false positive rate). We focus on the former, which we note is elsewhere referred to as *accuracy* [Aziz *et al.*, 2019]. The connection with classification metrics will be further explored in Section 4. Now, the *expected recall* is simply the sum of the probability of selection over all true top k positions, divided by k:

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

Again, the complexity of these expressions hinders theoretical analysis but Figure 2b 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 4.

3.4 Strategyproofness and Monotonicity

Our main desired property is that of impartiality or strategyproofness. Luckily, this comes almost for free since the agents are chosen independently.

Proposition 1. *The mechanism is strategyproof, i.e., no agent can affect their chances of selection using any strategy.*

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

Proposition 2. *The mechanism is monotonic, i.e., if a reviewer increases their ranking of an agent, that agent's probability of selection is not decreased.*

Proof. Suppose j is reviewed by i and consider the probability of selecting j given the original review of i, and a modified one where j is ranked higher. There are three cases:

1. j was already inside the integer part of the nomination quota in the original review or j is still completely outside of the 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.
2. 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.
3. j moves from being not nominated to be 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. □

Notice that in the definition of the algorithm we stipulate that ε is part of the input. One could be tempted to calculate ε after collecting the reviews in order to adjust the output size to be exactly k, however this is undesirable for several reasons. Firstly, the run of the algorithm is non-deterministic, hence it might be impossible to find a value of ε that guarantees such output size on every run. Secondly, and most importantly, this would eliminate strategyproofness since now an agent could estimate that reporting an untruthful review could decrease the size of the accepting set, hence forcing the mechanism to increase ε and so increase their chances of selection.

4 Simulation Experiments

We draw a novel connection between inexact peer selection and the literature on classification in machine learning [Bishop, 2006]. With this empirical framework we run experiments to demonstrate that PEERNOMINATION outperforms other mechanisms proposed.

4.1 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 in [Aziz *et al.*, 2019]. However, comparison with inexact mechanisms is less obvious. Since the accepting set is not guaranteed to be of size k exactly, any output with more than k agents may artificially increase the accuracy of the inexact mechanism and the opposite for any smaller output. One option is to measure the accuracy 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 accuracy measures [Bishop, 2006] 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: TP $= |S^+|$, FP $= |S^-| = |S| -$ TP, true negatives TN $= |\{r \notin S \mid \text{rank}(r) > k\}| = n - k -$ FP, and false negatives FN $= |\{r \notin S \mid \text{rank}(r) \leq k\}| = n - |S| -$ TN.

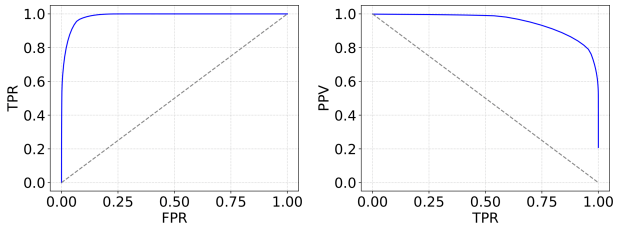
We can now look at some of the normal performance metrics: Positive Predictive Value (PPV) (aka Precision), True Positive Rate (TPR) (aka Recall) and False Positive Rate (FPR), defined as follows:

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

Furthermore, we can view the slack parameter ε as the sensitivity threshold akin to the probability threshold in the machine learning literature (see e.g., [Flach, 2012]).

This suggests a method to construct the Precision-Recall (PR) and Receiver-Operator Characteristic Curve (ROC): vary ε 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 Figure 3.

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, i.e., we do not need to accept *too many* extra agents to select all the deserving agents. 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 agents in the true top k if we concentrate on not selecting the "undeserving" individuals. 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.



(a) ROC Curve (b) PR Curve

Figure 3: ROC and PR curves for PEERNOMINATION. They were computed analytically with $n = 120, m = 8, k = 25$.

4.2 Experimental Setup

We extend the testing framework developed by Aziz *et al.* [2019] and using methods from PREFLIB [Mattei and Walsh, 2017]. Our code and data is available online [1]. As in Aziz *et al.* [2019], 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 $5, 7, 9, 11$. For the algorithms that rely on the partition, we chose the number of partitions, l, to be 4.

For each setting of the parameters we generated a random m-regular assignment matching reviewers to reviewees. As in other works, we model he reviews of each agent using a Mallows Model [Mallows, 1957]. In a Mallows model we provide a (random) ground truth ranking π and a noise parameter ϕ. If we set $\phi = 0$ then agents will always report π as their ranking, i.e., they are all exactly correct. As we increase ϕ agents will report increasingly inaccurate rankings as a function of the Kendall tau distance between π and all possible rankings. Note that each agent draws from this distribution independently. Hence, by varying ϕ we can test the robustness of our algorithms to errors in the rankings submitted by the agents. Mallows models have a long history in machine learning and group decision-making as they can simulate noisy observations of a ground truth ranking, and be sampled efficiently [Xia, 2019].

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 ε to achieve the right expected size of the accepting set. The error bars in Figures 4 and 5 represent 1 standard deviation of the data. In line with the observation in [Aziz *et al.*, 2019], varying the dispersion parameter ϕ for Mallows' noise did not have significant effect on the accuracy of all algorithms until we approach $\phi = 1.0$ when all reviewers report a completely random ordering. The results for $\phi = 0.5$ are presented in Figure 4.

PEERNOMINATION does not require an explicit partitioning making it more flexible. Another issue with partitioning, as pointed out in [Aziz *et al.*, 2019], is that the performance of both EDP and Partition degrade as we increase the number

[1] https://github.com/nmattei/peerselection

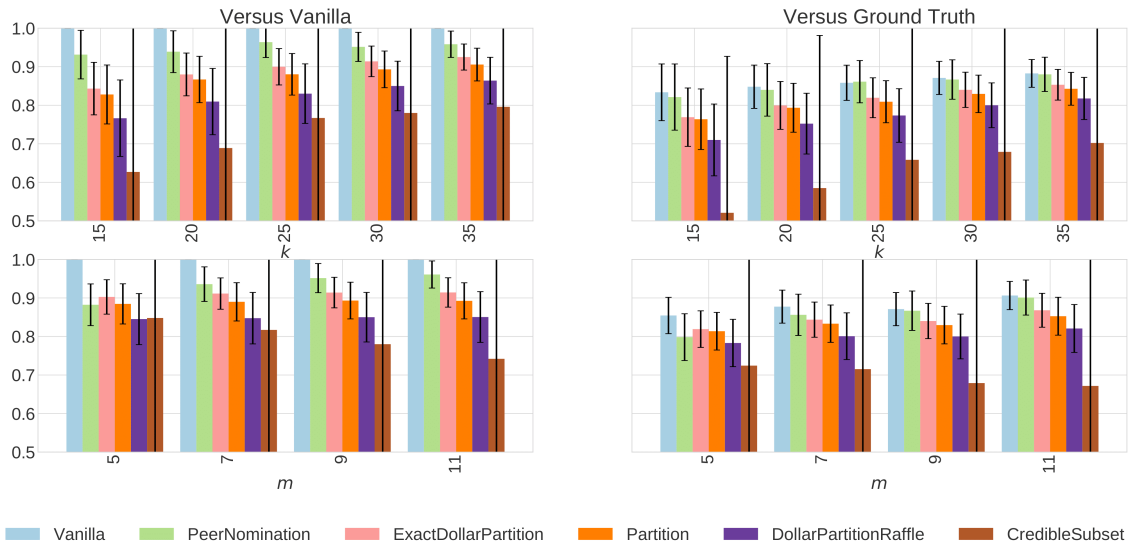


Figure 4: Comparison of prominent algorithms against a Vanilla baseline (left) and against the ground truth ranking of a Mallows Model (right). $n = 120, l = 4, \varphi = 0.5$. On top $m = 9$. On the bottom $k = 30$. PEERNOMINATION out performs across settings except $m = 5$.

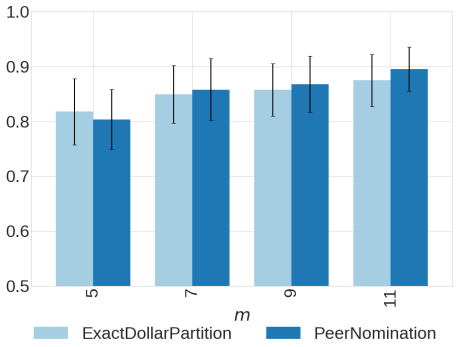


Figure 5: Results of the forced size experiment: PEERNOMINATION and EDP are always guaranteed to return the same number of agents.

of clusters. In another test we varied the number of clusters ℓ between 2 and 10. We saw a decrease in performance of about $3-4\%$ for the partition based methods while the performance of PEERNOMINATION remained constant.

In another testing setup we adopted a slightly different procedure in order to ensure a level comparison. In each simulation, we generate a random m-regular assignment, run PEERNOMINATION using the target k as an input, measure the size of the output and run EDP using this size as the input k. A similar experiment was also performed for the inexact version of Dollar Partition in Aziz *et al.* [2016]. This ensures that during each simulation both algorithms return the same number of agents for selection. The results of this comparison are presented in Figure 5.

4.3 Results

Again following Aziz *et al.* [2019] and depicted in Figure 4, we compared a selection of impartial peer selection algorithms and Vanilla (Borda count). Borda is a classic social choice rule that is known not to be strategyproof but is optimal in the ordinal peer-ranking setting under the assumption of no noise [Caragiannis *et al.*, 2016], and thus represents an optimistic baseline in the presence of no manipulators. PEERNOMINATION outperforms EDP significantly in the the majority of the settings we have considered. The only setting where EDP outperforms PEERNOMINATION is at $m = 5$, which is a low information setting, where reviewers are given a nomination quota fractionally above 1. However, our algorithm improves quickly with m. Even at $m = 9$ shown in Figure 4 PEERNOMINATION approaches the performance of Borda across the values of k we considered.

It is worth noting that PEERNOMINATION tends to return a slightly larger than k set on average (usually < 1 additional agent). Nevertheless, even if the testing forces PEERNOMINATION and EDP to return the same number of agents every time, we see that PEERNOMINATION has an overall advantage as shown in Figure 5. Again, EDP only does better in a low information setting ($m = 5$).

5 Conclusion

There are many avenues for future work: PEERNOMINATION, which already does not rely on predefined clustering, can be extended to not require an m-regular assignment. Moreover, each reviewer does not even need to declare a full ranking over their review pool, but simply declare the nominees for the selection and one nominee to be fractionally selected. This also suggests that there might be a possible extension of the algorithm which makes use of the declared rankings in full, as this data is currently discarded.

Crucially, the usefulness of our algorithm depends on returning an accepting set of size close to k. We saw that this can be achieved using the parameter ε. However, we saw that very high levels of noise can affect the size of the accepting set. This suggests that an important research direction will be that of testing different models of agent behaviour in detail.

References

[Alon *et al.*, 2011] Noga Alon, Felix Fischer, Ariel Procaccia, and Moshe Tennenholtz. Sum of us: Strategyproof selection from the selectors. In *Proceedings of the 13th Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, pages 101–110, 2011.

[Aziz *et al.*, 2016] Haris Aziz, Omer Lev, Nicholas Mattei, Jeffrey S. Rosenschein, and Toby Walsh. Strategyproof peer selection: Mechanisms, analyses, and experiments. In Dale Schuurmans and Michael P. Wellman, editors, *AAAI*, pages 397–403. AAAI Press, 2016.

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

[Berger and Müller-Hannemann, 2010] Annabell Berger and Matthias Müller-Hannemann. Uniform sampling of digraphs with a fixed degree sequence. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 220–231. Springer, 2010.

[Bishop, 2006] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[Bjelde *et al.*, 2017] Antje Bjelde, Felix Fischer, and Max Klimm. Impartial selection and the power of up to two choices. *ACM Transactions on Economics and Computation*, 5(4):1–20, 2017.

[Bousquet *et al.*, 2014] N. Bousquet, S. Norin, and A. Vetta. A near-optimal mechanism for impartial selection. In *Proceedings of the 10th International Workshop on Internet and Network Economics (WINE)*, Lecture Notes in Computer Science (LNCS), pages 133–146, 2014.

[Caragiannis *et al.*, 2016] Ioannis Caragiannis, George A. Krimpas, and Alexandros A. Voudouris. How effective can simple ordinal peer grading be? *Proceedings of the 2016 ACM Conference on Economics and Computation - EC 16*, 2016.

[de Clippel *et al.*, 2008] Geoffroy de Clippel, Hervé Moulin, and Nicolaus Tideman. Impartial division of a dollar. *Journal of Economic Theory*, 139:176–191, 2008.

[Douceur, 2009] John R. Douceur. Paper rating vs. paper ranking. *SIGOPS Oper. Syst. Rev.*, 43(2):117–121, April 2009.

[Fischer and Klimm, 2014] Felix Fischer and Max Klimm. Optimal impartial selection. In *Proceedings of the 15th ACM Conference on Economics and Computation (ACM-EC)*, pages 803–820, 2014.

[Flach, 2012] Peter A. Flach. *Machine Learning - The Art and Science of Algorithms that Make Sense of Data*. Cambridge University Press, 2012.

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

[Kahng *et al.*, 2018] Anson Kahng, Yasmine Kotturi, Chinmay Kulkarni, David Kurokawa, and Ariel Procaccia. Ranking wily people who rank each other. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[Keynes, 1936] John Maynard Keynes. *The General Theory of Employment, Interest and Money*. Palgrave Macmillan, 1936.

[Kurokawa *et al.*, 2015] David Kurokawa, Omer Lev, Jamie Morgenstern, and Ariel D. Procaccia. Impartial peer review. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 582–588. AAAI Press, 2015.

[Langford, 2015] John Langford. The NIPS experiment, Jan 2015.

[Lian *et al.*, 2018] Jing Wu Lian, Nicholas Mattei, Renee Noble, and Toby Walsh. The conference paper assignment problem: Using order weighted averages to assign indivisible goods. In *Proceedings of the32ndAAAI Conference on Artificial Intelligence (AAAI)*, pages 1138–1145, 2018.

[Mallows, 1957] Colin Lingwood Mallows. Non-null ranking models. I. *Biometrika*, 44(1-2):114–130, June 1957.

[Mattei and Walsh, 2017] Nicholas Mattei and Toby Walsh. A PREFLIB.ORG Retrospective: Lessons Learned and New Directions. In U. Endriss, editor, *Trends in Computational Social Choice*, chapter 15, pages 289–309. AI Access Foundation, 2017.

[Merrifield and Saari, 2009] Michael Merrifield and Donald Saari. Telescope time without tears: a distributed approach to peer review. *Astronomy and Geophysics*, 50(4):4.16–4.20, 2009.

[Naghizadeh and Liu, 2013] Parinaz Naghizadeh and Mingyan Liu. Incentives, quality, and risks: A look into the NSF proposal review pilot. *CoRR*, abs/1307.6528, 2013.

[Piech *et al.*, 2013] Chris Piech, Jonathan Huang, Zhenghao Chen, Chuong Do, Andrew Ng, and Daphne Koller. Tuned models of peer assessment in moocs. *arXiv preprint arXiv:1307.2579*, 2013.

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

[Wang and Shah, 2019] Jingyan Wang and Nihar B Shah. Your 2 is my 1, your 3 is my 9: Handling arbitrary miscalibrations in ratings. In *Proceedings of the18th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 864–872, 2019.

[Xia, 2019] Lirong Xia. *Learning and Decision-Making from Rank Data*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan and Claypool, January 2019.

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