of committing ranking error by swapping neighboring elements according to σ (**?**). Mallows models are used when each agent is assumed to have the same reference ranking subject to noise.

Each of the algorithms takes as input a (sparse) $n \times n$ score matrix. In most settings where peer selection is used there is a set of scores that can be given by a reviewer. This creates a set of equivalence classes of proposals that are assigned the same overall score. For example, when reviewing papers for a conference, a reviewer may assign the highest score to only a very small percentage of papers if he were to see all of the papers. We suppose that agents are able to express these equivalence classes by assigning a set number of *grades*, G. To generate our input we define two functions F and D common to all agents (this is generalizable in our testing framework) that describe the scoring and distribution, respectively. For example, using Borda scoring where F : $[4, 3, 2, 1, 0]$ and a distribution function D : $[0.2, 0.2, 0.2, 0.2, 0.2]$; all agents in the top 20% of agents receive a score of 4, the next 20% a score of 3, and so on. The functions D and F are passed as parameters to the profile generator, allowing flexibility in testing. Formally, first we generate a complete, strict relation for agent i. Given a probability density function (PDF) D for each grade $g \in G, D(g) \rightarrow R^+$ where $\sum_{g \in G} D(g) = 1.0$ and a scoring function F for each grade $g \in G, F(g) \rightarrow Z^+$.

Each agent reviews m of the n proposals and is also reviewed by m other agents. Since we are dealing with clusters, we additionally have the constraint that each agent reviews m agents *outside* his cluster. We refer to review assignments satisfying these constraints as balanced m-regular assignments. We convert a complete $n \times n$ score matrix into a sparse score matrix by drawing a balanced m-regular assignment with respect to a given clustering. In order to maximize inter-cluster comparison, we would also like that the m agents that agent i is to review are reasonably balanced among the clusters (not including i's cluster) so that each agent in each cluster C_i reviews in total $\frac{|C_i| \cdot m}{\ell - 1}$ agents from each other cluster. We generate this assignment randomly and as close to balanced as possible. Given the balanced m-regular assignment for agent i, we remove all other candidates from i's complete score vector. Hence we are left with a sparse, m-regular score matrix which respects a clustering of the agents into ℓ clusters. The resulting score matrix resembles what a conference organizer or NSF program manager sees: a sparse and noisy observation of the ground truth filtered through equivalence classes

Results for an NSF-Like Program: Using numbers from the NSF[7] we settled on a set of realistic parameters that one may see in the real world. The "Mechanism Design" pilot, which used the mechanism proposed by **?** (**?**) had 131 proposals, with each submitter reviewing 7 other proposals. The acceptance numbers are not broken out from the global acceptance rate for the program. Consequently we assume an $\approx$20% acceptance rate, the same as NSF as a whole and also similar to other conference and funding acceptance rates.

We use a "normal" distribution giving $|D| = [4, 7, 15, 20, 39, 20, 15, 7, 3]$ and a Borda scoring function that one would expect to find in most conference reviewing $F = [8, 7, 6, 5, 4, 3, 2, 1, 0]$ corresponding to the grades $G = [A+, A, B+, B, C+, C, D+, D, F]$. Without loss of generality we assume that the ground truth ordering σ is in agent order, i.e., $1, \ldots, 130$. The ground truth ordering σ gives us an indication of which agents are objectively better than the others. However, this ground truth is filtered not only through the noise of the individual agents (ϕ) but also by the inexactness of the m-regular assignment. Given D and k we can establish how many of the selections *should* come from each grade. In a competitive setting, we want to select those agents at the top of the ground truth ordering.

Figure 1 shows the performance of the six mechanisms discussed on two different metrics as we vary the number of reviews received. We fixed $\phi = 0.1$ for this testing as setting $\phi \in \{0.0, 0.1, 0.25, 0.4\}$ had no significant effect. The graphs show the mean cumulative proportion of the agents in each grade that are selected by each of the mechanisms over 1000 samples. For instance, the 1.0 score received by Vanilla for both A+ and A+|A for all settings of m mean that Vanilla always selects the 11 highest scoring agents in the ground truth ranking (σ). We use cumulative selection with respect to the ground truth ordering. This partial sum is well defined for each set of grades and clearly shows where a particular mechanism is over- or under-performing. Each mechanism was allowed to select a number of proposals equal to the number of agents returned by Dollar Partition per iteration, hence the average cumulative selection is > 1.0. Whilst Vanilla is the best in our experiment, strictly dominating all other mechanisms, it is the only non-strategyproof mechanism. In practice, agents may not report truthfully with Vanilla and so it can perform much worse. The other generalizations of Dollar are strictly dominated by Dollar Partition; our more nuanced mechanism yields a better selection.

Comparing Dollar Partition and Partition ($m = \{10, 15\}$), both mechanisms select all of the A+ grade agents on every iteration. Partition selects only $9/11$, in the worst case, of the A+|A, while Dollar Partition selects $10/11$, an 11% improvement. Considering the A+|A|B+ agents, Partition only selects $17/26$, while Dollar Partition selects $20/26$, a $\geq 17\%$ performance increase. Neither mechanism ever selects an agent with rank lower than C+; even in the worst case, both perform better than every other strategyproof mechanism in our study.[8] Standard deviation is also higher for Partition for all these cases, indicating Partition is much more likely to make mistakes and select agents from a lower grade over agents in a higher grade. Dollar Partition performs better than Partition in the worst case, and performs better on average. In a low information setting (i.e., $m \leq 5$), Partition does perform slightly better on average than Dollar Partition. However, Dollar Partition shows a lower variance and better worst case performance across all settings to m, demonstrating its robustness to lopsided clusterings.

General Results: We explored a realistic part of the large

[7] http://www.nsf.gov/nsb/publications/2015/nsb201514.pdf

[8] It is hard to directly compare results for Credible Subset due to the large probability of returning an empty set. This problem is not easily overcome; removing the ability to return an empty set means Credible Subset is no longer strategyproof. When Credible Subset does return a set, it slightly outperforms other mechanisms.

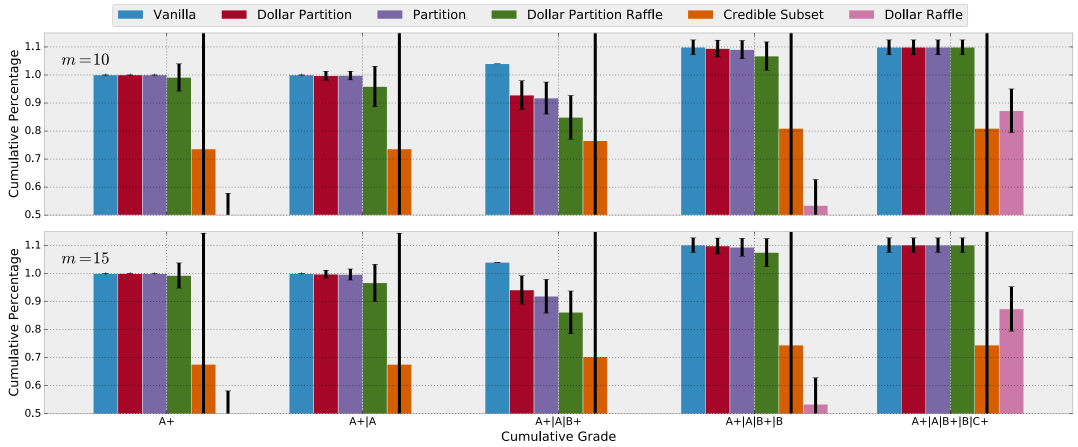

Figure 1: Mean cumulative percentage of each grade of agent selected by the six peer selection algorithms presented in this paper on 1000 random iterations selecting $k = 25$ agents from a population of $n = 130$ agents providing $m = 10$ (top) and $m = 15$ (bottom) reviews divided into $l = 5$ clusters with a Mallows dispersion $\phi = 0.1$. To enable comparisons, every mechanism selects $|W|$ equal to that of Dollar Partition; hence the ≥ 1.0 averages as $k = 25$ is the denominator. Error bars represent one standard deviation from the mean. Dollar Partition selects more agents from a higher grade more often, selects more agents from a higher grade in the worst case, and does so more consistently, than any other strategyproof mechanism. To highlight Partition and Dollar Partition we have cropped results where they are the same (cutting off Dollar Raffle).

parameter space to investigate the mechanisms. The practical upshot, after running hundreds of thousands of instances, is that there are numerous tradeoffs that system designers must consider, critically depending on their target domain. In general, varying other parameters, such as k, ℓ, D and F did not change the ranking of mechanisms shown here. However, increasing the number of clusters improved Dollar Partition's performance in comparison to Partition's, which may stem from the increased chance that Partition will select the bottom candidates of a given cluster instead of better ranked candidates in a different cluster. Accordingly, as it generally selects the top candidates, Partition's performance improves when scoring rules are exponential in comparison to less extreme scoring rules, such as Borda. Dollar Partition is much better when there is sufficient information, in terms of the number of reviews and the granularity of the grades, to have a chance of recovering the ground truth ordering. Settings like conferences with $n = 2000$ papers and $m = 5$ reviews split into 5–8 grades often have no clear cutoff between accept and reject; the grades contain too many items. In these cases all the mechanisms perform poorly, as selecting a set of winners is akin to randomly selecting agents from the set of possible winners. See, e.g., the NIPS experiment[9] and the recent paper on the limits of noisy rank aggregation using data from the KDD conference (**?**). As the ratio of m to n grows, and the granularity of the grades increases, it becomes possible to recover the ground truth ranking, and Dollar Partition outperforms the other mechanisms.

6 Conclusion

We introduce a novel peer selection mechanism—Dollar Partition. Overall, Dollar Partition's flexibility in setting the number of agents to be selected from each cluster addresses the worst-case instances where partitions may be lopsided, allowing Dollar Partition to reach higher quality, more consistent results than existing mechanisms. Combined with the ability to always return a winning set, it is an improvement over current mechanisms. Among strategyproof mechanisms, Partition and Dollar Partition may have a certain 'psychological' advantage: they may incentivize agents to report truthfully because an agent's contribution in selecting other agents (with whom he is not competing) is more direct. Moreover, partitioning into groups helps deal with conflict of interest cases, when there is fear of collusion among several agents; putting them in the same cluster prevents them from influencing one another's chance of success. Peer selection is a fundamental problem that has received less attention than voting rules. We envisage the need to develop robust solutions with good incentive properties, as these are widely applicable in large-scale, crowdsourcing settings.

Acknowledgments

Data61 (formerly known as NICTA) is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program. This research has also been partly funded by Microsoft Research through its PhD Scholarship Program, and by Israel Science Foundation grant #1227/12. This work has also been partly supported by COST Action IC1205 on Computational Social Choice.

Aperiam hic odit architecto nisi cumque unde libero, quia suscipit eos quasi unde autem deleniti ea placeat quas ex voluptate, qui sed ex odit repellendus iste fuga voluptas ea laborum voluptates pariatur?

[9] http://blog.mrtz.org/2014/12/15/the-nips-experiment.html