the power of combining greedy and randomized steps (indeed, if one were to use our algorithms to compute the result of a political election, as in the indirect approach from the introduction, one would likely try as many various combinations of the parameters of the algorithms as time would permit).

3.3 Putting the Algorithms Together

Altogether, in our experimental section we consider six algorithms. The first three are, simply, the initialization algorithms G1, G5, and R(1,5) that we have described above. The next three, algorithms G1C, G5C, and R(1,5)C, first use the respective initialization algorithm to find a set of committees (we let the algorithms output all the committees they construct, without picking the best one), then we apply the clustering heuristic to each of them, and finally output the committee with the highest voter satisfaction. The algorithms with the clustering step are bound to be better than those without it. It is interesting—and we evaluate it in the next section—how often and to what extent the clustering step improves the solution.

For algorithms R(1,5) and R(1,5)C, we use the following parameters. We set $p = 0.75$ and $e = 1$ (e.g., in the first iteration, D1 has 0.75 probability of executing a randomized step, in the second iteration ≈ 0.56, in the third iteration ≈ 0.42, and so on).

4. EXPERIMENTAL ANALYSIS

We now present our experimental results. We have three main goals. First, we would like to evaluate how well our algorithms perform (that is, how close are the solutions they produce to optimal outcomes). Second, we would like to evaluate how effective is the randomization in algorithms R(1,5) and R(1,5)C, as compared to the other, fully deterministic, algorithms. Third, we would like to evaluate how effective the clustering step is.

4.1 Experimental Setup

We have run our algorithms in a number of experiments. For each experiment we have generated 500 instances of elections (we used the Polya-Eggenberger urn model; see below), each with $m = 100$ candidates and $n = 100$ voters. For each experiment we used three satisfaction functions: γ_{Borda}, γ_{convex}, and $\gamma_{\mathrm{concave}}$.

Function γ_{Borda} is the Borda satisfaction function, i.e., $\gamma_{\mathrm{Borda}}(i) = m - i$, for $i \in [m]$. Functions γ_{convex} and $\gamma_{\mathrm{concave}}$ are certain convex and concave functions (respectively) and are depicted in Figure 1. Formally, they are defined as:

$$\gamma_{\mathrm{concave}}(i) = m\left(\frac{-m+i+1}{m-1}\right)^7 + m\ ,\ \gamma_{\mathrm{convex}}(i) = m\left(\frac{m-i-1}{m-1}\right)^7.$$

In most cases we have considered relatively small committees ($k = 3$ and $k = 5$), but we have also considered several much larger committees ($k = 27$ and $k = 47$). For each experiment, we run our six algorithms, and additionally we computed the optimal solution using an ILP solver (see the works of Lu and Boutilier [9] and Skowron et al. [18] for ILP formulations of the winner determination problems).[6]

[6]This is why we chose elections with 100 candidates and 100 voters: They are small enough to compute optimal solutions using ILP, but large enough to have some nontrivial structure.

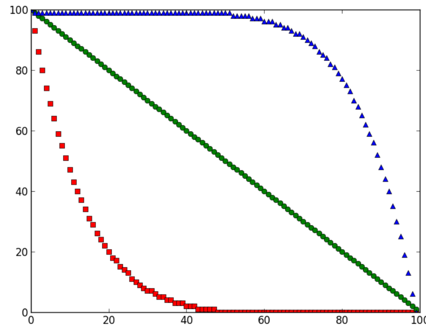

Figure 1: Our satisfaction functions: γ_{convex} (red rectangles; bottom), γ_{Borda} (green circles; center), and $\gamma_{\mathrm{concave}}$ (blue triangles; top).

We generated the elections using the Polya–Eggenberger urn model (see the works of Skowron et al. [18], Berg and Lepelley [1], McCabe-Dansted and Slinko [12], Walsh [21], and Erdelyi et al. [7] for some other examples where this model was used). This model operates as follows: For a candidate set C, we create an urn which initially contains one copy of each of the $|C|!$ possible preference orders over C. To generate a vote, we draw a preference order randomly from the urn and include it in the election. Then we return this order back to the urn, together with $\alpha \cdot |C|!$ additional copies, where α is a parameter of contagion (for $\alpha = 0$ we would get the impartial culture model, where votes are drawn uniformly at random).

The sheer number of experiments that we have run is too large for presenting all the results which we have obtained. Thus, in the discussion below, we present only some representative results (we show results for $k = 3$, $\alpha \in \{0.05, 0.25\}$, for both Monroe and CC, for all our satisfaction functions; we also show three other interesting experiments).

4.2 Discussion of the Results

We present the results of our experiments in Figures 2–16. For each of the plots we indicate the voting rule (Chamberlin–Courant or Monroe), the satisfaction function ($\gamma_{\mathrm{concave}}$, γ_{Borda}, γ_{convex}), the size of the committee (3, 5, 27, 47), and the value of the parameter α for the urn model (0.05 or 0.25). The plots show the distribution of the voter satisfaction achieved by each of our algorithms (averaged over 500 elections generated in each given experiment), scaled so that the optimal satisfaction is 1. The red line (in the middle of each rectangle) indicates the median quality of the obtained solutions. The blue rectangle covers solutions from the 25th to 75th percentile (i.e., it shows the range of the satisfaction values of solutions worse than the best quarter of the obtained solutions, but better than the worst quarter of the solutions). The top and bottom black lines indicate 1.5 times the interquartile range, and smaller blue crosses below and above them indicate outliers.

In each of our plots, each column stands for one of our six algorithms. The number next to G1C (G5C, R(1,5)C) is the number of instances for which G1C (G5C, R(1,5)C) performed better than G1 (G5, R(1,5)). This number shows how frequently the clustering step provides an improvement.

How well do our algorithms perform? All our algorithms achieve very good approximation ratios, though there are noticeable differences in their quality. Typically, G1 performs worst of all (but still, the median result tends to be at least 0.95 of the voter satisfaction in the optimal solution) and R(1,5)C tends to perform best.

How effective are the randomization steps? It is quite evident that introducing the randomization steps has very positive impact on the quality of the generated solutions. If we compare the results for R(1,5) and R(1,5)C to the results for G5 and G5C, respectively, then the randomized algorithms never perform noticeably worse, and often achieve slightly better results (but see the next point).

How effective is the clustering step? It turns out that in most scenarios applying the clustering step improves the solution by, at least, some amount, and in some scenarios it improves it greatly. This is most evident when we compare the results of algorithms R(1,5) and R(1,5)C for the Monroe family of rules. In this case, applying the clustering step often improves (quite significantly, judging by the improvement of the median satisfaction of the voters) the solution in many more than half of the instances. On the other hand, if we compare the results of G1 and G1C, or of G5 and G5C, then the clustering step does help, but on fewer instances and to a lesser extent. This confirms that generating an appropriate initial committee is very important.

Contrasting the figures corresponding to $\alpha = 0.05$ to the figures corresponding to $\alpha = 0.25$, we see that with the increase of the coefficient of contagion (i.e., the homogeneity of the population) the clustering step is becoming noticeably more useful. The reason for this is not exactly clear but, intuitively, with the increase of contagion the clusters in the data become more pronounced, hence the clustering step becomes more useful.

5. SUMMARY

We have considered the problem of computing approximate solutions for the Monroe and Chamberlin–Courant rules via algorithms combining the existing greedy approximation algorithms, randomized steps, and clustering. We have shown that putting these ideas together leads to noticeably more effective algorithms than previously known from the literature. In particular, our results have shown that interleaving greedy steps with randomized steps leads to committees that are very good starting points for the clustering step. Our algorithms can quite easily be modified to satisfy soft constraints such as "try to find as good a committee as possible with as many members of a given party as possible."

For future work, it would be interesting to perform similar experiments as ours on real election data (e.g., from PrefLib [11]). Second, it would be interesting to find a principled way of setting the parameters of our procedures (we relied on ad hoc settings). Third, we are interested in theoretical explanations of why interleaving greedy and random step gives better outcomes than greedy steps alone.

Acknowledgments

We are grateful to the reviewers for their comments. Piotr Faliszewski was supported by NCN grant DEC-2012/06/M/ST1/00358. Arkadii Slinko was supported by the Royal Society of NZ Marsden Fund UOA-254. Kolja Stahl was supported by DFG project PAWS (NI 369/10). Nimrod Talmon was supported by the DFG Research Training Group MDS (GRK 1408).

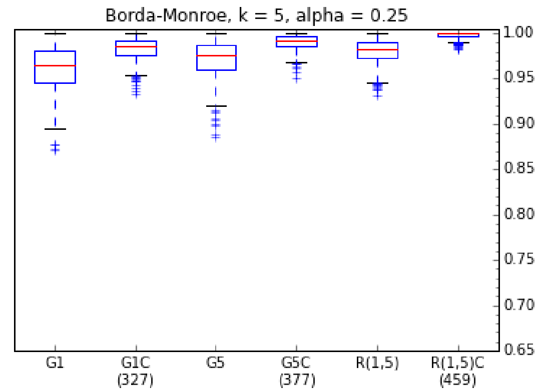

Figure 2: Performance of our algorithms for the Monroe voting rule, committee size = 5, Borda scores, and the urn model with $\alpha = 0.25$.

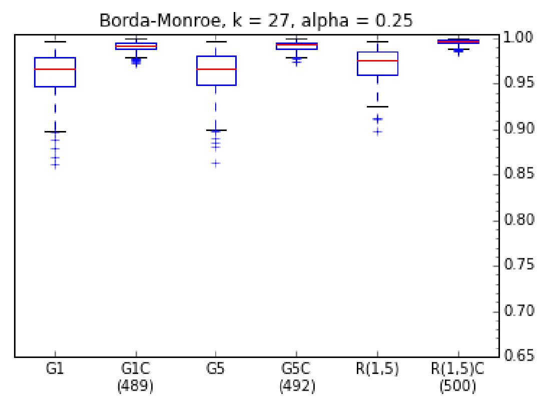

Figure 3: Performance of our algorithms for the Monroe voting rule, committee size = 27, Borda scores, and the urn model with $\alpha = 0.25$.

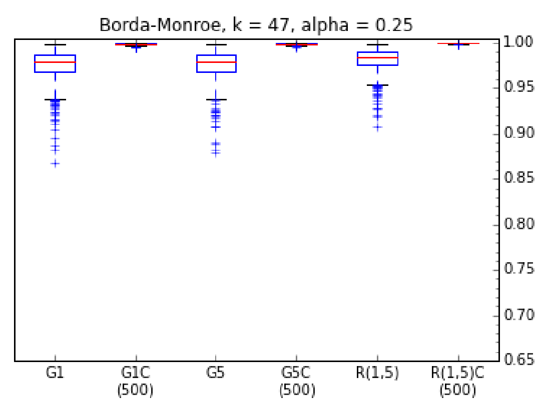

Figure 4: Performance of our algorithms for the Monroe voting rule, committee size = 47, Borda scores, and the urn model with $\alpha = 0.25$.

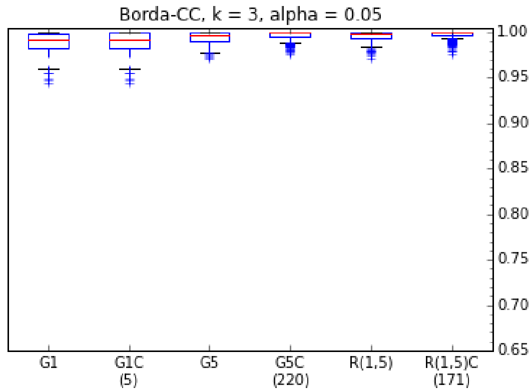

Figure 5: Performance of our algorithms for the Chamberlin–Courant voting rule, committee size = 3, Borda scores, and the urn model with $\alpha = 0.05$.

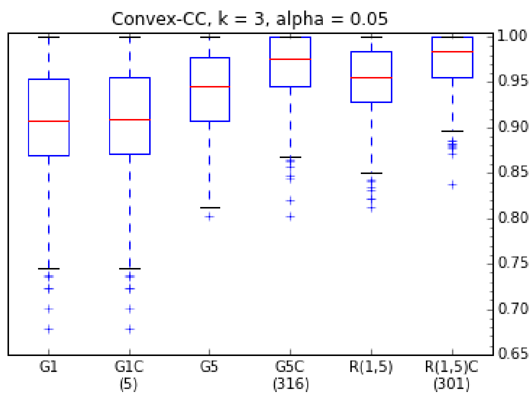

Figure 6: Performance of our algorithms for the Chamberlin–Courant voting rule, committee size = 3, Convex scores, and the urn model with $\alpha = 0.05$.

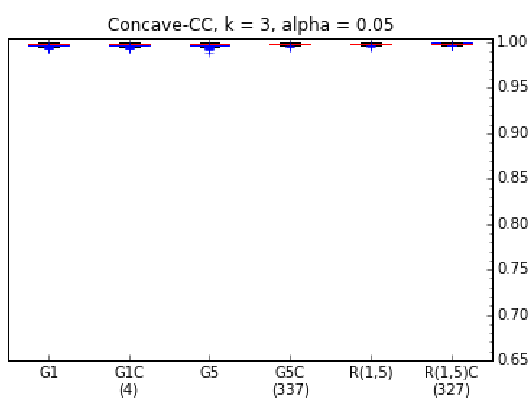

Figure 7: Performance of our algorithms for the Chamberlin–Courant voting rule, committee size = 3, Concave scores, and the urn model with $\alpha = 0.05$.

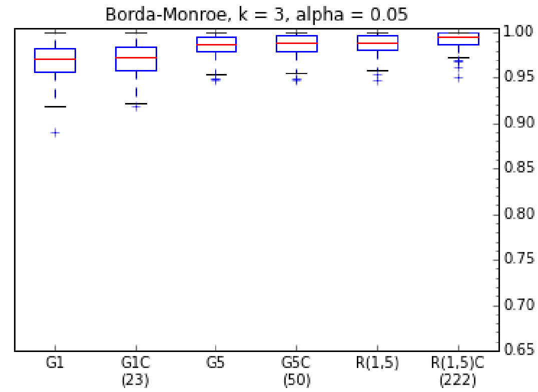

Figure 8: Performance of our algorithms for the Monroe voting rule, committee size = 3, Borda scores, and the urn model with $\alpha = 0.05$.

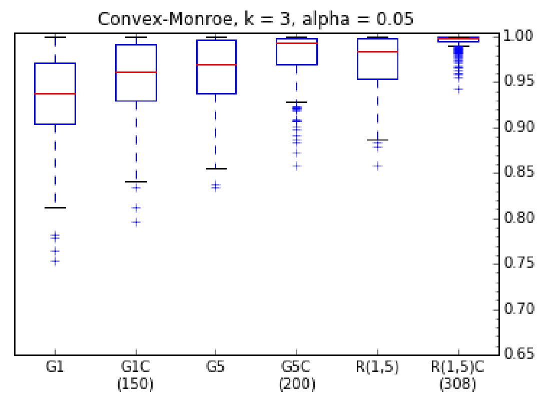

Figure 9: Performance of our algorithms for the Monroe voting rule, committee size = 3, Convex scores, and the urn model with $\alpha = 0.05$.

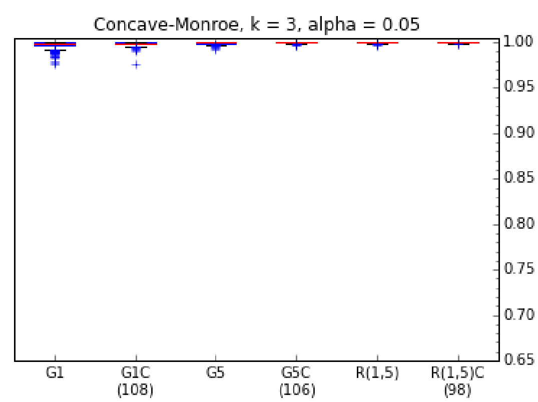

Figure 10: Performance of our algorithms for the Monroe voting rule, committee size = 3, Concave scores, and the urn model with $\alpha = 0.05$.

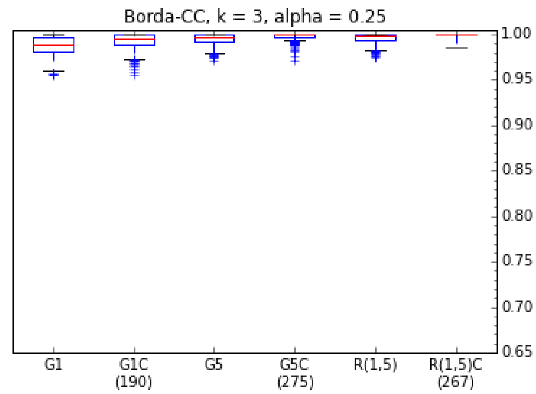

Figure 11: Performance of our algorithms for the Chamberlin–Courant voting rule, committee size = 3, Borda scores, and the urn model with $\alpha = 0.25$.

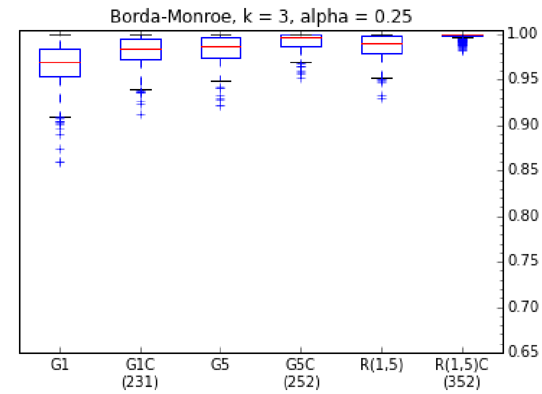

Figure 14: Performance of our algorithms for the Monroe voting rule, committee size = 3, Borda scores, and the urn model with $\alpha = 0.25$.

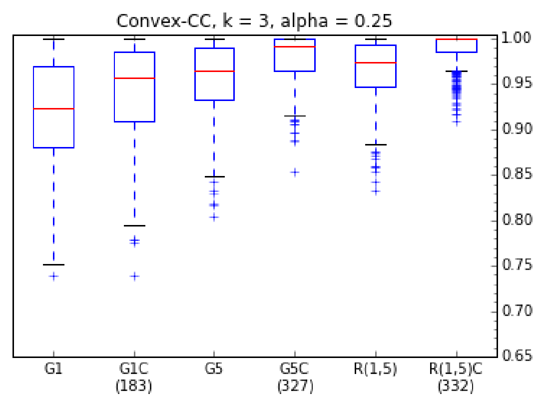

Figure 12: Performance of our algorithms for the Chamberlin–Courant voting rule, committee size = 3, Convex scores, and the urn model with $\alpha = 0.25$.

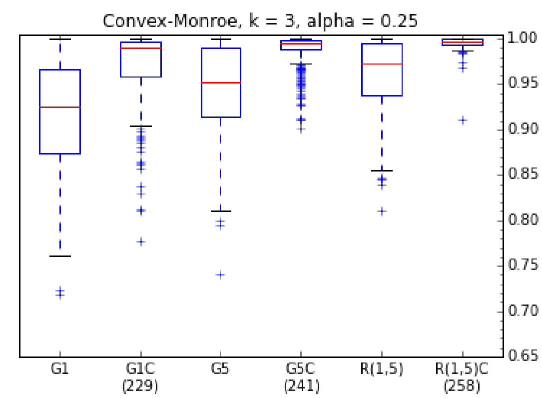

Figure 15: Performance of our algorithms for the Monroe voting rule, committee size = 3, Convex scores, and the urn model with $\alpha = 0.25$.

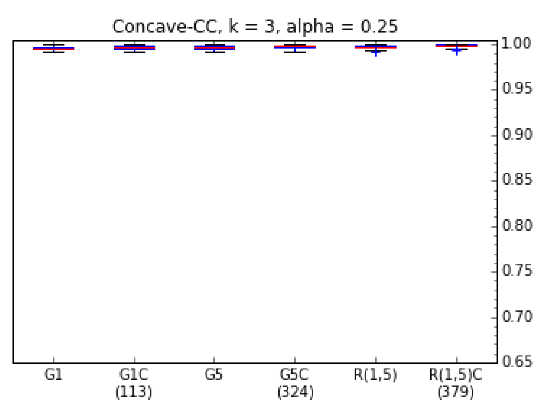

Figure 13: Performance of our algorithms for the Chamberlin–Courant voting rule, committee size = 3, Concave scores, and the urn model with $\alpha = 0.25$.

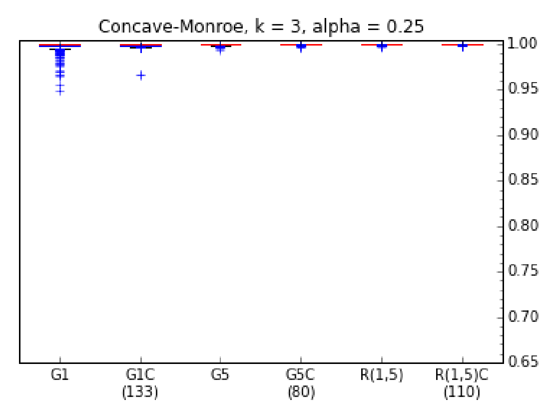

Figure 16: Performance of our algorithms for the Monroe voting rule, committee size = 3, Concave scores, and the urn model with $\alpha = 0.25$.

REFERENCES

[1] S. Berg and D. Lepelley. On Probability Models in Voting Theory. *Statistica Neerlandica*, 48(2):133–146, 1994.

[2] N. Betzler, A. Slinko, and J. Uhlmann. On the computation of fully proportional representation. *Journal of Artificial Intelligence Research*, 47:475–519, 2013.

[3] I. Caragiannis, C. Kaklamanis, N. Karanikolas, and A. Procaccia. Socially desirable approximations for Dodgson's voting rule. In *Proceedings of the 11th ACM Conference on Electronic Commerce*, pages 253–262. ACM Press, June 2010.

[4] B. Chamberlin and P. Courant. Representative deliberations and representative decisions: Proportional representation and the Borda rule. *American Political Science Review*, 77(3):718–733, 1983.

[5] R. O. Duda, P. E. Hart, et al. *Pattern classification and scene analysis*, volume 3. Wiley New York, 1973.

[6] E. Elkind, P. Faliszewski, P. Skowron, and A. Slinko. Properties of multiwinner voting rules. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems*, pages 53–60, May 2014.

[7] G. Erdélyi, M. R. Fellows, J. Rothe, and L. Schend. Control complexity in Bucklin and fallback voting: An experimental analysis. *J. Comput. Syst. Sci.*, 81(4):661–670, 2015.

[8] J. A. Hartigan. *Clustering algorithms*. John Wiley & Sons, Inc., 1975.

[9] T. Lu and C. Boutilier. Budgeted social choice: From consensus to personalized decision making. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 280–286, 2011.

[10] T. Lu and C. Boutilier. Value-directed compression of large-scale assignment problems. In *Proceedings of the 29th AAAI Conference on Artificial Intelligecne*, pages 1182–1190, 2015.

[11] N. Mattei and T. Walsh. Preflib: A library for preferences. In *Proceedings of the 3nd International Conference on Algorithmic Decision Theory*, pages 259–270, 2013.

[12] J. C. McCabe-Dansted and A. Slinko. Exploratory analysis of similarities between social choice rules. *Group Decision and Negotiation*, 15(1):77–107, 2006.

[13] B. Monroe. Fully proportional representation. *American Political Science Review*, 89(4):925–940, 1995.

[14] J. Oren and B. Lucier. Online (budgeted) social choice. In *Proceedings of the 28th AAAI Conference on Artificial Intelligecne*, pages 1456–1462, July 2014.

[15] A. Procaccia, J. Rosenschein, and A. Zohar. On the complexity of achieving proportional representation. *Social Choice and Welfare*, 30(3):353–362, 2008.

[16] P. Skowron and P. Faliszewski. Fully proportional representation with approval ballots: Approximating the MaxCover problem with bounded frequencies in FPT time. In *Proceedings of the 29th AAAI Conference on Artificial Intelligecne*, pages 2124–2130, 2015.

[17] P. Skowron, P. Faliszewski, and J. Lang. Finding a collective set of items: From proportional multirepresentation to group recommendation. In *Proceedings of the 29th AAAI Conference on Artificial Intelligecne*, 2015.

[18] P. Skowron, P. Faliszewski, and A. Slinko. Achieving fully proportional representation: Approximability result. *Artificial Intelligence*, 222:67–103, 2015.

[19] P. Skowron, L. Yu, P. Faliszewski, and E. Elkind. The complexity of fully proportional representation for single-crossing electorates. *Theoretical Computer Science*, 569:43–57, 2015.

[20] S. Thompson. *Sampling*. Wiley, 2012.

[21] T. Walsh. Where are the hard manipulation problems? *Journal of Artificial Intelligence Research*, pages 1–29, 2011.

[22] L. Yu, H. Chan, and E. Elkind. Multiwinner elections under preferences that are single-peaked on a tree. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pages 425–431, 2013.