Artificial Intelligence 222 (2015) 67–103

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Achieving fully proportional representation: Approximability results [☆]

Piotr Skowron [a,*], Piotr Faliszewski [b], Arkadii Slinko [c]

[a] *University of Warsaw, Warsaw, Poland*
[b] *AGH University, Krakow, Poland*
[c] *University of Auckland, Auckland, New Zealand*

A R T I C L E   I N F O

Article history:
Received 17 December 2013
Received in revised form 5 January 2015
Accepted 12 January 2015
Available online 30 January 2015

Keywords:
Algorithms
Parliamentary elections
Winner-determination

A B S T R A C T

We study the complexity of (approximate) winner determination under the Monroe and Chamberlin–Courant multiwinner voting rules, which determine the set of representatives by optimizing the total satisfaction or dissatisfaction of the voters with their representatives. The total (dis)satisfaction is calculated either as the sum of individual (dis)satisfactions in the utilitarian case or as the (dis)satisfaction of the worst off voter in the egalitarian case. We provide good approximation algorithms for the satisfaction-based utilitarian versions of the Monroe and Chamberlin–Courant rules, and inapproximability results for the dissatisfaction-based utilitarian versions of these rules and also for all egalitarian cases. Our algorithms are applicable and particularly appealing when voters submit truncated ballots. We provide experimental evaluation of the algorithms both on real-life preference-aggregation data and on synthetic preference data. These experiments show that our simple and fast algorithms can, in many cases, find near-perfect solutions.

1. Introduction

We study the complexity of (approximate) winner determination under the Monroe [40] and Chamberlin–Courant [13] multiwinner voting rules, both of which aim to select a group of candidates that best represent the voters. Multiwinner elections are important both for human societies (e.g., in indirect democracies for electing committees of representatives such as parliaments) and for software multiagent systems (e.g., for recommendation systems [32,41]), and thus it is important to have good multiwinner rules and good algorithms for them. The Monroe and Chamberlin–Courant rules are particularly appealing because they create an explicit (and, in some sense, optimal) connection between the elected committee members and the voters; each voter knows her representative and each committee member knows to whom she is accountable.

We assume that m candidates participate in the election and that the society consists of n voters, each of whom rank the candidates, expressing their preferences about who they would like to see as their representative. When choosing a K-member committee, the Monroe and Chamberlin–Courant rules work as follows. For each voter these rules assign a single candidate as their representative, respecting the following constraints:

[☆] This paper combines and extends results presented at IJCAI-2013 (paper titled “Fully Proportional Representation as Resource Allocation: Approximability Results”; the paper contained most of our theoretical results) and at AAMAS-2012 (paper titled “Achieving Fully Proportional Representation is Easy in Practice”; the paper contained most of our experimental results).

[*] Corresponding author.
E-mail address: p.skowron@mimuw.edu.pl (P. Skowron).

http://dx.doi.org/10.1016/j.artint.2015.01.003
0004-3702/© 2015 Elsevier B.V. All rights reserved.

(a) For the Monroe rule exactly K candidates are assigned to the voters and each selected candidate is assigned to about $\frac{n}{K}$ voters (see the definitions in Section 2 for details); for the Chamberlin–Courant rule the number of selected candidates is at most K and there is no restriction on the number of voters each committee member might represent. (The elected committee should take this into account in its operation, e.g., by means of weighted voting.)
(b) The candidates are selected and assigned to the voters optimally, either by minimizing the total (societal) dissatisfaction or by maximizing the total (societal) satisfaction (both views are equivalent in terms of optimality of solutions, but—as we will show—are different from the approximability perspective).

The total satisfaction is calculated on the basis of individual satisfactions of the agents. We assume that there is a function $\alpha : \mathbb{N} \to \mathbb{N}$ such that $\alpha(i)$ measures how well a voter is represented by the candidate that this voter ranks as i-th best. This function is the same for each voter. Naturally, it is a decreasing function; a voter is more satisfied with the candidates that she ranks higher. For example, it is typical to use the Borda count scoring function whose satisfaction in case of m candidates is $\alpha^{m}_{\mathrm{B,dec}}(i) = m - i$. The dissatisfaction of a voter is calculated analogously, except that we assume that α is an increasing function (a voter is more dissatisfied with the candidates that she ranks lower). The dissatisfaction variant of the Borda count scoring function is $\alpha^{m}_{\mathrm{B,inc}}(i) = i - 1$.

In the utilitarian variants of the rules, the assignment should maximize the total satisfaction (minimize the total dissatisfaction) calculated as the sum of the voters' individual satisfactions (dissatisfactions) with their representatives. In the egalitarian variants, the assignment should maximize the satisfaction (minimize the dissatisfaction) of the worst-off voter.

The Monroe and Chamberlin–Courant rules create a useful connection between the voters and their representatives that makes it possible to achieve both the accountability of candidates to the voters, and proportional representation of the views of the voters. Among common multi-winner voting rules, the Monroe and Chamberlin–Courant rules are quite special in having *both* the aforementioned accountability and proportionality properties simultaneously. For example, First Past the Post system (where the voters are partitioned into districts with a separate single-winner Plurality election in each) can give very disproportionate results (forcing some of the voters to be represented by candidates they dislike). On the other side of the spectrum are the party-list systems, which achieve perfect proportionality. In those systems the voters vote for the parties, and, based on the numbers, each party receives its quota of seats in the parliament, which is filled from a publicly available ordered list of candidates of this party. This process makes the elected candidates feel more accountable to the apparatchiks of their parties than to the voters. We point the readers to the recent book of Pukelsheim [49] for a detailed discussion of such issues.

Unfortunately, the Monroe and Chamberlin–Courant rules have one crucial drawback that makes them impractical. It is NP-hard to tell who the winners are! Specifically, NP-hardness of winner determination under the Monroe and Chamberlin–Courant rules was shown by Procaccia et al. [48] and by Lu and Boutilier [32]. Further, Betzler et al. [9] have studied parameterized complexity of winner determination for these rules and, in particular, have shown W[2]-hardness for winner determination parameterized by the size K of the desired committee. Nonetheless, they have also shown XP algorithms for this parameter (an XP algorithm runs in polynomial time provided the parameter is treated as a constant; thus, in other words, Betzler et al. [9] have given a separate polynomial-time algorithm for each fixed size of the committee). It is also possible to compute Chamberlin–Courant winners provided that the voters have either single-peaked preferences [9] or single-crossing preferences [58]. (For the case of the Monroe rule, single-crossing preferences do not help and winner determination remains NP-hard in this case [58]; the complexity for the case of single-peaked preferences remains open.)

Lu and Boutilier [32] proposed the use of approximation algorithms as a mean of dealing with hardness of the Chamberlin–Courant rule, and have given the first such algorithm for it. Their procedure outputs an assignment that achieves no less than $1 - \frac{1}{e} \approx 0.63$ fraction of the optimal voter satisfaction. However, the approximation ratio 0.63 here means that it is possible that, on average, each agent is represented by a candidate that this agent prefers to only about 63% of other candidates, even if there is a perfect solution that assigns each agent to her most preferred candidate. Such issues, however, would not occur if we had a constant-factor approximation algorithm minimizing the total dissatisfaction. Indeed, if a perfect solution exists, then the optimal dissatisfaction is zero and a constant-factor approximation algorithm must also output this perfect solution.

The use of approximation algorithms in real-life applications requires some discussion. For example, their use is naturally justified in the context of recommendation systems. Here the strive for optimality is not crucial since a good but not optimal recommendation still has useful information and nobody would object if we replaced the exact recommendation with an approximate one (given that the exact one is hard to calculate). For example, Amazon.com may recommend a book on gardening which may not be the best book for you on this topic, but which is still full of useful advice. For such situations, Herbert Simon [55] used the term 'satisficing,' instead of optimizing, to explain the behavior of decision makers under circumstances in which an optimal solution cannot be easily determined. On page 129 he wrote: "Evidently, organisms adapt well enough to 'satisfice'; they do not, in general, 'optimize'." The use of approximation algorithms therefore fits well with the human nature.

Still, the use of approximation algorithms in elections requires some care. It is conceivable that the electoral commission finds an allocation of voters to candidates with a certain value of (dis)satisfaction and one of the parties participating in the election finds an allocation with a better value. If the commission does not accept such an improved allocation, the society may lose faith in the fairness of the electoral process. There are several ways of handling this possibility. Firstly, an approximation algorithm can be fixed by law. In such a case, it becomes an acting voting rule and a new way to measure fairness in the society (indeed, the results of Caragiannis et al. [11] for the case of Dodgson's rule and of Elkind et al. [19], for one of our approximation algorithms, indicate that approximation algorithms may even have some more desirable axiomatic properties than the voting rules they approximate). Secondly, an electoral commission may calculate the allocation, but also publish the raw data and issue a call for submissions. If, within the period specified by law, nobody can produce a better allocation, then the committee goes ahead and announces the result. If someone produces a better allocation, then the electoral commission uses the latter one. If the electoral commission adopts this approach, then it makes sense to seek randomized approximation algorithms for our rules. While it is hard to imagine that a randomized algorithm would be fixed as an acting voting rule, a group of supporters of a party can certainly try to use it to find an allocation which is more beneficial for the party than the one found by the electoral commission.[1]

The use of approximation algorithms is even more natural in elections with partial ballots. Indeed, even an exact algorithm would produce approximate results because the voters provide us with approximate preferences.

1.1. Our results

In this paper we focus on the computational complexity of achieving fully proportional representation. Specifically, we are interested in approximation algorithms for winner determination under two fully proportional representation systems—the Monroe system and the Chamberlin–Courant system. Our main conclusion is the following:

> Achieving near-perfect proportional representation of the voters (as defined by the Monroe and Chamberlin–Courant rules) is computationally easy both theoretically and practically.

To support this statement, we provide several polynomial-time approximation algorithms, provide guaranteed approximation ratios for them, and evaluate exact approximation ratios empirically. The former depend on two parameters: the number m of alternatives and the size K of the committee to be selected. Our two most important algorithms have the following properties.

1. The first algorithm (Algorithm P from Section 4.6) finds a set of K candidates such that, on average, each agent is represented by someone whom she prefers to at least $(1 - \frac{2\mathrm{w}(K)}{K})$ fraction of the candidates, where w is Lambert's W-function, for which it is known that $\mathrm{w}(x) = \ln x - \ln\ln x + o(1)$. We stress that this guarantee holds for every possible profile of (complete) preference orders. This algorithm approximates the Chamberlin–Courant rule.
2. The second algorithm (Algorithm A from Section 4.1; see also its improvements, Algorithms B and C) guarantees that each agent is, on average, represented by someone whom she prefers to at least $(1 - \frac{K-1}{2(m-1)} - \frac{H_K}{K})$ fraction of the candidates, where H_K is the K-th harmonic number. It guarantees that each member of the committee represents roughly the same number of agents.[2] As before, this guarantee holds for every possible profile of (complete) preference orders. This algorithm approximates the Monroe rule.

The guarantees we provide for our algorithms are very strong. For instance, in the Polish parliamentary elections, where $K = 460$ and $m \approx 6000$, our algorithms guarantee that each voter will be, on average, represented by a candidate that she prefers to 99% of the candidates (in the case of Algorithm P) and to 96% of the candidates (in the case of Algorithm A). Our experimental data shows that in real-life settings we indeed achieve such high-quality results. Additionally, we show that the algorithms preserve their approximation properties even if we consider truncated ballots, in which agents rank only several most preferred candidates. The fact that our algorithms work well with truncated ballots may be very practically important. When the ballots are truncated, the results of an election are intrinsically approximate since the exact information about preferences of voters is not available. Hence the use of approximation algorithms in this case is even more pertinent.

Our paper gives a detailed account of the approximability of the Monroe and Chamberlin–Courant voting rules. Below we provide a more comprehensive summary of our results.

1. The utilitarian variants of the Monroe and Chamberlin–Courant rules are hard to approximate up to any constant factor for the dissatisfaction-based cases (see Theorems 1 and 2). These results justify studying the satisfaction-based setting. Constant-factor approximation algorithms for the dissatisfaction-based cases would have been more powerful, but, unless unlikely complexity class collapses occur, they do not exist.
2. For the satisfaction-based utilitarian framework we show the following. For the Monroe rule with the Borda scoring function we give a randomized $(0.715 - \epsilon)$-approximation algorithm (often, the ratio is much better; see Section 4 and the comment on Algorithm A above). In the case of an arbitrary positional scoring function we give a $(1 - \frac{1}{e})$-approximation algorithm (Theorem 10). For the Chamberlin–Courant rule with the Borda scoring function we give a polynomial-time approximation scheme (that is, for each ϵ, $0 < \epsilon < 1$, we have a polynomial-time $(1 - \epsilon)$-approximation algorithm; see Corollary 13 and the comment on Algorithm P above).

[1] We are very grateful to one of the anonymous reviewers for this observation.
[2] The exact meaning of this will be explained later.

3. We provide empirical evaluation of our algorithms for the satisfaction-based utilitarian framework, both on synthetic data and on real-life data. This evaluation shows that in practice our best algorithms achieve at least 0.9 approximation ratios, and even better results are typical (see Section 5).
4. We show that our algorithms work very well in the setting where voters do not necessarily rank all the candidates, but only provide truncated ballots, in which they rank several most preferred candidates (usually at least three). We provide theoretical guarantees on the performance of our algorithms (Propositions 6 and 14) as well as empirical evaluation (see Section 5.4).
5. We show that egalitarian variants of the Monroe and Chamberlin–Courant rules are hard to approximate up to any constant factor, both for the dissatisfaction-based case and for the satisfaction-based case (see Theorems 15, 16, 17 and 19; the discussion of the egalitarian variants is presented in Appendix C).

Our approximability results show that, as long as one is willing to accept approximate solutions, it is possible to use the utilitarian variants of the Monroe and Chamberlin–Courant rules in practice. This view is justified both from the theoretical and from the empirical point of view.

1.2. Related work

The Monroe and Chamberlin–Courant rules belong to the class of social assignment rules as they produce a representative committee and an assignment of voters to candidates. There are other voting rules that can perform this function but only one of them comes close in terms of axiomatic properties. This is the multiwinner Single Transferrable Vote rule (STV). The comparison of various axiomatic properties of multiwinner rules conducted by Elkind et al. [19] shows that variants of STV have many nice properties and, in particular, are rather strong on proportionality. One has to have in mind that a multiwinner STV is not a single rule but a large family of them. A number of different quotas can be used (the most common is the Droop quota). There are also variations, such as in how to transfer surplus votes from winning candidates and whether to transfer votes to already elected candidates (see [59] for the details). Tie-breaking under STV may be challenging and, indeed, there are natural tie-breaking rules that make the winner-determination problem in STV NP-hard [15]. It is still possible that a variant STV can be extended to a useful social assignment rule which may serve as an alternative to the Chamberlin–Courant rule (but not the Monroe one) but this is yet to be done.

There is a large body of work related to our research either in terms of motivation, perspective, methodology, or the exact subject of study. For example, in Appendix A we present a resource-allocation view of the Monroe and Chamberlin–Courant rules and, so, our results relate to those on resource-allocation. In terms of formal similarity, the winner determination problem for the Chamberlin–Courant rule can be seen as a close relative of the facility location problem. Finally, other authors have studied the complexity of and the algorithms for Monroe and Chamberlin–Courant voting rules before. Below we review this related literature.

Our methodology is similar to that previously used in the study of approximation algorithms for other NP-hard voting rules. These rules include, for example, Dodgson's rule [4,8,24], Young's rule [8,51], and Kemeny's rule [4,7,25]. For the single-transferable vote rule (STV), the winner determination problem becomes NP-hard if we use the parallel-universes tie-breaking [15]. Many of these hardness results hold even in the sense of parameterized complexity theory (however, there are also fixed-parameter tractability results; see the references above for the details).

The aforementioned hardness results motivated the search for approximation algorithms. There are now very good approximation algorithms for Kemeny's rule [1,16,28] and for Dodgson's rule [10,11,20,26,38]. In both cases the results are, in essence, optimal. For Kemeny's rule there is a polynomial-time approximation scheme [28] (although the running time of this algorithm is not very practical) and for Dodgson's rule the achieved approximation ratio is optimal under standard complexity-theoretic assumptions [10] (unfortunately, the approximation ratio is not constant but depends logarithmically on the number of candidates). On the other hand, for Young's rule it is known that no good approximation algorithms exist [10].

The work of Caragiannis et al. [11] and of Faliszewski et al. [20] on approximate winner determination for Dodgson's rule is particularly interesting from our perspective. In the former, the authors advocate treating approximation algorithms for Dodgson's rule as voting rules in their own right and design them to have desirable properties. In the latter, the authors show that a well-established voting rule, the Maximin rule, is a reasonable, though not optimal, approximation of Dodgson's rule. This perspective is important for anyone interested in using approximation algorithms for winner determination in elections and, in particular, for the case of Monroe and Chamberlin–Courant voting rules. For example, Elkind et al. [19] provide a discussion of axiomatic properties of multiwinner voting rules and show that some of our approximation algorithms have more desirable properties than the rules they approximate.

The complexity of the winner determination problem for the Monroe and Chamberlin–Courant rules has been considered in several papers. Procaccia, Rosenschein and Zohar [48] were the first to show the hardness of these two rules for the case of a particular approval-style dissatisfaction function. Their results were complemented by Lu and Boutilier [32] who proved the hardness of the Chamberlin–Courant rule for the case of Borda satisfaction function, by Betzler, Slinko and Uhlmann [9] who focused on the parameterized complexity of the rules for both the case of unrestriced voter preferences and the case of single-peaked preferences, by Yu, Chan, and Elkind [61] who considered voters with preferences that are single-peaked on a tree, and by Skowron et al. [58] who considered voters with single-crossing preferences. Further, Lu and Boutilier [32] initiated the study of approximability for the Chamberlin–Courant rule and were the first to use the satisfaction-based framework. Specifically, they gave the greedy $(1 - \frac{1}{e})$-approximation algorithm for the Chamberlin–Courant rule. Lu and Boutilier were motivated by applications to recommendation systems and, in that sense, our view of the rules is quite similar to theirs.

In this paper we take the view that the Monroe and Chamberlin–Courant rules are special cases of the following resource allocation problem (presented in Appendix A in more detail). The alternatives are viewed as shareable resources, each with a certain capacity defined as the maximal number of agents that may share this resource. Each agent has preferences over the resources and is entitled in getting exactly one. The goal is to select a predetermined number K of resources and to find an optimal allocation of these resources. This provides a unified framework for the two rules and reveals the connection of proportional representation problems to other resource allocation problems. In particular, it closely resembles multi-unit resource allocation with single-unit demand [54, Chapter 11] (see also the work of Chevaleyre et al. [14] for a survey of the most fundamental issues in the multiagent resource allocation theory) and resource allocation with sharable indivisible goods [2,14]. Below, we point out other connections of the Monroe and Chamberlin–Courant rules to several other problems.

Facility Location Problems. In the facility location problem, there are n customers located in some area and an authority, say a city council, that wants to establish a fixed number K of facilities to serve those customers. Customers incur certain costs (say transportation costs) of using the facilities. Further, establishing a facility may have additional cost, possibly depending on aspects such as the facility's location. The problem is to find K locations for the facilities that would minimize the total cost for the society. If these facilities have unlimited capacities and can serve any number of customers, then each customer would use her most preferred (i.e., closest) facility and the problem is similar to finding the Chamberlin–Courant assignment. If the capacities of the facilities are finite and equal, the problem looks like finding an assignment under the Monroe rule. An essential difference between this problem and the winner determination problem under the Chamberlin–Courant rule are the setup costs and the distance metric. The parameterized complexity of the Facility Location Problem was investigated by Fellows and Fernau [21]. The papers of Procaccia et al. [48] and of Betzler et al. [9] contain a brief discussion of the connection between the Facility Location Problem and the winner determination problem under the Chamberlin–Courant rule. Note that in the context of facility location, it makes sense to look at both the utilitarian setting and at the egalitarian setting. The latter is particularly appealing, for example, when we seek to locate such facilities as fire stations or intensive care units and we want to guarantee a particular level of service to every member of the society.[3]

Group Activity Selection Problem. In the group activity selection problem [17] we have a group of agents (say, conference attendees) and a set of activities (say, options that they have for a free afternoon such as a bus city tour or a wine tasting). The agents express preferences regarding the activities and organizers try to allocate agents to activities to maximize their total satisfaction. If the number of agents that can participate in each activity is unrestricted, this reduces to the winner determination problem under the Chamberlin–Courant rule. Similarly, if roughly the same number of agents should participate in each activity, the problem becomes very close to computing the winners under the Monroe rule. The Group Activity Selection Problem is, however, far more general and allows the agents to express quite complicated preferences. For example, an agent might have preferences of the form: "I like wine-tasting best provided that at most 10 people participate in it, and otherwise I prefer a bus city tour provided that at least 15 people participate, and otherwise I prefer to not take part in any activity." Darmann et al. [17] give some hardness and easiness results for this problem, but its investigation has only started.

Finally, we mention that finding winners under the Monroe and Chamberlin–Courant rules can be seen from the perspective of matching theory. One of the closest problems in this context is that of demand matching [52,56]. We are given a biparite graph $G = (U, V; E)$ and three functions: the profit function $\pi: E \to \mathbb{R}_+$, the demand function $d: E \to \mathbb{R}_+$, and the capacity function $b: U \cup V \to \mathbb{R}_+$. If we write $\delta(v)$ to denote the set of edges incident to a vertex v, the goal can be formulated as follows: Find a subset $M \subseteq E$ that maximizes the profit $\sum_{e \in M} \pi(e)$ under the constraint that for each vertex $w \in U \cup V$ we have $\sum_{e \in M \cap \delta(w)} d(e) \le b(w)$.

Intuitively, the connection of demand matching to our problems is as follows. The vertices on one side, say V, correspond to the agents and for each such vertex v in this group we set $b(v) = 1$. The vertices on the other side, U, correspond to the candidates; under Mondroe rule, for each vertex u on this side we set $b(u)$ to be the number of agents divided by the size of the committee and under Chamberlin–Courant, we set $b(u)$ to be the number of agents. We have a complete bipartite graph, where the profits of the edges are the satisfactions derived by the agents from being represented by the candidates. The significant difference between the winner-determination problems for our two rules and the demand matching problem is that the demand matching does not put constraints on the number of vertices from U involved in the matching (and we need that at most K candidate vertices are involved, where K is the size of the committee to be elected). Also, the profit functions that we focus on are of a very special form. The results of Shepherd and Vetta [52] and of Singh and Wu [56] show

[3] We thank one of the anonymous reviewers for this comment.

that the demand matching problem appears to be much harder to solve (approximately) than our problems. Nonehteless, we believe that exploring the connections between (demand) matching and our problems is a very promising direction.

The above connections show that, indeed, the complexity of winner determination under the Monroe and Chamberlin–Courant voting rules is interesting, can lead to progress in several other directions, and may have impact on applications within artificial intelligence.

2. Preliminaries

We first define basic notions such as preference orders and positional scoring rules. Then we present the formal definitions of the winner determination problems for the Monroe and Chamberlin–Courant voting rules. Finally, we briefly recall relevant notions regarding computational complexity.

Preferences. For each $n \in \mathbb{N}$, by $[n]$ we mean $\{1, \dots, n\}$. We assume that there is a set $N = [n]$ of *agents* and a set $A = \{a_1, \dots, a_m\}$ of *alternatives*. Each agent i has a *preference order* $\succ_i$ over A, i.e., a strict linear order of the form $a_{\pi(1)} \succ_i a_{\pi(2)} \succ_i \cdots \succ_i a_{\pi(m)}$ for some permutation π of $[m]$. For an alternative a, by $\mathrm{pos}_i(a)$ we mean the position of a in the i-th agent's preference order. For example, if a is the most preferred alternative for i then $\mathrm{pos}_i(a) = 1$, and if a is the least preferred one then $\mathrm{pos}_i(a) = m$. A collection $V = (\succ_1, \dots, \succ_n)$ of agents' preference orders is called a *preference profile*.

We often include subsets of the alternatives in the descriptions of preference orders. For example, if A is the set of all the alternatives and B is some nonempty strict subset of A, then by $B \succ A - B$ we mean that for the preference order $\succ$ all alternatives in B are preferred to those outside of B.

A *positional scoring function* (PSF) is a function $\alpha^m : [m] \to \mathbb{N}$. A PSF α^m is an *increasing positional scoring function* (IPSF) if for each $i, j \in [m]$, if $i < j$ then $\alpha^m(i) < \alpha^m(j)$. Analogously, a PSF α^m is a *decreasing positional scoring function* (DPSF) if for each $i, j \in [m]$, if $i < j$ then $\alpha^m(i) > \alpha^m(j)$.

Intuitively, if β^m is an IPSF then $\beta^m(i)$ can represent the *dissatisfaction* that an agent suffers when assigned to an alternative that is ranked i-th in her preference order. Thus, we assume that for each IPSF β^m it holds that $\beta^m(1) = 0$ (an agent is not dissatisfied by her top alternative). Similarly, a DPSF γ^m measures an agent's satisfaction and we assume that for each DPSF γ^m it holds that $\gamma^m(m) = 0$ (an agent is completely not satisfied with her least desired alternative). Sometimes we write α instead of α^m, when it cannot lead to confusion.

We will often speak of families α of IPSFs (DPSFs) of the form $\alpha = (\alpha^m)_{m \in \mathbb{N}}$, where α^m is a PSF on $[m]$, such that:

1. For a family of IPSFs it holds that $\alpha^{m+1}(i) = \alpha^m(i)$ for all $m \in \mathbb{N}$ and $i \in [m]$.
2. For a family of DPSFs it holds that $\alpha^{m+1}(i+1) = \alpha^m(i)$ for all $m \in \mathbb{N}$ and $i \in [m]$.

In other words, we build our families of IPSFs (DPSFs) by appending (prepending) values to functions with smaller domains. To simplify notation, we will refer to such families of IPSFs (DPSFs) as *normal* IPSFs (normal DPSFs). We assume that each function α^m from a family can be computed in polynomial time with respect to m. Indeed, we are particularly interested in the Borda families of IPSFs and DPSFs defined by $\alpha^m_{\mathrm{B,inc}}(i) = i - 1$ and $\alpha^m_{\mathrm{B,dec}}(i) = m - i$, respectively.

Assignment functions. A *K-assignment function* (or simply *K-assignment*) is any function $\Phi : N \to A$, such that $\|\Phi(N)\| \le K$ (that is, it assigns agents to at most K alternatives). A *Monroe K-assignment function* is an assignment function that additionally satisfies the following cardinality constraint: For each alternative $a \in A$ we have $\|\Phi(N)\| = K$ and either $\lfloor \frac{\|N\|}{K} \rfloor \le \|\Phi^{-1}(a)\| \le \lceil \frac{\|N\|}{K} \rceil$ or $\|\Phi^{-1}(a)\| = 0$.

We also consider partial assignment functions. A partial K-assignment function is defined in the same way as the regular one, except that it may assign a null alternative, $\perp$, to some of the agents. It is convenient to think that for each agent i we have $\mathrm{pos}_i(\perp) = m$. A partial Monroe K-assignment is a partial K-assignment that can be extended to a regular Monroe K-assignment (that is, a partial Monroe K-assignment can break the cardinality constraints, but it must be possible to extend it in such a way that these constraints are no longer broken). Given an assignment Φ, for each agent i we refer to alternative $\Phi(i)$ as the *representative* of i.

Given a normal IPSF (DPSF) α, we consider the following three functions, each assigning a positive integer to a given assignment Φ:

$$\ell^{\alpha}_{\mathrm{sum}}(\Phi) = \sum_{i=1}^{n} \alpha\big(\mathrm{pos}_i\big(\Phi(i)\big)\big),$$

$$\ell^{\alpha}_{\mathrm{max}}(\Phi) = \max\nolimits_{i=1}^{n} \alpha\big(\mathrm{pos}_i\big(\Phi(i)\big)\big),$$

$$\ell^{\alpha}_{\mathrm{min}}(\Phi) = \min\nolimits_{i=1}^{n} \alpha\big(\mathrm{pos}_i\big(\Phi(i)\big)\big).$$

These functions aggregate individual dissatisfaction (satisfaction) values of the agents to measure the overall quality of the assignment. In the utilitarian framework, we use the first function, $\ell^{\alpha}_{\mathrm{sum}}(\Phi)$, as the *total (societal) dissatisfaction function* in the IPSF case and as the *total (societal) satisfaction function* in the DPSF case. We use the second and the third functions, respectively, as the total dissatisfaction and satisfaction functions for IPSF and DPSF cases in the egalitarian framework (the egalitarian framework is discussed in Appendix C). We sometimes omit the word "total" if no confusion may arise.

The Monroe and Chamberlin–Courant rules. Let us now define the problems of winner determination under the Monroe and Chamberlin–Courant (CC) rules. In both cases the goal is to find an optimal assignment, where the optimality is relative to one of the total dissatisfaction or satisfaction functions that we have just introduced. The former is to be minimized and the latter is to be maximized.

Definition 1. Let α be a normal IPSF. An instance of α-CC-DisWinner problem consists of a set of agents $N = [n]$, a set of alternatives $A = \{a_1, \dots, a_m\}$, a preference profile V of the agents, and a positive integer K. We ask for a K-assignment function Φ such that $\ell_{\mathrm{sum}}^{\alpha}(\Phi)$ is minimized. The problem α-Monroe-DisWinner is defined in the same way but we additionally require Φ to be a Monroe K-assignment function.

Definition 2. Let α be a normal DPSF. The problem α-CC-SatWinner is defined in the same way as α-CC-DisWinner, except that we seek a K-assignment Φ such that $\ell_{\mathrm{sum}}^{\alpha}(\Phi)$ is maximized. The problem α-Monroe-SatWinner is defined in the same way but we additionally require Φ to be a Monroe K-assignment function.

Definitions 1 and 2 regard the utilitarian variants of the rules: We consider the egalitariant variants in Appendix C. Below we provide an example illustrating the utilitarian variants.

Example 1. Consider a set $N = \{1, 2, 3, 4, 5, 6\}$ of six agents and a set $A = \{a, b, c, d\}$ of four alternatives. Suppose the agents have the following preference orders:

$$1: a \succ b \succ c \succ d$$
$$2: a \succ c \succ b \succ d$$
$$3: a \succ d \succ c \succ b$$
$$4: a \succ b \succ d \succ c$$
$$5: b \succ c \succ a \succ d$$
$$6: c \succ d \succ b \succ a$$

and we are looking for a committee of size $K = 2$. Consider the dissatisfaction-based variant of Chamberlin–Courant rule with the Borda dissatisfaction function, $\alpha_{\mathrm{B,inc}}$ (i.e., problem $\alpha_{\mathrm{B,inc}}$-CC-DisWinner). In this case, there is a unique optimal K-assignment Φ:

$$\Phi(1) = \Phi(2) = \Phi(3) = \Phi(4) = a, \qquad \Phi(5) = \Phi(6) = c.$$

Agent 5 ranks her representative on the second position and all the other agents rank their representatives on the first positions. Thus, the aggregated dissatisfaction of the agents is $\ell_{\mathrm{sum}}^{\alpha_{\mathrm{B,inc}}}(\Phi) = 0 + 0 + 0 + 0 + 1 + 0 = 1$. If we considered the satisfaction-based variant of the problem ($\alpha_{\mathrm{B,dec}}$-CC-SatWinner), the assignment would be the same, but with the satisfaction $\ell_{\mathrm{sum}}^{\alpha_{\mathrm{B,dec}}}(\Phi) = 3 + 3 + 3 + 3 + 2 + 3 = 17$.

On the other hand, consider the dissatisfaction variant of the Monroe rule, that is, problem $\alpha_{\mathrm{B,inc}}$-Monroe-DisWinner. In this case we are looking for a Monroe K-assignment function, so each alternative who is a representative has to represent exactly three agents. The following assignment function, Φ', is optimal:

$$\Phi'(1) = \Phi'(3) = \Phi'(4) = a, \qquad \Phi'(2) = \Phi'(5) = \Phi'(6) = c.$$

The aggregated dissatisfaction is $\ell_{\mathrm{sum}}^{\alpha_{\mathrm{B,inc}}}(\Phi') = 0 + 1 + 0 + 0 + 1 + 0 = 2$. In the satisfaction based variant the same assignment is optimal, and its satisfaction is $\ell_{\mathrm{sum}}^{\alpha_{\mathrm{B,dec}}}(\Phi') = 3 + 2 + 3 + 3 + 2 + 3 = 16$.

As is clearly visible in the above example, in terms of exact solutions our satisfaction-based problems are equivalent to the dissatisfaction-based ones (indeed, one can always transform an IPSF to an equivalent DPSF and vice versa). However, as we will see, these two families of problems behave differently in terms of their approximability.

The goal in our problems is to compute a particular (Monroe) K-assignment. Naturally, such an assignment defines a set of K alternatives, who are viewed as the winners of the given multiwinner election. In this paper we make the following assumption.

Assumption. *We assume that the size K of the committee that we seek always divides the number n of the agents.*

Note that while this assumption looks very innocent at first, in fact in many related settings it can be crucial. For example, Elkind et al. [19] have shown that the Monroe rule satisfies homogeneity property when K divides the number of agents, but fails it in general. Nonetheless, this assumption does not affect our hardness proofs at all, and one can verify that all our approximation algorithms also work when it is not satisfied. However, this assumption simplifies the presentation.

For each subset of the alternatives $S \subseteq A$ such that $\|S\| \le K$, we write Φ^S_α to denote the partial (Monroe) K-assignment that assigns agents only to the alternatives from S and such that Φ^S_α maximizes the utilitarian satisfaction $\ell^\alpha_{\mathrm{sum}}(\Phi^S_\alpha)$. (We introduce this notation only for the utilitarian satisfaction-based setting because it is useful to express appropriate algorithms for this case. For other settings we only have hardness results and this notation would not be useful. Whether we mean a K-assignment or a Monroe K-assignment will always be clear from the context.)

Computational Issues. For many normal IPSFs α and, in particular, for the Borda IPSF, our problems are NP-hard [9,48]. Thus we seek approximate solutions.

Definition 3. Let r be a real number such that $r \ge 1$, let α be a normal IPSF (or, let α be a normal DPSF and let r be such that $0 < r \le 1$), and let $\mathcal{R}$ be either Monroe or CC. An algorithm is an r-approximation algorithm for α-$\mathcal{R}$-DisWinner problem (for α-$\mathcal{R}$-SatWinner problem) if on each instance I it returns a feasible assignment Φ such that $\ell^\alpha_{\mathrm{sum}}(\Phi) \le r \cdot \mathrm{OPT}$ (such that $\ell^\alpha_{\mathrm{sum}}(\Phi) \ge r \cdot \mathrm{OPT}$), where OPT is the optimal total dissatisfaction (satisfaction) $\ell^\alpha_{\mathrm{sum}}(\Phi_{\mathrm{OPT}})$.

Example 2. Let us once again analyze the setting from Example 1. Consider the dissatisfaction-based variant of Chamberlin–Courant rule and the following assignment:

$$\Phi''(1) = \Phi''(4) = \Phi''(5) = b, \qquad \Phi''(2) = \Phi''(3) = \Phi''(6) = c.$$

(Note that this is the best K-assignment function if both b and c are to be the representatives; also, this assignment happens to be a Monroe 2-assignment.) The total dissatisfaction of the agents is $\ell^{\alpha_{\mathrm{B,inc}}}_{\mathrm{sum}}(\Phi'') = 1+1+2+1+0+0 = 4$. Since the optimal dissatisfaction is 1, this is a 4-approximate solution for $\alpha_{\mathrm{B,inc}}$-CC-DisWinner. (For the case of Monroe the optimal dissatisfaction is 2, so this is also a 2-approximate solution for $\alpha_{\mathrm{B,inc}}$-Monroe-DisWinner.)

On the other hand, consider the satisfaction-based setting. The optimal satisfaction of the voters is 17 but under Φ'' the satisfaction is $\ell^{\alpha_{\mathrm{B,dec}}}_{\mathrm{sum}}(\Phi'') = 2+2+1+2+3+3 = 13$. That is, Φ'' is a 0.76-approximate solution for $\alpha_{\mathrm{B,dec}}$-CC-SatWinner (and, since the optimal satisfaction for the case of Monroe is 16, this is a 0.81-approximate solution for $\alpha_{\mathrm{B,dec}}$-Monroe-SatWinner).

Lu and Boutilier [32] gave a $(1-\frac{1}{e})$-approximation algorithm for the CC-SatWinner family of problems. Throughout this paper, for each of our problems we either prove inapproximability with respect to any constant r (under standard complexity-theoretic assumptions) or we present an approximation algorithm. In our inapproximability proofs, we use the following two classic NP-complete problems [23].

Definition 4. An instance I of Set-Cover consists of set $U = [n]$ (called the ground set), family $\mathcal{F} = \{F_1, F_2, \dots, F_m\}$ of subsets of U, and positive integer K. We ask if there exists a set $J \subseteq [m]$ such that $\|J\| \le K$ and $\bigcup_{j \in J} F_j = U$.

Definition 5. X3C is a variant of Set-Cover where $\|U\|$ is divisible by 3, each member of $\mathcal{F}$ has exactly three elements, and $K = \frac{\|U\|}{3}$. (Since K is defined implicitly, an X3C instance consists only of the set U and the family $\mathcal{F}$.)

Set-Cover remains NP-complete even if we restrict each member of U to be contained in at most two sets from $\mathcal{F}$ (it suffices to note that this restriction is satisfied by Vertex-Cover, which is a special case of Set-Cover). X3C remains NP-complete even if we additionally assume that n is divisible by 2 and each member of U appears in at most 3 sets from $\mathcal{F}$ [23].

We also use results from the theory of parameterized complexity developed by Downey and Fellows [18]. This theory allows one to single out a particular parameter, say k, and analyze its 'contribution' to the overall complexity of the problem. An analogue of the class P here is the class FPT which contains problems that can be solved in time $f(k)n^{O(1)}$, where n is the size of the input instance and f is some computable function. The class of problems that can be solved in polynomial time provided the value of the parameter is fixed is called XP; note that FPT $\subseteq$ XP. Parameterized complexity theory also operates with classes W[1] $\subseteq$ W[2] $\subseteq \cdots$ which are believed to form a hierarchy of classes of *hard* problems (combined, they are analogous to the class NP). It holds that FPT $\subseteq$ W[1], but it seems unlikely that FPT = W[1], let alone FPT = W[2]. We point the reader to the books of Niedermeier [44] and Flum and Grohe [22] for detailed overviews of parametrized complexity theory. Interestingly, while both Set-Cover and Vertex-Cover are NP-complete, the former is W[2]-complete and the latter belongs to FPT (see, e.g., the book of Niedermeier [44] for these now-standard results and their history).

3. Hardness of approximation

We now present our inapproximability results for the Monroe and Chamberlin–Courant rules in the dissatisfaction-based utilitarian case. We show similar inapproximability results for the egalitarian variants of our rules in Appendix C.

Theorem 1. *For each normal IPSF α and each constant factor $r > 1$, there is no polynomial-time r-approximation algorithm for α-Monroe-DisWinner unless* P = NP.

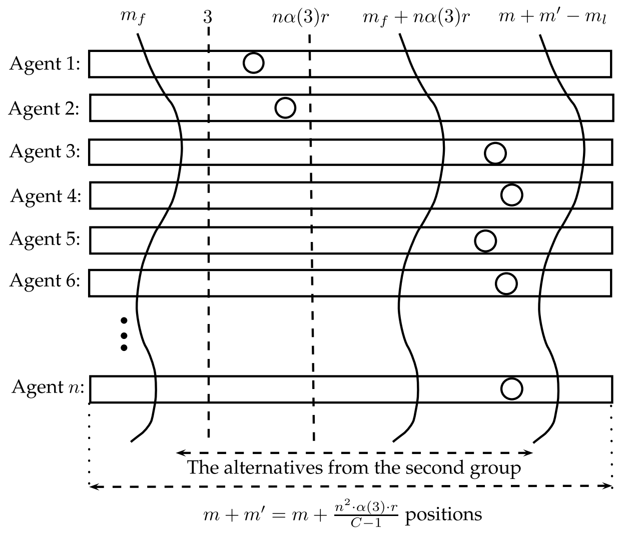


Fig. 1. The alignment of the positions in the preference orders of the agents. The positions are numbered from the left to the right. The left wavy line shows the positions $m_f(\cdot)$, each no greater than 3. The right wavy line shows the positions $m_l(\cdot)$, each higher than $nr \cdot \alpha(3)$. The alternatives from A_2 (positions of one such an alternative is illustrated with circles) are placed only between the peripheral wavy lines. Each alternative from A_2 is placed on the left from the middle wavy line exactly 2 times, thus each such alternative is placed on the left from the right dashed line no more than 2 times (exactly two times in the figure).

Proof. Let us fix a normal IPSF α and let us assume, aiming at getting a contradiction, that there is some constant $r > 1$ and a polynomial-time r-approximation algorithm $\mathcal{A}$ for α-Monroe-DisWinner.

Let I be an instance of X3C with a ground set $U = [n]$ and a family $\mathcal{F} = \{F_1, F_2, \dots, F_m\}$ of 3-element subsets of U. Without loss of generality, we assume that n is divisible by both 2 and 3 and that each member of U appears in at most 3 sets from $\mathcal{F}$.

Using I, we build an instance I_M of α-Monroe-DisWinner as follows. We set $N = U$ (that is, the elements of the ground set are the agents) and we set $A = A_1 \cup A_2$, where $A_1 = \{a_1, \dots, a_m\}$ is a set of alternatives corresponding to the sets from the family $\mathcal{F}$, and A_2 is a set of dummy alternatives of cardinality $\|A_2\| = \frac{1}{2}n^2 r \cdot \alpha(3)$, needed for the construction. We let $m' = \|A_2\|$ and rename the alternatives in A_2 so that $A_2 = \{b_1, \dots, b_{m'}\}$. We set $K = \frac{n}{3}$.

We build agents' preference orders using the following algorithm. For each $j \in N$, set $M_f(j) = \{a_i \mid j \in F_i\}$ and $M_l = \{a_i \mid j \notin F_i\}$. Set $m_f(j) = \|M_f(j)\|$ and $m_l(j) = \|M_l(j)\|$. As the frequency of the elements from U is bounded by 3, we have $m_f(j) \le 3$. For each agent j we set her preference order to be of the form $M_f(j) \succ_j A_2 \succ_j M_l(j)$, where the alternatives in $M_f(j)$ and $M_l(j)$ are ranked in an arbitrary way and the alternatives from A_2 are placed at positions $m_f(j)+1, \dots, m_f(j)+m'$ in the way described below (see Fig. 1 for a high-level illustration of the construction).

We place the alternatives from A_2 in the preference orders of the agents in such a way that for each alternative $b_i \in A_2$ there are at most two agents that rank b_i among their $nr \cdot \alpha(3)$ top alternatives. The following construction achieves this effect. If $(i + j) \bmod n < 2$, then alternative b_i is placed at one of the positions $m_f(j) + 1, \dots, m_f(j) + nr \cdot \alpha(3)$ in j's preference order. Otherwise, b_i is placed at a position with index higher than $m_f(j) + nr \cdot \alpha(3)$ (and, thus, at a position with index higher than $nr \cdot \alpha(3)$). This construction can be implemented because for each agent j there are exactly $m' \cdot \frac{2}{n} = nr \cdot \alpha(3)$ alternatives $b_{i_1}, b_{i_2}, \dots, b_{i_{nr\alpha(3)}}$ such that $(i_t + j) \bmod n < 2$, where $t \in [nr\alpha(3)]$.

Let Φ be an assignment computed by $\mathcal{A}$ on I_M. We will show that $\ell^{\alpha}_{\text{sum}}(\Phi) \le n \cdot \alpha(3) \cdot r$ if and only if I is a *yes*-instance of X3C.

($\Leftarrow$) If there exists a solution for I (i.e., an exact cover of U with $\frac{n}{3}$ sets from $\mathcal{F}$), then we can easily show an assignment in which each agent j is assigned to an alternative from the top $m_f(j)$ positions of her preference order (namely, one that assigns each agent j to the alternative $a_i \in A_1$ that corresponds to the set F_i, from the exact cover of U, that contains j). Thus, for the optimal assignment Φ_{OPT} it holds that $\ell^{\alpha}_{\text{sum}}(\Phi_{\text{OPT}}) \le \alpha(3) \cdot n$. In consequence, since $\mathcal{A}$ is an r-approximation algorithm, it must return an assignment with the total dissatisfaction at most $nr \cdot \alpha(3)$.

($\Rightarrow$) Let us now consider the opposite direction. We assume that $\mathcal{A}$ found an assignment Φ such that $\ell^{\alpha}_{\text{sum}}(\Phi) \le nr \cdot \alpha(3)$ and we will show that I is a *yes*-instance of X3C. Since we require each alternative to be assigned to either 0 or 3 agents, if some alternative b_i from A_2 were assigned to some 3 agents, at least one of them would rank b_i at a position worse than $nr \cdot \alpha(3)$. This would mean that $\ell^{\alpha}_{\text{sum}}(\Phi) \ge nr \cdot \alpha(3) + 1$. Analogously, no agent j can be assigned to an alternative that is placed at one of the $m_l(j)$ bottom positions of j's preference order. Thus, only the alternatives in A_1 have agents assigned to them and, further, if agents x, y, z, are assigned to some $a_i \in A_1$, then it holds that $F_i = \{x, y, z\}$ (recall that F_i is a set from the input X3C instance; we will call each set F_i for which alternative a_i is assigned to some agents x, y, z *selected*). Since each agent is assigned to exactly one alternative, the selected sets are disjoint. Since the number of selected sets is $K = \frac{n}{3}$, it must be the case that the selected sets form an exact cover of U. Thus, I is a *yes*-instance of X3C. □

One may wonder if hardness of approximation for α-Monroe-DisWinner is not an artifact of the strict requirements regarding the number of chosen candidates. It turns out that unless $\mathrm{P} = \mathrm{NP}$, then for each $r, s > 1$ there is no (r, s)-approximation algorithm that finds an assignment with the following properties: (1) the aggregated dissatisfaction $\ell^{\alpha}_{\mathrm{sum}}(\Phi)$ is at most r times higher than the optimal one, (2) the number of alternatives to which agents are assigned is at most sK and (3) each selected alternative (i.e., each alternative that has agents assigned), is assigned to no more than $s\frac{n}{K}$ and no less than $\frac{1}{s}\frac{n}{K}$ agents. (The proof is similar to the one used for Theorem 1.) Thus, in our further study we do not consider such relaxations of the problem.

A result analogous to Theorem 1 also holds for the CC-DisWinner family of problems.

Theorem 2. *For each normal IPSF α and each constant factor $r > 1$, there is no polynomial-time r-approximation algorithm for α-CC-DisWinner unless $\mathrm{P} = \mathrm{NP}$.*

Proof. The proof is in Appendix D. □

The above results show that the task of computing winner sets with aggregated voter dissatisfaction close to the optimal one is computationally hard both for the Monroe and Chamberlin–Courant rules. On the other hand, if we focus on agents' total satisfaction then polynomial-time constant-factor approximation algorithms exist in many cases (see, e.g., the work of Lu and Boutilier [32] and the next section).

4. Algorithms for the utilitarian, satisfaction-based cases

We now turn to approximation algorithms for the Monroe and Chamberlin–Courant multiwinner voting rules in the satisfaction-based framework. Indeed, if one focuses on agents' total satisfaction then it is possible to obtain high-quality approximation results. In particular, we show the first nontrivial (randomized) approximation algorithm for $\alpha_{\mathrm{B,dec}}$-Monroe-SatWinner (recall that $\alpha_{\mathrm{B,dec}}$ is the Borda family of DPSFs; for the case of m candidates, we have $\alpha^{m}_{\mathrm{B,dec}}(i) = m - i$). We show that for each $\epsilon > 0$ we can provide a randomized polynomial-time algorithm that achieves $0.715 - \epsilon$ approximation ratio; the algorithm usually achieves even a better approximation. For the case of arbitrarily selected DPSF we show a $(1 - \frac{1}{e})$-approximation algorithm. Finally, we present the first polynomial-time approximation scheme (PTAS) for $\alpha_{\mathrm{B,dec}}$-CC-SatWinner. These results stand in sharp contrast to those from the previous section, where we have shown that approximation is hard for the dissatisfaction-based framework.

The core difficulty in computing the winners under the Monroe and Chamberlin–Courant rules is in selecting the alternatives that should be assigned to the agents. Given a preference profile and a set S of up to K alternatives, using the standard network-flow argument, it is easy to find a (possibly partial) optimal assignment Φ^{S}_{α} of the agents to the alternatives from S.

Proposition 3. *(Implicit in the paper of Betzler et al. [9].) Let α be a normal DPSF, N be a set of agents, A be a set of alternatives, V be a preference profile of N over A, and S a K-element subset of A (where K divides $\|N\|$). Then there is a polynomial-time algorithm that computes a (possibly partial) optimal K-assignment Φ^{S}_{α} (Monroe K-assignment Φ^{S}_{α}) of the agents to the alternatives from S.*

Note that for the case of the Chamberlin–Courant rule the algorithm from the above proposition can be greatly simplified: To each voter we assign the candidate that she ranks highest among those from S. For the case of Monroe, unfortunately, we need the expensive, though still polynomial-time, network-flow-based approach. Nonetheless, Proposition 3 allows us to focus on the issue of selecting the winning alternatives and not on the issue of matching them to the agents.

Below we describe our algorithms for $\alpha_{\mathrm{B,dec}}$-Monroe-SatWinner and for $\alpha_{\mathrm{B,dec}}$-CC-SatWinner. Formally speaking, every approximation algorithm for $\alpha_{\mathrm{B,dec}}$-Monroe-SatWinner also gives feasible results for $\alpha_{\mathrm{B,dec}}$-CC-SatWinner. However, some of our algorithms are particularly well-suited for both problems and some are tailored to only one of them. Thus, for each algorithm we clearly indicate if it is meant only for the case of Monroe, only for the case of Chamberlin–Courant, or if it naturally works for both systems.

4.1. Algorithm A (Monroe)

Perhaps the most natural approach to solve $\alpha_{\mathrm{B,dec}}$-Monroe-SatWinner is to build a solution iteratively: In each step we pick some not-yet-assigned alternative a_i (using some criterion) and assign it to those $\frac{n}{K}$ agents that (a) are not assigned to any other alternative yet, and (b) whose satisfaction of being matched with a_i is maximal. It turns out that this idea, implemented formally as Algorithm A (see pseudocode in Fig. 2), works very well in many cases. We provide a lower bound on the total satisfaction it guarantees in the next lemma. We remind the reader that the j-th *harmonic number* $H_j = \sum_{i=1}^{j}\frac{1}{i}$ has asymptotics $H_j = \Theta(\log j)$.

Lemma 4. *For every preference profile, Algorithm A returns a solution in which each agent is, on average, represented by an alternative that she prefers to at least $(1 - \frac{K-1}{2(m-1)} - \frac{H_K}{K})$ fraction of the alternatives.*

```
Notation: $\Phi \leftarrow$ a map defining a partial assignment, iteratively built by the algorithm.
    $\Phi^{\leftarrow} \leftarrow$ the set of agents for which the assignment is already defined.
    $\Phi^{\rightarrow} \leftarrow$ the set of alternatives already used in the assignment.
if $K \le 2$ then
    compute the optimal solution using an algorithm of Betzler et al. [9] and return.
$\Phi = \{\}$
for $i \leftarrow 1$ to $K$ do
    $score \leftarrow \{\}$
    $bests \leftarrow \{\}$
    foreach $a_i \in A \setminus \Phi^{\rightarrow}$ do
        $agents \leftarrow$ sort $N \setminus \Phi^{\leftarrow}$ so that if agent $j$ precedes agent $j'$ then $pos_j(a_i) \le pos_{j'}(a_i)$
        $bests[a_i] \leftarrow$ choose first $\frac{n}{K}$ elements from $agents$
        $score[a_i] \leftarrow \sum_{j \in bests[a_i]}(m - pos_j(a_i))$
    $a_{best} \leftarrow \mathrm{argmax}_{a \in A \setminus \Phi^{\rightarrow}}\, score[a]$
    foreach $j \in bests[a_{best}]$ do
        $\Phi[j] \leftarrow a_{best}$
```

Fig. 2. The pseudocode for Algorithm A.

Proof. Let $n = \|N\|$. When $K \le 2$, our algorithm computes an optimal solution exactly so we assume that $K \ge 3$. Let us consider the situation in the algorithm after the i-th iteration of the outer loop (we have $i = 0$ if no iteration has been executed yet). So far, the algorithm has picked i alternatives and assigned them to $i\frac{n}{K}$ agents (recall that for simplicity we assume that K divides n evenly). Hence, each agent has $\lceil \frac{m-i}{K-i} \rceil$ unassigned alternatives among her $i + \lceil \frac{m-i}{K-i} \rceil$ top-ranked alternatives. By the pigeonhole principle, this means that there is an unassigned alternative a_ℓ who is ranked among the top $i + \lceil \frac{m-i}{K-i} \rceil$ positions by at least $\frac{n}{K}$ agents. To see this, note that there are $(n - i\frac{n}{K})\lceil \frac{m-i}{K-i} \rceil$ slots for unassigned alternatives among the top $i + \lceil \frac{m-i}{K-i} \rceil$ positions in the preference orders of unassigned agents, and that there are $m - i$ unassigned alternatives. As a result, there must be an alternative a_ℓ for whom the number of agents that rank him or her among the top $i + \lceil \frac{m-i}{K-i} \rceil$ positions is at least:

$$\frac{1}{m-i}\left(\left(n - i\frac{n}{K}\right)\left\lceil \frac{m-i}{K-i} \right\rceil\right) \ge \frac{n}{m-i}\left(\frac{K-i}{K}\right)\left(\frac{m-i}{K-i}\right) = \frac{n}{K}.$$

In consequence, the $\frac{n}{K}$ agents assigned in the next step of the algorithm will have the total satisfaction at least $\frac{n}{K} \cdot (m - i - \lceil \frac{m-i}{K-i} \rceil)$. Thus, summing over the K iterations, the total satisfaction guaranteed by the assignment Φ computed by Algorithm A is at least the following value: (to derive the fifth line from the fourth one we note that $K(H_K - 1) - H_K \ge 0$ when $K \ge 3$):

$$\begin{aligned}
\ell_{\mathrm{sum}}^{\alpha_{\mathrm{B,dec}}}(\Phi) &\ge \sum_{i=0}^{K-1} \frac{n}{K} \cdot \left(m - i - \left\lceil \frac{m-i}{K-i} \right\rceil\right) \\
&\ge \sum_{i=0}^{K-1} \frac{n}{K} \cdot \left(m - i - \frac{m-i}{K-i} - 1\right) \\
&= \sum_{i=1}^{K} \frac{n}{K} \cdot \left(m - i - \frac{m-1}{K-i+1} + \frac{i-2}{K-i+1}\right) \\
&= \frac{n}{K}\left(\frac{K(2m-K-1)}{2} - (m-1)H_K + K(H_K - 1) - H_K\right) \\
&\ge \frac{n}{K}\left(\frac{K(2m-K-1)}{2} - (m-1)H_K\right) \\
&\ge (m-1)n\left(1 - \frac{K-1}{2(m-1)} - \frac{H_K}{K}\right)
\end{aligned}$$

This completes the proof. $\square$

The relation between m, the ratio K/m, and the guarantee of satisfaction of an average voter is depicted in Fig. 3.

Corollary 5. *Algorithm A is a polynomial-time* $(1 - \frac{K-1}{2(m-1)} - \frac{H_K}{K})$*-approximation algorithm for* $\alpha_{\mathrm{B,dec}}$-MONROE-SATWINNER.

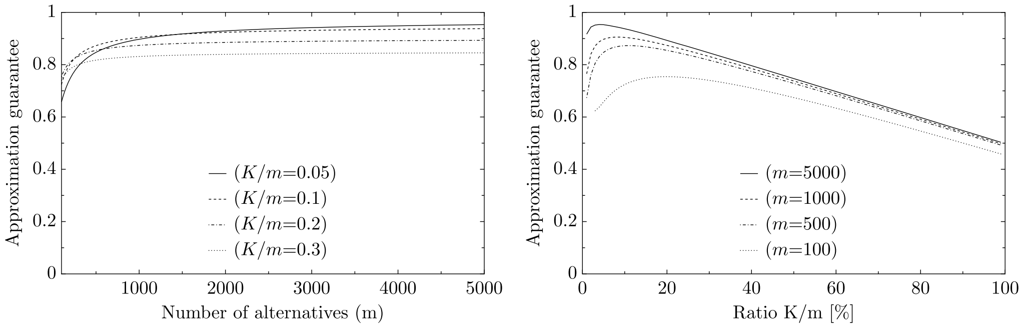


Fig. 3. The relation between the number of alternatives m (left figure) or the ratio K/m (right figure) and the guarantee of satisfaction of an average voter, $1 - \frac{K-1}{2(m-1)} - \frac{H_K}{K}$, under Algorithm A for the Monroe system.

Proof. If each agent were assigned to her top alternative, the total satisfaction would be equal to $(m-1)n$. Thus, by applying Lemma 4 we get the following bound:

$$\frac{\ell_{\mathrm{sum}}^{\alpha_{\mathrm{B,dec}}}(\Phi)}{\mathrm{OPT}} \le 1 - \frac{K-1}{2(m-1)} - \frac{H_K}{K}. \qquad \square$$

Note that in the above proof we measure the quality of our assignment against a, perhaps-impossible, perfect solution, where each agent is assigned to her top alternative. This means that for relatively large m and K, and small $\frac{K}{m}$ ratio, the algorithm can achieve a close-to-ideal solution irrespective of the voters' preference orders. We believe that this is an argument in favor of using Monroe's system in multiwinner elections. On the flip side, to obtain a better approximation ratio, we would have to use a more involved bound on the quality of the optimal solution. To see that this is the case, form an instance I of $\alpha_{\mathrm{B,dec}}$-Monroe-SatWinner with n agents and m alternatives, where all the agents have the same preference order, and where we seek to elect K candidates (and where K divides n). It is easy to see that each solution that assigns the K universally top-ranked alternatives to the agents is optimal. Thus the total dissatisfaction of the agents in the optimal solution is:

$$\frac{n}{K}\big((m-1)+\cdots+(m-K)\big) = \frac{n}{K}\left(\frac{K(2m-K-1)}{2}\right) = n(m-1)\left(1 - \frac{K-1}{2(m-1)}\right).$$

By taking large enough m and K (even for a fixed value of $\frac{m}{K}$), the fraction $1 - \frac{K-1}{2(m-1)}$ can be arbitrarily close to the approximation ratio of our algorithm (the reasoning here is somewhat in the spirit of the idea of identifying maximally robust elections, as studied by Shiryaev, Yu, and Elkind [53]).

For small values of K, it is possible that the $\frac{H_K}{K}$ part of our approximation ratio would dominate the $\frac{K-1}{2(m-1)}$ part. In such cases we can use the result of Betzler et al. [9], who showed that for each fixed constant K, $\alpha_{\mathrm{B,dec}}$-Monroe-SatWinner can be solved in polynomial time. Thus, for the finite number of cases, where $\frac{H_K}{K}$ is too large, we can solve the problem optimally using their algorithm. In consequence, the quality of the solution produced by Algorithm A most strongly depends on the ratio $\frac{K-1}{2(m-1)}$. In most cases we can expect it to be small. For example, in Polish parliamentary elections $K = 460$ and $m \approx 6000$; in this case the greedy algorithm's approximation ratio is about 0.96. Put differently, in this case our algorithm guarantees to provide a solution in which each voter is, on average, represented by a candidate that she prefers to at least 96% of the candidates.

Our algorithm has one more great advantage: Since it focuses on the top parts of voters' preference orders, it can achieve very good results even if the voters submit truncated ballots (that is, if they rank some of their top alternatives only).[4] Below we present the formal analysis of the algorithm's approximation ratio for this case. Unfortunately, we did not obtain a closed form formula and, instead, we present the guaranteed approximation ratio as a sum, in Proposition 6 below. We also present the relation between the fraction of the top alternatives ranked by each of the voters and the approximation ratio for few values of m and K in Fig. 4.

Proposition 6. *Let P be the number of top positions in the agents' preference orders that are known by the algorithm. In this case Algorithm A returns a solution in which each agent is, on average, represented by an alternative that she prefers to at least r fraction of the alternatives (in particular, this algorithm is a polynomial-time r-approximation algorithm for $\alpha_{\mathrm{B,dec}}$-Monroe-SatWinner), where:*

[4] The study of truncated ballots is receiving more and more attention in the context of computational social choice. For example, Baumeister et al. [5] consider campaign management for truncated ballots, Narodytska and Walsh [42] consider the complexity of strategic voting, and Lackner [30] studies algorithms for recognizing single-peakedness in this case.

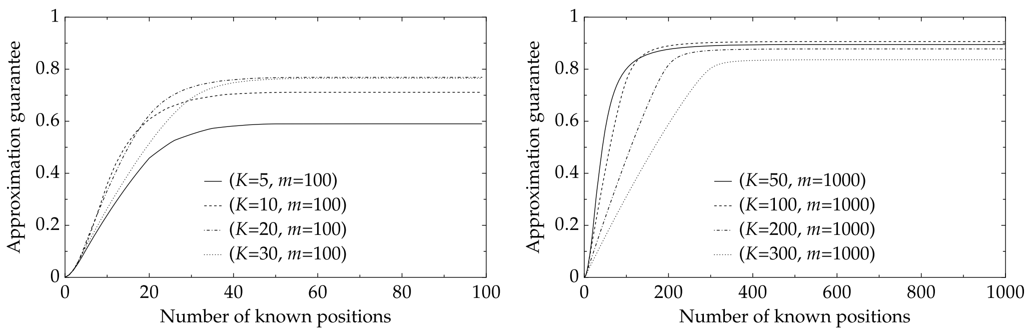


Fig. 4. The relation between the number of the known positions and the approximation ratio of Algorithm A for $\alpha_{\mathrm{B,dec}}$-Monroe-SatWinner.

$$r = \sum_{i=0}^{K-1} \frac{1}{n(m-1)} \max\bigl(r(i), 0\bigr)$$

and

$$r(i) = \begin{cases} \frac{n}{K}(m - i - \frac{m-i}{K-i}) & \textit{if } (i + \frac{m-i}{K-i}) \le P, \\ \frac{n}{K}\frac{(K-i)(m-i)}{4} & \textit{if } (i + \frac{m-i}{K-i}) > P \textit{ and } (2P - m) \ge i \ge (K-2), \\ \frac{n}{K}\frac{(m-P)(K-i)(P-i)}{m-i} & \textit{otherwise.} \end{cases}$$

Proof. The proof is in Appendix D. □

For example, for the case of Polish parliamentary elections ($K = 460$ and $m = 6000$), to achieve 90% of voters' optimal satisfaction, each voter would have to rank about 8.7% of the candidates. Of course, 8.7% of 6000 candidates is still over 500. It is too much for the algorithm to be directly useful in these elections, but it indicates the power of truncated ballots. Extending our approach to further limit the number of candidates each voter has to rank is an important open issue. We stress again that our results give even stronger guarantees than just an approximation. For instance, in our example, if each voter ranks about 8.7% of the candidates then, independently of the agents' preferences, on average each agent will be represented by someone who she prefers to 90% of the candidates.

Our results show that for most settings there is very little reason to ask the agents to rank all the alternatives. Using Proposition 6, election designers can estimate how many alternatives the agents should rank to obtain a particular level of satisfaction and, since computing preference orders can be expensive for the agents, this can save a large amount of effort.

4.2. Algorithm B (Monroe)

There are simple ways in which we can improve the quality of the assignment Φ produced by Algorithm A. For example, we can keep the elected committee $S = \Phi(N)$ but optimally reassign the alternatives to the voters using the algorithm in Proposition 3 obtaining the assignment Φ_{α}^{S}. This is exactly what our Algorithm B is doing. As shown in Section 5, this very simple trick turns out to noticeably improve the results of the Algorithm A in practice (and, of course, the theoretical approximation guarantees of Algorithm A carry over to Algorithm B).

4.3. Algorithm C (Monroe, CC)

Algorithm C is a further heuristic improvement over Algorithm B. This time the idea is that instead of keeping only one partial function Φ that is iteratively extended up to the full assignment, we keep a list of up to d partial assignment functions, where d is a parameter of the algorithm. At each iteration, for each assignment function Φ among the d stored ones and for each alternative a to which Φ has not assigned agents yet, we compute an optimal extension of this Φ that assigns agents to a. As a result we obtain possibly more than d (partial) assignment functions. For the next iteration we keep those d that give highest satisfaction.

We provide pseudocode for Algorithm C in Fig. 5. If we take $d = 1$, we obtain Algorithm B. If we also disregard the last two lines prior to returning the solution, we obtain Algorithm A.

Algorithm C can also be adapted for the Chamberlin–Courant rule. The only difference concerns creating the assignment functions: we replace the contents of the first foreach loop with the following code:

```
Notation: $\Phi \leftarrow$ a map defining a partial assignment, iteratively built by the algorithm.
          $\Phi^{\leftarrow} \leftarrow$ the set of agents for which the assignment is already defined.
          $\Phi^{\rightarrow} \leftarrow$ the set of alternatives already used in the assignment.
          $Par \leftarrow$ a list of partial representation functions
$Par = []$
$Par$.push({})
for $i \leftarrow 1$ to $K$ do
    $newPar = []$
    for $\Phi \in Par$ do
        $bests \leftarrow \{\}$
        foreach $a_i \in A \setminus \Phi^{\rightarrow}$ do
            $agents \leftarrow$ sort $N \setminus \Phi^{\leftarrow}$ (agent $j$ precedes agent $j'$ implies that $pos_j(a_i) \le pos_{j'}(a_i)$)
            $bests[a_i] \leftarrow$ choose first $\frac{N}{K}$ elements of $agents$
            $\Phi' \leftarrow \Phi$
            foreach $j \in bests[a_i]$ do
                $\Phi'[j] \leftarrow a_i$
            $newPar$.push($\Phi'$)
        sort $newPar$ according to descending order of the total satisfaction of the assigned agents
        $Par \leftarrow$ chose first $d$ elements of $newPar$
for $\Phi \in Par$ do
    $\Phi \leftarrow$ compute the optimal representative function using an algorithm of Betzler et al. [9] for the set of winners $\Phi^{\rightarrow}$
return the best representative function from Par
```

Fig. 5. The pseudocode for Algorithm C.

```
foreach $a_i \in A \setminus \Phi^{\rightarrow}$ do
    $\Phi' \leftarrow \Phi$
    foreach $j \in N$ do
        if agent $j$ prefers $a_i$ to $\Phi'(j)$ then
            $\Phi'(j) \leftarrow a_i$
    $newPar$.push($\Phi'$)
```

Note that, for the case of the Chamberlin–Courant rule, Algorithm C can also be seen as a generalization of Algorithm GM that we discuss in Section 4.5.

4.4. Algorithm R (Monroe, CC)

Algorithms A, B and C achieve very high approximation ratios for the cases where K is small relative to m. For the remaining cases, where K and m are comparable, we can use a sampling-based randomized algorithm (denoted as Algorithm R) described below. We focus on the case of Monroe and we briefly mention the case of CC at the end.

The idea of this algorithm is to randomly pick K alternatives and match them optimally to the agents, using Proposition 3. Naturally, such an algorithm might be very unlucky and pick K alternatives that all of the agents rank low. Yet, if K is comparable to m then it is likely that such a random sample would include a large chunk of some optimal solution. In the lemma below, we assess the expected satisfaction obtained with a single sampling step (relative to the satisfaction given by the optimal solution) and the probability that a single sampling step gives satisfaction close to the expected one. Naturally, in practice one should try several sampling steps and pick the one with the highest satisfaction.

Lemma 7. *A single sampling step of the randomized algorithm for $\alpha_{\mathrm{B,dec}}$-Monroe-SatWinner achieves expected approximation ratio of $\frac{1}{2}(1 + \frac{K}{m} - \frac{K^2}{m^2-m} + \frac{K^3}{m^3-m^2})$. Let p_ϵ denote the probability that the relative difference between the expected total satisfaction and the obtained total satisfaction is higher than ϵ. Then we have $p_\epsilon \le \frac{1}{1+\epsilon}$.*

Proof. Let $N = [n]$ be the set of agents, $A = \{a_1, \dots, a_m\}$ be the set of alternatives, and V be the preference profile of the agents. Let us fix some optimal solution Φ_{opt} and let A_{opt} be the set of alternatives assigned to the agents in this solution. For each $a_i \in A_{\mathrm{opt}}$, we write $\mathrm{sat}(a_i)$ to denote the total satisfaction of the agents assigned to a_i in Φ_{opt}. Naturally, we have $\sum_{a \in A_{\mathrm{opt}}} \mathrm{sat}(a) = \mathrm{OPT}$. In a single sampling step, we choose uniformly at random a K-element subset B of A. Then, we form a solution Φ_B by matching the alternatives in B optimally to the agents (via Proposition 3). We write K_{opt} to denote the random variable equal to $\|A_{\mathrm{opt}} \cap B\|$, the number of sampled alternatives that belong to A_{opt}. We define $p_i = \Pr(K_{\mathrm{opt}} = i)$ to be the probability that B contains i alternatives from A_{opt}. For each $j \in \{1, \dots, K\}$, we write X_j to denote the random variable equal to the total satisfaction of the agents assigned to the j-th alternative from the sample. We claim that for each i, $0 \le i \le K$, it holds that:

$$\mathbb{E}\left(\sum_{j=1}^{K} X_j \;\middle|\; K_{\mathrm{opt}} = i\right) \ge \frac{i}{K}\mathrm{OPT} + \frac{m-i-1}{2} \cdot \left(n - i\frac{n}{K}\right).$$

Why is this so? Given a sample B that contains i members of A_{opt}, our algorithm's solution is at least as good as a solution that matches the alternatives from $B \cap A_{\mathrm{opt}}$ in the same way as Φ_{opt}, and the alternatives from $B - A_{\mathrm{opt}}$ in a random manner. Since $K_{\mathrm{opt}} = i$ and each $a_j \in A_{\mathrm{opt}}$ has equal probability of being in the sample, it is easy to see that the expected value of $\sum_{a_j \in B \cap A_{\mathrm{opt}}} \mathrm{sat}(a_j)$ is $\frac{i}{K}\mathrm{OPT}$. After we allocate the agents from $B \cap A_{\mathrm{opt}}$, each of the remaining, unassigned agents has $m - i$ positions in her preference order where she ranks the agents from $A - A_{\mathrm{opt}}$. For each unassigned agent, the average score value associated with these positions is at least $\frac{m-i-1}{2}$ (this is so, because in the worst case the agent could rank the alternatives from $B \cap A_{\mathrm{opt}}$ in the top i positions). There are $(n - i\frac{n}{K})$ agents that are not yet assigned, and so the expected total satisfaction from assigning them randomly to the alternatives is $\frac{m-i-1}{2} \cdot (n - i\frac{n}{K})$. This proves our bound on the expected satisfaction of a solution yielded by optimally matching a random sample of K alternatives.

Since OPT is upper bounded by $(m-1)n$ (consider a possibly-nonexistent solution where every agent is assigned to his or her top preference), we get that:

$$\mathbb{E}\left(\sum_{j=1}^{K} X_j \;\middle|\; K_{\mathrm{opt}} = i\right) \ge \frac{i}{K}\mathrm{OPT} + \frac{m-i-1}{2(m-1)} \cdot \left(1 - \frac{i}{K}\right)\mathrm{OPT}.$$

We can compute the unconditional expected satisfaction of Φ_B as follows:

$$\begin{aligned}\mathbb{E}\left(\sum_{j=1}^{K} X_j\right) &= \sum_{i=0}^{K} p_i \mathbb{E}\left(\sum_{j=1}^{K} X_j \;\middle|\; K_{\mathrm{opt}} = i\right) \\ &\ge \sum_{i=0}^{K} p_i \left(\frac{i}{K}\mathrm{OPT} + \frac{m-i-1}{2(m-1)} \cdot \left(1 - \frac{i}{K}\right)\mathrm{OPT}\right).\end{aligned}$$

Since $\sum_{i=1}^{K} p_i \cdot i$ is the expected value of K_{opt}, we have that $\sum_{i=1}^{K} p_i \cdot i = \frac{K^2}{m}$ (one can think of summing the expected values of K indicator random variables; one for each element of A_{opt}, taking the value 1 if a given alternative is selected and taking the value 0 otherwise). Further, from the generalized mean inequality we obtain $\sum_{i=1}^{K} p_i \cdot i^2 \ge (\frac{K^2}{m})^2$. In consequence, through routine calculation, we get that:

$$\begin{aligned}\mathbb{E}\left(\sum_{j=1}^{K} X_j\right) &\ge \left(\frac{K}{m}\mathrm{OPT} + \frac{m^2 - K^2 - m}{2m(m-1)} \cdot \left(1 - \frac{K}{m}\right)\mathrm{OPT}\right) \\ &= \frac{\mathrm{OPT}}{2}\left(1 + \frac{K}{m} - \frac{K^2}{m^2 - m} + \frac{K^3}{m^3 - m^2}\right).\end{aligned}$$

It remains to assess the probability that the total satisfaction obtained through Φ_B is close to its expected value. By the definition in the statement of the theorem, we have:

$$p_\epsilon = \Pr\left(\sum_{j=1}^{K} X_j < (1-\epsilon)\mathbb{E}\left(\sum_{j=1}^{K} X_j\right)\right).$$

We define random variable Y, which with probability p_ϵ has value $(1-\epsilon)\mathbb{E}(\sum_{j=1}^{K} X_j)$ and with probability $1 - p_\epsilon$ has value OPT. Since the value of $\sum_{j=1}^{K} X_j$ is smaller or equal to OPT with probability 1, and by the definition of p_ϵ, it must hold that $\mathbb{E}(Y) \ge \mathbb{E}(\sum_{j=1}^{K} X_j)$. After substituting the exact formula for $\mathbb{E}(Y)$ in this inequality, it becomes:

$$p_\epsilon \cdot \left((1-\epsilon)\mathbb{E}\left(\sum_{j=1}^{K} X_j\right)\right) + (1 - p_\epsilon) \cdot \mathrm{OPT} \ge \mathbb{E}\left(\sum_{j=1}^{K} X_j\right).$$

Since $\mathbb{E}(\sum_{j=1}^{K} X_j) \ge \frac{\mathrm{OPT}}{2}$, we get:

$$p_\epsilon \cdot \left((1-\epsilon)\mathbb{E}\left(\sum_{j=1}^{K} X_j\right)\right) + (1 - p_\epsilon) \cdot 2\mathbb{E}\left(\sum_{j=1}^{K} X_j\right) \ge \mathbb{E}\left(\sum_{j=1}^{K} X_j\right).$$

From which we get that $p_\epsilon \le \frac{1}{1+\epsilon}$. This completes the proof. □

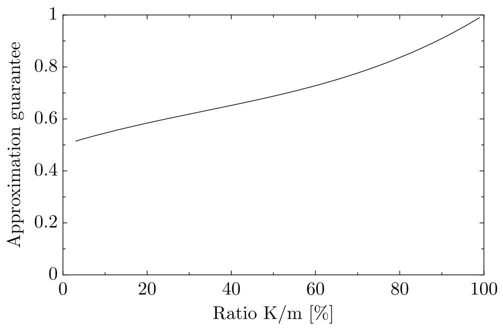


Fig. 6. The relation between the ratio K/m and the expected approximation guarantee of Algorithm R, $\frac{1}{2}(1+\frac{K}{m}-\frac{K^2}{m^2-m}+\frac{K^3}{m^3-m^2})$, for the Monroe system (as opposed to the case of Algorithm A, we do not include a figure that shows the relation between the approximation ratio and the number of candidates m for fixed values of ratio K/m because for this algorithm this is, in essence, a constant).

```
Parameters: $\lambda \leftarrow$ required probability of achieving the approximation ratio equal $0.715-\epsilon$
if $\frac{H_K}{K} \geq \frac{\epsilon}{2}$ then
    compute the optimal solution using an algorithm of Betzler et al. [9] and return.
if $m \leq 1+\frac{2}{\epsilon}$ then
    compute the optimal solution using a simple brute force algorithm and return.
$\Phi_1 \leftarrow$ solution returned by Algorithm A
$\Phi_2 \leftarrow$ run the sampling-based algorithm $-\log(1-\lambda)\cdot\frac{2+\epsilon}{\epsilon}$ times; select the assignment of the best quality
return the better assignment among $\Phi_1$ and $\Phi_2$
```

Fig. 7. Algorithm AR—combination of Algorithms A and R.

In the next theorem we show that to have a high chance of obtaining a high quality assignment, we need to repeat the sampling step many times. Thus, for practical purposes, by Algorithm R we mean an algorithm that repeats the sampling process a given number of times (this parameter is given as input) and returns the best solution found (the assignment is created using Proposition 3).

The relation between the ratio K/m and the expected approximation guarantee of Algorithm R is depicted in Fig. 6.

The threshold for $\frac{K}{m}$, where the sampling step is (in expectation) better than the greedy algorithm is about 0.57. Thus, by combining Algorithm R and Algorithm A, we can guarantee an expected approximation ratio of $0.715-\epsilon$, for each fixed constant ϵ. We give the pseudocode for the combination of the two algorithms (Algorithm AR) in Fig. 7.

Theorem 8. *For each fixed ϵ, Algorithm AR provides a $(0.715-\epsilon)$-approximate solution for the problem $\alpha_{\mathrm{B,dec}}$-Monroe-SatWinner with probability λ in time polynomial with respect to the input instance size and $-\log(1-\lambda)$.*

Proof. Let ϵ be a fixed constant. We are given an instance I of $\alpha_{\mathrm{B,dec}}$-Monroe-SatWinner.

We consider the following cases. If $m \leq 1+\frac{2}{\epsilon}$, then we solve I using a brute-force algorithm (note that in this case the number of alternatives is at most a fixed constant). Similarly, if $\frac{H_K}{K} \geq \frac{\epsilon}{2}$ then we use the exact algorithm of Betzler et al. [9] for a fixed value of K (note that in this case K is no greater than a certain fixed constant).

On the other hand, if neither of the above conditions hold (that is, if we have $m > 1+\frac{2}{\epsilon}$ and $\frac{H_K}{K} \leq \frac{\epsilon}{2}$), then we try both Algorithm A and a number of runs of the sampling-based algorithm. It is easy to check through routine calculation that if $\frac{H_K}{K} \leq \frac{\epsilon}{2}$ and $m > 1+\frac{2}{\epsilon}$ then Algorithm A achieves approximation ratio no worse than $(1-\frac{K}{2m}-\epsilon)$. We run the sampling-based algorithm $-\log(1-\lambda)\frac{2+\epsilon}{\epsilon}$ times. The probability that a single run fails to find a solution with approximation ratio at least $\frac{1}{2}(1+\frac{K}{m}-\frac{K^2}{m^2-m}+\frac{K^3}{m^3-m^2})-\frac{\epsilon}{2}$ is $p_{\frac{\epsilon}{2}} \leq \frac{2}{2+\epsilon}$. Thus, the probability that at least one run will find a solution with at least this approximation ratio is at least:

$$1-p_{\frac{\epsilon}{2}}^{-\log(1-\lambda)\frac{2+\epsilon}{\epsilon}} = 1-\left(\frac{2}{2+\epsilon}\right)^{\frac{2+\epsilon}{2}\cdot(-\log(1-\lambda))} \geq 1-\exp\bigl(\log(1-\lambda)\bigr) = \lambda.$$

Since $m \leq 1+\frac{2}{\epsilon}$, by routine calculation we see that the sampling-based algorithm with probability λ finds a solution with approximation ratio at least $\frac{1}{2}(1+\frac{K}{m}-\frac{K^2}{m^2}+\frac{K^3}{m^3})-\epsilon$. By solving the equation

$$\frac{1}{2}\left(1+\frac{K}{m}-\frac{K^2}{m^2}+\frac{K^3}{m^3}\right) = 1-\frac{K}{2m}$$

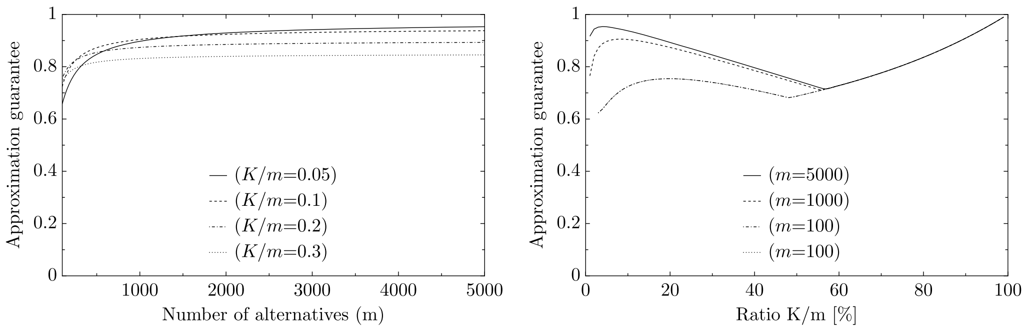


Fig. 8. The relation between the number of alternatives m (left figure) or the ratio K/m (right figure) and the approximation guarantee of Algorithm AR for the Monroe system. For small values of m the lines on right plot indicate smaller approximation guarantee than 0.715—this is because the plots take into account the factor H_K/K (see Lemma 4), which can be decreased to an arbitrarily small value, improving the quality of approximation to at least 0.715. However, doing so would require one to solve instances with small values of K using a brute-force algorithm, which may not be feasible in practice (even though theoretically possible). Thus, we chose to present the figure for the setting that one can apply in practice.

```
Notation: $\Phi_\alpha^S$—the partial assignment that assigns a single alternative to at most $\frac{n}{K}$ agents, that assigns to the agents only the
          alternatives from $S$, and that maximizes the utilitarian satisfaction $\ell_{\mathrm{sum}}^\alpha(\Phi_\alpha^S)$.
$S \leftarrow \emptyset$
for $i \leftarrow 1$ to $K$ do
    $a \leftarrow \mathrm{argmax}_{a \in A \setminus S}\ell_{\mathrm{sum}}^\alpha(\Phi_\alpha^{S \cup \{a\}})$
    $S \leftarrow S \cup \{a\}$
return $\Phi_\alpha^S$
```

Fig. 9. Pseudocode for Algorithm GM.

we can find the value of $\frac{K}{m}$ for which the two algorithms give the same approximation ratio. By substituting $x = \frac{K}{m}$ we get equality $1 + x - x^2 + x^3 = 2 - x$. One can calculate that this equality has a single solution within $\langle 0, 1 \rangle$ and that this solution is $x \approx 0.57$. For this x both algorithms guarantee approximation ratio of $0.715 - \epsilon$. For $x < 0.57$ the deterministic algorithm guarantees a better approximation ratio and for $x > 0.57$, the randomized algorithm does better. □

The relation between m, the ratio K/m, and the approximation guarantee of Algorithm AR is depicted in Fig. 8.

Let us now consider the case of CC. It is just as natural to try a sampling-based approach for solving $\alpha_{\mathrm{B,dec}}$-CC-SatWinner, as we did for the Monroe variant. Indeed, as recently (and independently) observed by Oren [45], this leads to a randomized algorithm with expected approximation ratio of $(1 - \frac{1}{K+1})(1 + \frac{1}{m})$. However, since we will later see an effective, deterministic, polynomial-time approximation scheme for $\alpha_{\mathrm{B,dec}}$-CC-SatWinner, there is little reason to explore the sampling based approach.

4.5. Algorithm GM (Monroe, CC)

Algorithm GM (greedy marginal improvement) was introduced by Lu and Boutilier for the case of the Chamberlin–Courant rule. Here we generalize it to apply to the Monroe rule as well, and we show that it is a $(1 - \frac{1}{e})$-approximation algorithm for α-Monroe-SatWinner. We point out that this approximation result for the Monroe rule applies to all non-decreasing PSFs α. For the Monroe rule, the algorithm can be viewed as an extension of Algorithm B.

The algorithm proceeds as follows. We start with an empty set S. Then we execute K iterations. In each iteration we find an alternative a that is not assigned to agents yet, and that maximizes the value $\Phi_\alpha^{S \cup \{a\}}$. (A certain disadvantage of this algorithm for the case of Monroe is that it requires a large number of computations of Φ_α^S. Since the Monroe rule assigns each alternative to at most $\frac{n}{K}$ agents in the partial assignment Φ_α^S, computing Φ_α^S is a slow process based on min-cost/max-flow algorithm.) We provide the pseudocode for Algorithm GM in Fig. 9.

We derive the approximation ratio of Algorithm GM as a corollary of a more general result regarding weighted b-matchings in bipartite graphs.

Definition 6 *(*Weighted-B-Matching *(for bipartite graphs)).* Let (U, V, E) be a bipartite undirected graph, where $U \cup V$ is the set of vertices and E is the set of edges. For each edge $\{u, v\} \in E$, by $w_{(u,v)} \in \mathbb{R}_{\geq 0}$ we denote its weight, and for each left-side vertex $u \in U$, by $s_u \in \mathbb{N}$ we denote its capacity. The goal is to find a subset of the edges with the maximal total weight, such that each vertex $u \in U$ belongs to at most s_u of the selected edges, each vertex $v \in V$ belongs to at most one of the selected edges, and altogether at most K vertices from U belong to the selected edges.

```
Notation: $\Phi \leftarrow$ a map defining a partial assignment, iteratively built by the algorithm.
          $\Phi^{\leftarrow} \leftarrow$ the set of agents for which the assignment is already defined.
          $\Phi^{\rightarrow} \leftarrow$ the set of alternatives already used in the assignment.
          $\mathrm{num\_pos}(a, x) \leftarrow \|\{i \in [n] \setminus \Phi^{\leftarrow} : pos_i(a) \le x\}\|$ (the number of not-yet assigned
          agents that rank alternative $a$ in one of their first $x$ positions)
          $\mathrm{w}(\cdot)$ denotes Lambert's W-function.
$\Phi = \{\}$
$x = \lceil \frac{m\mathrm{w}(K)}{K} \rceil$
for $i \leftarrow 1$ to $K$ do
    $a_i \leftarrow \mathrm{argmax}_{a \in A \setminus \Phi^{\rightarrow}} \mathrm{num\_pos}(a, x)$
    foreach $j \in [n] \setminus \Phi^{\leftarrow}$ do
        if $pos_j(a_i) < x$ then
            $\Phi[j] \leftarrow a_i$
foreach $j \in A \setminus \Phi^{\leftarrow}$ do
    $a \leftarrow$ such server from $\Phi^{\rightarrow}$ that $\forall_{a' \in \Phi^{\rightarrow}} pos_j(a) \le pos_j(a')$
    $\Phi[j] \leftarrow a$
```

Fig. 10. The algorithm for $\alpha_{\mathrm{B,dec}}$-CC-SatWinner (Algorithm P).

Theorem 9. *Consider a greedy algorithm for* Weighted-B-Matching *that proceeds as follows. It starts with an empty set S of edges, and in each of K consecutive steps adds to S such a vertex $u \in U$ that $\{u\} \cup S \cup V$ induces a weighted B-matching with the maximal total weight. After K steps, the algorithm returns a weighted B-matching with maximal total weight for the graph induced by vertices $S \cup V$. This algorithm is a polynomial-time $(1 - \frac{1}{e})$-approximation algorithm for* Weighted-B-Matching.

Proof. The proof is in Appendix D. □

Theorem 10. *For each non-decreasing positional scoring function α Algorithm GM is a $(1 - \frac{1}{e})$-approximation algorithm for α-*Monroe-SatWinner.

Proof. This theorem follows from Theorem 9. To show that this is the case, we reduce α-Monroe-SatWinner to Weighted-B-Matching. Consider an instance of α-Monroe-SatWinner with the set of agents $[n]$, with the set of alternatives A, and with committee size K. We form an instance of Weighted-B-Matching with bipartite graph (U, V, E) as follows:

1. We set $U = A$, i.e., the vertices on the left correspond to the candidates.
2. We set $V = [n]$, i.e., the vertices on the right correspond to the agents.
3. For every candidate $a \in A$ and every agent $i \in [n]$, we form an edge $\{a, i\}$, whose weight is equal to the satisfaction of agent i from being assigned to candidate a.
4. Each candidate a has capacity $s_a = \frac{n}{K}$.

It is easy to see that solving so constructed instance of Weighted-B-Matching corresponds to solving α-Monroe-SatWinner, and that the algorithm described in Theorem 9, for this instance, becomes Algorithm GM. □

Formally speaking, Algorithm GM is never worse than Algorithm A. For the Borda satisfaction function, it inherits the approximation guarantees from Algorithm A, and for other cases Theorem 10 guarantees approximation ratio $1 - \frac{1}{e}$ (we do not know of any guarantees for Algorithm A for these cases). The comparison with Algorithms B and C is not nearly as easy. Algorithm GM is still likely better than both of them for satisfaction functions significantly different from the Borda one, but for the Borda case our experiments show that Algorithm GM is much slower than Algorithms B and C and obtains almost the same or slightly worse results (see Section 5).

4.6. Algorithm P (CC)

Algorithm P is presented in Fig. 10. The idea of this algorithm is to compute $x = \frac{m\mathrm{w}(K)}{K}$, where w is the Lambert's W-function and to greedily compute an assignment that (approximately) maximizes the number of agents assigned to one of their top x alternatives.[5] If after this process some agent has not been assigned to any alternative, we assign her to her most preferred alternative from those already selected. Somewhat surprisingly, it turns out that this greedy strategy achieves high-quality results.

[5] This is very similar to the MaxCover problem. Skowron and Faliszewski [57] have discussed the connection of MaxCover to the winner determination problem under the Chamberlin–Courant voting system (for approval-based satisfaction functions) and provided a number of FPT approximation schemes for it.

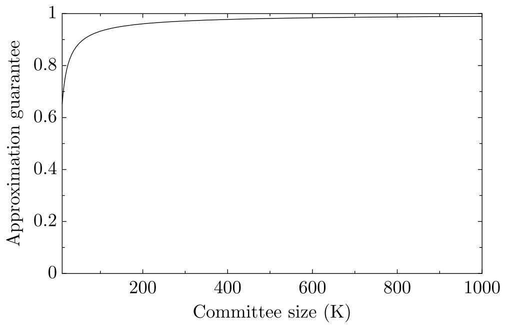


Fig. 11. The relation between the size of the committee K and the guarantee of satisfaction, $1-\frac{2\mathrm{w}(K)}{K}$, of an average voter under Algorithm P for the Chamberlin–Courant rule.

Theorem 11. *For every preference profile, Algorithm P returns a solution in which each agent is, on average, represented by an alternative that she prefers to at least* $1-\frac{2\mathrm{w}(K)}{K}$ *fraction of the alternatives.*

Proof. Let $x=\frac{m\mathrm{w}(K)}{K}$. We will first give an inductive proof that, for each i, $0 \le i \le K$, after the i-th iteration of the outer loop at most $n(1-\frac{\mathrm{w}(K)}{K})^i$ agents are unassigned. Based on this observation, we will derive the approximation ratio of our algorithm.

For $i=0$, the inductive hypothesis holds because $n(1-\frac{\mathrm{w}(K)}{K})^0=n$. For each i, let n_i denote the number of unassigned agents after the i-th iteration. Thus, after the i-th iteration there are n_i unassigned agents, each with x unassigned alternatives among her top x ranked alternatives. As a result, at least one unassigned alternative is present in at least $\frac{n_i x}{m-i}$ of the top x positions of unassigned agents. This means that after the $(i+1)$-th iteration the number of unassigned agents is:

$$n_{i+1} \le n_i - \frac{n_i x}{m-i} \le n_i\left(1-\frac{x}{m}\right) = n_i\left(1-\frac{\mathrm{w}(K)}{K}\right).$$

If for a given i the inductive hypothesis holds, that is, if $n_i \le n(1-\frac{\mathrm{w}(K)}{K})^i$, then:

$$n_{i+1} \le n\left(1-\frac{\mathrm{w}(K)}{K}\right)^i\left(1-\frac{\mathrm{w}(K)}{K}\right) = n\left(1-\frac{\mathrm{w}(K)}{K}\right)^{i+1}.$$

Thus the hypothesis holds and, as a result, we have that:

$$n_K \le n\left(1-\frac{\mathrm{w}(K)}{K}\right)^K \le n\left(\frac{1}{e}\right)^{\mathrm{w}(K)} = \frac{n\mathrm{w}(K)}{K}.$$

Let Φ be the assignment computed by our algorithm. To compare it against the optimal solution, it suffices to observe that the optimal solution has the value of satisfaction of at most $\mathrm{OPT} \le (m-1)n$, that each agent selected during the first K steps has satisfaction at least $m-x=m-\frac{m\mathrm{w}(K)}{K}$, and that the agents not assigned within the first K steps have satisfaction no worse than 0. Thus it holds that:

$$\begin{aligned}\ell_{\mathrm{sum}}^{\alpha_{\mathrm{B,dec}}}(\Phi) &\ge \left(n-\frac{n\mathrm{w}(K)}{K}\right)\left(m-\frac{m\mathrm{w}(K)}{K}\right)\\ &\ge n\cdot m\cdot\left(1-\frac{\mathrm{w}(K)}{K}\right)\left(1-\frac{\mathrm{w}(K)}{K}\right) \ge nm\left(1-\frac{2\mathrm{w}(K)}{K}\right).\end{aligned}$$

This completes the proof. □

The relation between K, and the guarantee of satisfaction of an average voter is given in Fig. 11.

Corollary 12. *Algorithm P is a polynomial-time* $(1-\frac{2\mathrm{w}(K)}{K})$*-approximation algorithm for* $\alpha_{\mathrm{B,dec}}$-CC-SatWinner.

Proof. Since a satisfaction of each single agent is bounded by $(m-1)$, by applying Theorem 11, we get the conclusion. □

Since for each $\epsilon > 0$ there is a value K_ϵ such that for each $K > K_\epsilon$ it holds that $\frac{2\mathrm{w}(K)}{K} < \epsilon$, and $\alpha_{\mathrm{B,dec}}$-CC-SatWinner problem can be solved optimally in polynomial time for each fixed constant K (see the XP algorithm for parameter K given by Betzler et al. [9]), there is a polynomial-time approximation scheme (PTAS) for $\alpha_{\mathrm{B,dec}}$-CC-SatWinner (i.e., a family of algorithms such that for each fixed r, $0 < r < 1$, there is a polynomial-time r-approximation algorithm for $\alpha_{\mathrm{B,dec}}$-CC-SatWinner

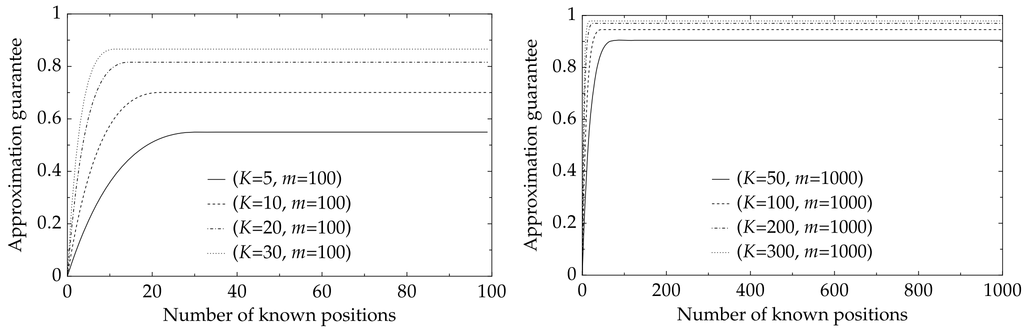


Fig. 12. The relation between the number of the known positions and the approximation ratio of Algorithm P for $\alpha_{\mathrm{B,dec}}$-CC-SatWinner.

in the family; note that in PTASes we measure the running time by considering r to be a fixed constant). The reason is that given the approximation parameter ϵ and an instance of the problem with a particular value of K, we can solve it as follows:

1. If $K \leq K_\epsilon$, then we use the brute-force algorithm of Betzler et al. [9] to solve the problem optimally (since K_ϵ is a constant, the algorithm runs in polynomial time).[6]
2. Otherwise, we use Algorithm P (whose approximation ratio is sufficiently good when $K \geq K_\epsilon$).

Corollary 13. *There is a PTAS for* $\alpha_{\mathrm{B,dec}}$-CC-SatWinner.

Finally, we show that Algorithm P performs very well even if the voters cast truncated ballots. Proposition 14 gives the relation between the number Q of positions used by the algorithm and the approximation ratio. In Fig. 12 we show this relation for some values of the parameters m and K.

Proposition 14. *Algorithm P that uses $x = Q$ instead of $x = \lceil \frac{m\mathrm{w}(K)}{K} \rceil$, where $Q \leq \frac{m\mathrm{w}(K)}{K}$, returns a solution in which each agent is, on average, represented by an alternative that she prefers to at least $\frac{m-Q}{m-1}(1 - e^{-\frac{QK}{m}})$ fraction of the alternatives. In particular, such algorithm is a polynomial-time $(\frac{m-Q}{m-1}(1 - e^{-\frac{QK}{m}}))$-approximation algorithm for* $\alpha_{\mathrm{B,dec}}$-CC-SatWinner.

Proof. The proof is in Appendix D. □

For example, for Polish parliamentary elections ($K = 460$, $m = 6000$), it suffices that each voter ranks only 0.5% of her top alternatives (that is, about 30 alternatives) for the algorithm to find a solution with guaranteed satisfaction at least 90% of the maximal satisfaction (possibly not achievable) when every voter is assigned to her top alternative.

5. Empirical evaluation of the algorithms

In this section we present the results of empirical evaluation of algorithms from Section 4. In the experiments we consider variants of the randomized algorithms that use exactly 100 sampling steps. In all cases but one, we use Borda DPSF to measure voter satisfaction. In one case, with six candidates, we use PSF defined through vector $(3, 3, 3, 2, 1, 0)$ (we made this choice due to the nature of the data set; see discussion later).

We have conducted four sets of experiments. First, we have tested all our algorithms on relatively small elections (up to 10 candidates, up to 100 agents). In this case we were able to compare the solutions provided by our algorithms with the optimal ones. (To obtain the optimal solutions, we were using the ILP formulations, see Appendix B, and the GLPK's ILP solver.) Thus we report the quality of our algorithms as the average of fractions C/C_{opt}, where C is the satisfaction obtained by a respective algorithm and C_{opt} is the satisfaction in the optimal solution. For each algorithm and data set, we also report the average fraction C/C_{ideal}, where C_{ideal} is the satisfaction that the voters would have obtained if each of them were matched to her most preferred alternative. In our further experiments, where we considered larger elections, we were not able to compute optimal solutions, but the fraction C/C_{ideal} gives a lower bound for C/C_{opt}. We report this value for small elections so that we can see an example of the relation between C/C_{opt} and C/C_{ideal} and so that we can compare the results for small elections with the results for the larger ones. Further, for the case of Borda PSF the C/C_{ideal} fraction has a very natural interpretation: If its value for a given solution is v, then, on the average, in this solution each voter is matched to an alternative that she prefers to $(m-1)v$ alternatives.

[6] The exact polynomial depends on ϵ and this is why the algorithm is not a *fully polynomial-time approximation scheme*.

In our second set of experiments, we have run our algorithms on large elections (thousands of agents, hundreds of alternatives), coming either from the NetFlix data set (see below) or generated by us using one of our models. Here we report the average fraction C/C_{ideal} only. We analyze the quality of the solutions as a function of the number of agents, the number of candidates, and the relative number of winners (fraction K/m). (This last set of results is particularly interesting because in addition to measuring the quality of our algorithms, it allows one to assess the size of a committee one should seek if one wants to achieve a given average satisfaction of agents.)

In the third set of experiments, we investigate the effect of submitting truncated ballots (i.e., preference orders where only some of the top alternatives are ranked). Specifically, we focus on the relation between the fraction of ranked alternatives and the approximation ratios of the algorithms. We ran our experiments on relatively large instances describing agents' preferences; thus, here as in the previous set of experiments, we used NetFlix data set and the synthetic data sets. We report the quality of the algorithms as the ratio C/C_{ideal}.

In the fourth set of experiments we have measured running times of our algorithms and of the ILP solver. Even though all our algorithms (except for the ILP based ones) are polynomial-time, in practice some of them are too slow to be useful.

5.1. Experimental data

For the evaluation of the algorithms we consider both real-life preference-aggregation data and synthetic data, generated according to a number of election models. The experiments reported in this paper predate the work of Mattei and Walsh [37] on gathering a large collection of data sets with preference data, but we mention that the conference version of this paper contributed several data sets to their collection.

5.1.1. Real-life data

We use real-life data regarding people's preference on sushi types, movies, college courses, and performance of athletes in figure-skating competitions. One of the major problems regarding real-life preference data is that either people express preferences over a very limited set of alternatives, or their preference orders are partial. To address the latter issue, for each such data set we extended the partial orders to total orders using the technique of Kamishima [27]. (The idea is to complete each preference order based on those reported preference orders that appear to be similar.)

Some of our data sets contain a single profile, whereas the others contain multiple profiles. When preparing data for a given number m of candidates and a given number n of voters from a given data set, we used the following method: We first uniformly at random chose a profile within the data set, and then we randomly selected n voters and m candidates. We used preference orders of these n voters restricted to these m candidates.

There are two important points that we should make about our real-life data sets and the way we use them. First, one should keep in mind that the method of extending partial preference orders to total ones can possibly have significant influence on the results. Here we used the Kamishima [27] technique, but other ones are possible and might give different results. Second, in our experiments on real-life data, for each experiment we redraw a sample from a given data set. This means that our samples are not independent and our results should be seen as "noisy observations about the same event," where the event corresponds to the type of preferences in a given data set. We believe that such results are still useful and do give some information on how our algorithms would behave in practice. Nonetheless, one should keep in mind their limitations. We point the reader to Regenwetter et al.'s book [50] for a detailed discussion of various issues in treating empirical data in social choice, including the two issues just mentioned.[7]

Sushi Preferences. We used the set of preferences regarding sushi types collected by Kamishima [27].[8] Kamishima has collected two sets of preferences, which we call S1 and S2. Data set S1 contains complete rankings of 10 alternatives collected from 5000 people. S2 contains partial rankings provided by 5000 people over a set of 100 alternatives (each "vote" ranks 10 alternatives). We used Kamishima's [27] technique to obtain total rankings.

Movie Preferences. Following Mattei et al. [35,36], we have used the NetFlix data set[9] of movie preferences (we call it Mv). NetFlix data set contains ratings collected from about 480 thousand distinct users regarding 18 thousand movies. The users rated movies by giving them a score between 1 (bad) and 5 (good). The set contains about 100 million ratings. We have generated 50 profiles using the following method: For each profile we have randomly selected 300 movies, picked 10 000 users that ranked the highest number of the selected movies, and for each user we have extended her ratings to a complete preference order using the method of Kamishima [27]. (However, we should mention that Popova et al. [46] presented a detailed discussion showing that the exact results—in their case for single-winner voting rules—that one could derive from the NetFlix data set vary strongly depending on the assumptions one makes about extending the ratings to complete preference profiles. Thus our results about this data set should be understood only as examples of the type of results one should expect.)

[7] We are very grateful to the anonymous reviewers for pointing out these issues.

[8] The sushi data set is available under the following url: http://www.kamishima.net/sushi/.

[9] http://www.netflixprize.com/.

Course Preferences. Each year the computer science students at the AGH University in Krakow choose courses that they would like to attend. In our case, the students were offered a choice of six courses of which they had to attend three. Thus the students were asked to give an unordered set of their three most preferred courses and a ranking of the remaining ones (in case too many students selected a particular course, those with the highest GPA were enrolled and the remaining ones were moved to their less preferred courses). In this data set, which we call Cr, we have 120 voters (students) and 6 alternatives (courses). However, due to the nature of the data, instead of using Borda count PSF as the satisfaction measure, we use the vector $(3, 3, 3, 2, 1, 0)$. Currently this data set is available as part of PrefLib [37].

Figure Skating. This data set, which we call Sk, contains preferences of the judges over performances in figure-skating competitions. The data set contains 48 profiles, each describing a single competition. Each profile contains preference orders of 9 judges over about 20 participants. The competitions include European skating championships, Olympic Games, World Junior, and World Championships, all from 1998.[10] (Note that while in figure skating judges provide numerical scores, this data set is preprocessed to contain preference orders.)

Most of the data sets that we used are currently available as part of the PrefLib [37] library of preference data.

5.1.2. Synthetic data

For our tests, we also use profiles generated using three well-known distributions of preference orders.

Impartial Culture. Under the impartial culture model of preferences (which we denote IC), for a given set A of alternatives, each voter's preference order is drawn uniformly at random from the set of all possible total orders over A. While not very realistic, profiles generated using impartial culture model are a standard testbed of election-related algorithms.

Polya–Eggenberger Urn Model. Following McCabe-Dansted and Slinko [39] and Walsh [60], we use the Polya–Eggenberger urn model [6] (which we denote Ur). In this model we generate votes as follows. We have a set A of m alternatives and an urn that initially contains a copy of each of the $m!$ preference orders over A. To generate a vote, we simply randomly pick one from the urn (this is our generated vote), and then—to simulate correlation between voters—we return a copies of this preference order to the urn. When generating an election with m candidates using the urn model, we set the parameter a so that $\frac{a}{m!} = 0.05$. (Both McCabe-Dansted and Slinko [39] and Walsh [60] refer to the value $\frac{a}{m!}$ as parameter b; we mention that those authors use much higher values of b but we felt that too high a value leads to a much too strong correlation between votes.)

Generalized Mallow's Model. We refer to this data set as Ml. Let $\succ$ and $\succ'$ be two preference orders over some alternative set A. Kendal–Tau distance between $\succ$ and $\succ'$, denoted $d_K(\succ, \succ')$, is defined as the number of pairs of candidates $x, y \in A$ such that either $(x \succ y) \wedge (y \succ' x)$ or $(y \succ x) \wedge (x \succ' y)$.

Under Mallow's distribution of preferences [34] we are given two parameters: a *center* preference order $\succ$ and a number ϕ between 0 and 1. The model says that the probability of generating preference order $\succ'$ is proportional to the value $\phi^{d_K(\succ,\succ')}$ To generate preference orders following Mallow's distribution, we use the algorithm given by Lu and Boutilier [33].

In our experiments, we use a mixture of Mallow's models. Let A be a set of alternatives and let ℓ be a positive integer. This mixture model is parametrized by three vectors, $\Lambda = (\lambda_1, \dots, \lambda_\ell)$ (where each λ_i is between 0 and 1, and $\sum_{i=1}^{\ell} \lambda_1 = 1$), $\Phi = (\phi_1, \dots, \phi_\ell)$ (where each ϕ_i is a number between 0 and 1), and $\Pi = (\succ_1, \dots, \succ_\ell)$ (where each $\succ_i$ is a preference order over A). To generate a vote, we pick a random integer i, $1 \le i \le \ell$ (each i is chosen with probability λ_i), and then generate the vote using Mallow's model with parameters $(\succ_i, \phi_i)$.

For our experiments, we use $a = 5$, and generate vectors Λ, Φ, and Π uniformly at random.

5.2. Evaluation on small instances

We now present the results of our experiments on small elections. For each data set, we generated elections with the number of agents $n = 100$ ($n = 9$ for data set Sk because there are only 9 voters there) and with the number of alternatives $m = 10$ ($m = 6$ for data set Cr because there are only 6 alternatives there) using the method described in Section 5.1.1 for the real-life data sets, and in the natural obvious way for synthetic data. For each algorithm and for each data set we ran 500 experiments on different instances for $K = 3$ (for the Cr data set we used $K = 2$) and 500 experiments for $K = 6$ (for Cr we set $K = 4$). For Algorithm C (both for Monroe and for CC) we set the parameter d, describing the number of assignment functions computed in parallel, to 15. The results (average fractions C/C_{opt} and C/C_{ideal}) for $K = 3$ are given in Tables 1 and 3; the results for $K = 6$ are given in Tables 2 and 4 (they are almost identical as for $K = 3$). For each experiment in this section we also computed the standard deviation; it was always on the order of 0.01. We also computed the minimum satisfaction values returned by our algorithms in each of the experiments. For all the algorithms except Algorithm P, these values were never lower than 90% of the reported average values (for Algorithm P, they were never lower than 85% of the reported average values). These results lead to the following conclusions:

[10] This data set is available under the following url: http://rangevoting.org/SkateData1998.txt.

Table 1
The average quality of the algorithms compared with the optimal solution (C/C_{opt}) for the small instances of data and for $K = 3$ ($K = 2$ for Cr); $m = 10$ ($m = 6$ for Cr); $n = 100$ ($n = 9$ for Sk).

	Monroe					CC			
	A	B	C	GM	R	C	GM	P	R
S1	0.94	0.99	≈ 1.0	0.99	0.99	1.0	≈ 1.0	0.99	0.99
S2	0.95	0.99	1.0	≈ 1.0	0.99	1.0	≈ 1.0	0.98	0.99
Mv	0.96	≈ 1.0	1.0	≈ 1.0	0.98	1.0	≈ 1.0	0.96	≈ 1.0
Cr	0.98	0.99	1.0	≈ 1.0	0.99	1.0	≈ 1.0	1.0	≈ 1.0
Sk	0.99	≈ 1.0	1.0	≈ 1.0	0.94	1.0	≈ 1.0	0.82	0.99
IC	0.94	0.99	≈ 1.0	0.99	0.99	1.0	≈ 1.0	0.99	0.99
Ml	0.94	0.99	1.0	0.99	0.99	1.0	≈ 1.0	0.99	0.99
Ur	0.95	0.99	≈ 1.0	0.99	0.99	1.0	0.99	0.97	0.99

Table 2
The average quality of the algorithms compared with the optimal solution (C/C_{opt}) for the small instances of data and for $K = 6$ ($K = 4$ for Cr); $m = 10$ ($m = 6$ for Cr); $n = 100$ ($n = 9$ for Sk).

	Monroe					CC			
	A	B	C	GM	R	C	GM	P	R
S1	0.95	≈ 1.0	1.0	0.99	0.99	1.0	≈ 1.0	0.97	0.99
S2	0.94	0.99	≈ 1.0	0.99	0.99	1.0	≈ 1.0	0.98	≈ 1.0
Mv	0.95	0.99	1.0	≈ 1.0	0.98	1.0	≈ 1.0	0.97	≈ 1.0
Cr	0.96	≈ 1.0	1.0	≈ 1.0	0.99	1.0	1.0	1.0	1.0
Sk	0.99	≈ 1.0	1.0	≈ 1.0	0.88	1.0	1.0	0.91	≈ 1.0
IC	0.95	0.99	≈ 1.0	0.99	0.99	1.0	≈ 1.0	0.99	0.99
Ml	0.95	0.99	≈ 1.0	0.99	0.99	1.0	≈ 1.0	0.98	0.99
Ur	0.96	0.99	≈ 1.0	0.99	≈ 1.0	1.0	≈ 1.0	0.96	0.99

Table 3
The average quality of the algorithms compared with the simple lower bound (C/C_{ideal}) for the small instances of data and for $K = 3$ ($K = 2$ for Cr); $m = 10$ ($m = 6$ for Cr); $n = 100$ ($n = 9$ for Sk).

	Monroe					CC			
	A	B	C	GM	R	C	GM	P	R
S1	0.85	0.89	0.9	0.89	0.89	0.92	0.92	0.91	0.92
S2	0.85	0.89	0.89	0.89	0.89	0.93	0.93	0.91	0.92
Mv	0.88	0.92	0.92	0.92	0.91	0.97	0.97	0.93	0.97
Cr	0.94	0.96	0.97	0.96	0.96	0.97	0.97	0.97	0.97
Sk	0.96	0.96	0.97	0.97	0.91	1.0	1.0	0.82	0.99
IC	0.8	0.84	0.85	0.84	0.84	0.85	0.85	0.84	0.85
Ml	0.83	0.88	0.88	0.88	0.88	0.92	0.9	0.89	0.9
Ur	0.8	0.85	0.86	0.85	0.85	0.9	0.89	0.87	0.89

Table 4
The average quality of the algorithms compared with the simple lower bound (C/C_{ideal}) for the small instances of data and for $K = 6$ ($K = 4$ for Cr); $m = 10$ ($m = 6$ for Cr); $n = 100$ ($n = 9$ for Sk).

	Monroe					CC			
	A	B	C	GM	R	C	GM	P	R
S1	0.91	0.96	0.96	0.95	0.95	0.98	0.98	0.96	0.98
S2	0.88	0.93	0.93	0.93	0.93	0.98	0.98	0.96	0.98
Mv	0.85	0.89	0.89	0.89	0.88	0.99	0.99	0.97	0.99
Cr	0.95	0.98	0.99	0.99	0.98	1.0	1.0	1.0	1.0
Sk	0.91	0.92	0.92	0.92	0.81	1.0	1.0	0.91	≈ 1.0
IC	0.91	0.95	0.95	0.94	0.95	0.96	0.96	0.95	0.95
Ml	0.89	0.94	0.94	0.94	0.93	0.98	0.98	0.95	0.97
Ur	0.91	0.95	0.95	0.94	0.95	0.98	0.98	0.94	0.97

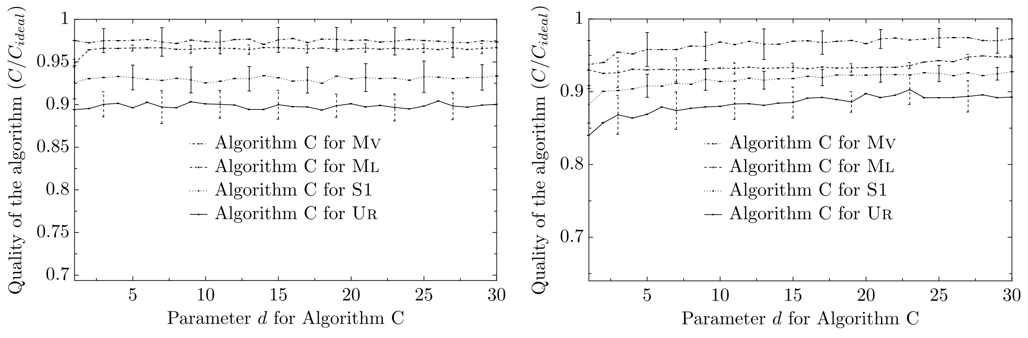


Fig. 13. The relation between the value of the parameter d for Algorithm C and the quality of the algorithms C/C_{ideal} for the Chamberlin–Courant system (left figure) and for the Monroe system (right figure); $m = 10$; $K = 3$; $n = 100$.

1. For the case of Monroe, Algorithm A obtains very good results, but nonetheless Algorithms B and C improve noticeably upon Algorithm A. In particular, Algorithm C (for $d = 15$) obtained the highest satisfaction on all data sets and in almost all cases was able to find an optimal solution.
2. Both for Monroe and for CC, Algorithm R gives slightly worse solutions than Algorithm C.
3. The results do not seem to depend on the data sets used in the experiments (the only exception is Algorithm R for the Monroe system on data set Sk; however Sk has only 9 voters so we view it as a border case).

Since Algorithm C achieves very good results, it is natural to ask for the impact of the value of the parameter d on the quality of approximation. We present this relation for $n = 100$, $m = 10$, and $K = 3$ in Fig. 13. We see that for the case of Chamberlin–Courant system even a small value of d is sufficient to obtain algorithm's optimal behavior. For the case of the Monroe system, slightly larger values of d lead to optimal behavior. In the next section we show that for the larger values of m Algorithm A gives high quality results by itself, and thus the impact of the value of d is even less pronounced. For this reason we omit presentation of the analogous results regarding parameter d for larger values of m.

5.3. Evaluation on larger instances

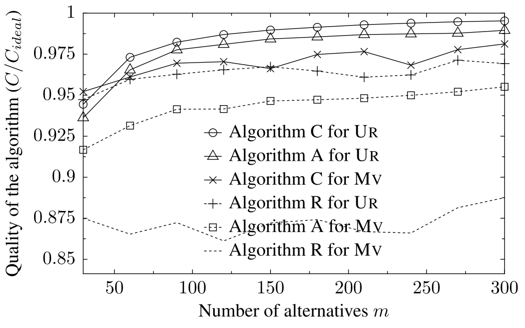
For experiments on larger instances we needed data sets with at least $n = 10\,000$ agents. Thus we used the NetFlix data set and synthetic data. (Additionally, we ran the subset of experiments (for $n \le 5000$) also for the S2 data set.) For the Monroe rule we present results for Algorithms A, C, and R, and for the Chamberlin–Courant rule we present results for Algorithms C and R. We limit the set of algorithms for the sake of clarity of presentation. For Monroe we chose Algorithm A because it is the simplest and the fastest one, Algorithm C because it is the best generalization of Algorithm A that we were able to run in reasonable time, and Algorithm R to compare a randomized algorithm to deterministic ones. For the Chamberlin–Courant rule we chose Algorithm C because it is, intuitively, the best one, and we chose Algorithm R for the same reason as in the case of Monroe.

First, for each data set and for each algorithm we fixed the value of m and K and for each n ranging from 1000 to 10 000 with the step of 1000 we ran 50 experiments. We repeated this procedure for 4 different combinations of m and K: ($m = 10$, $K = 3$), ($m = 10$, $K = 6$), ($m = 100$, $K = 30$) and ($m = 100$, $K = 60$). We measured the statistical correlation between the number of voters and the quality of the algorithms C/C_{ideal}. The ANOVA test in most cases showed that there is no such correlation. The only exception was S2 data set, for which we obtained an almost negligible correlation. For example, for ($m = 10$, $K = 3$) Algorithm C under data set S2 for the Monroe system for $n = 5000$ gave $C/C_{\text{ideal}} = 0.88$, while for $n = 100$ (in the previous section) we got $C/C_{\text{ideal}} = 0.89$. Thus we conclude that in practice the number of agents has almost no influence on the quality of the results provided by our algorithms.

Next, we fixed the number of voters $n = 1000$ and the ratio $K/m = 0.3$, and for each m ranging from 30 to 300 with step size 30 (naturally, as m changed, so did K to maintain the ratio K/m), we ran 50 experiments. We repeated this procedure for $K/m = 0.6$. The relation between m and C/C_{ideal} for Mv and Ur, under both the Monroe rule and the Chamberlin–Courant rule, is given in Figs. 14 and 15 (the results for $K/m = 0.6$ look similar).

Finally, we fixed $n = 1000$ and $m = 100$, and for each K/m ranging from 0.1 and 0.5 with the step of 0.1 we ran 50 experiments. The relation between the ratio K/m and the quality C/C_{ideal} is presented in Figs. 16 and 17.

For the case of Chamberlin–Courant system, increasing the size of the committee to be elected improves overall agents' satisfaction. Indeed, since there are no constraints on the number of agents matched to a given alternative, a larger committee means more opportunities to satisfy the agents. For the Monroe rule, a larger committee may lead to a lower total satisfaction. This happens if many agents like a particular alternative a lot, but only some of them can be matched to this alternative and others have to be matched to their less preferred ones. Indeed, this observation is—to some extent—supported by our theoretical results: For example, if we fix the number of candidates m, the approximation guarantee of Algorithm P (for Chamberlin–Courant; see Fig. 11) is monotonically increasing with K, whereas this is not the case for the approximation



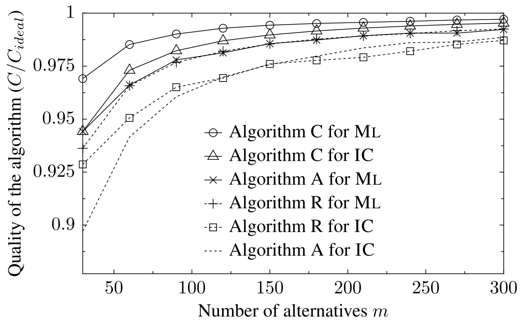
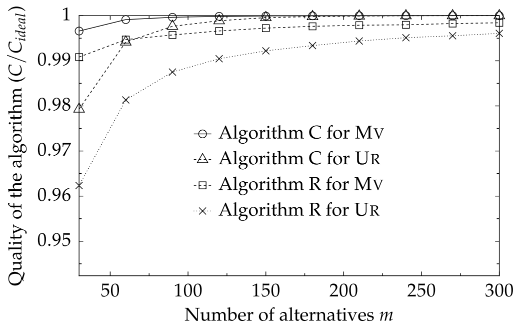


Fig. 14. The relation between the number of alternatives m and the quality of the algorithms C/C_{ideal} for the Monroe system; $K/m = 0.3$; $n = 1000$.



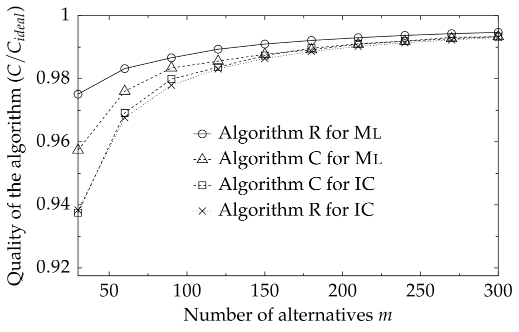
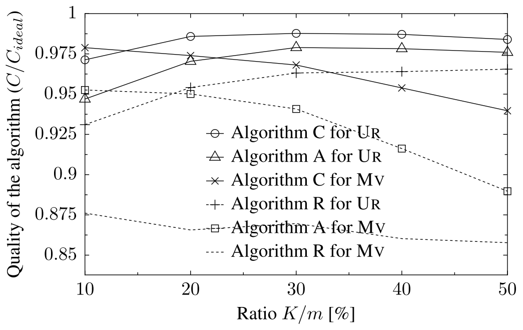


Fig. 15. The relation between the number of alternatives m and the quality of the algorithms C/C_{ideal} for the Chamberlin–Courant system; $K/m = 0.3$; $n = 1000$.



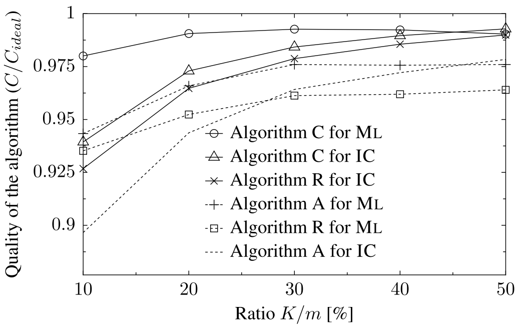
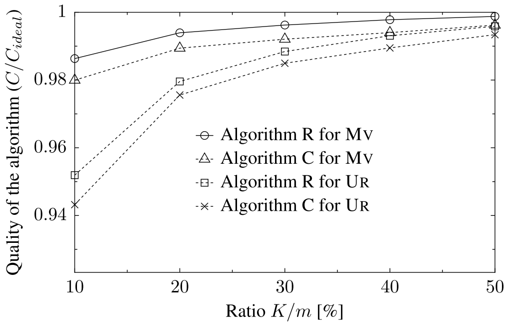


Fig. 16. The relation between the ratio K/m and the quality of the algorithms C/C_{ideal} for the Monroe system; $m = 100$; $n = 1000$.



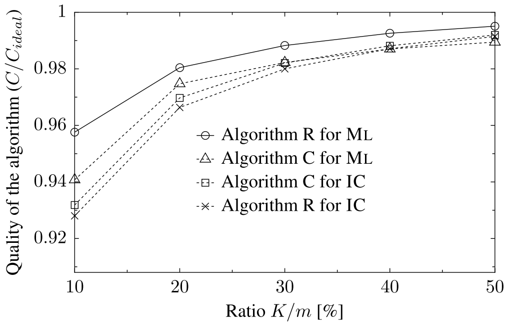


Fig. 17. The relation between the ratio K/m and the quality of the algorithms C/C_{ideal} for the Chamberlin–Courant system; $m = 100$; $n = 1000$.

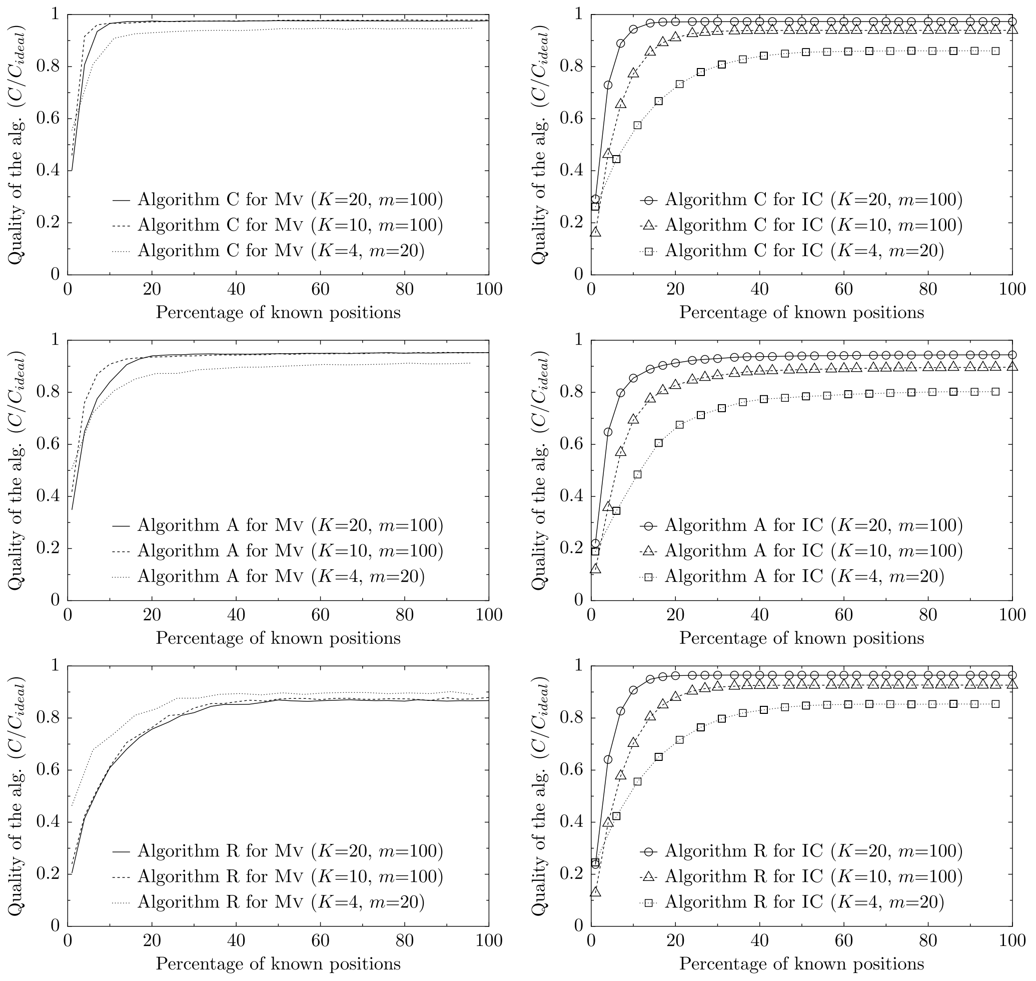


Fig. 18. The relation between the number of known positions P and the quality of the algorithm C/C_{ideal} for Algorithms C, A, and R for the Monroe system. Each row of the plots describes one algorithm; each column describes one data set; $n = 1000$.

guarantee of Algorithm A (for Monroe; see the right-hand side of Fig. 3). Nonetheless, we see that for Monroe Algorithm C achieves $C/C_{\text{ideal}} = 0.925$ even for $K/m = 0.5$ for the NetFlix data set.

Our conclusions from these experiments are the following. For the Monroe rule, even Algorithm A achieves very good results. However, Algorithm C consistently achieves better (indeed, almost perfect) ones. For the Chamberlin–Courant rule the randomized algorithm on some datasets performs better than the deterministic ones. However, even in such cases, the improvement over Algorithm C is small.

5.4. Truncated ballots

The purpose of our third set of experiments is to see how our algorithms behave in practical settings with truncated ballots. We conducted this part of evaluation on relatively large instances, including $n = 1000$ agents and up to $m = 100$ alternatives. Thus, in this set of experiments, we used the same sets of data as in the previous subsection: the NetFlix data set and the synthetic distributions. Similarly, we evaluated the same algorithms: Algorithms A, C, and R for the case of the Monroe system, and Algorithms C and R for the case of the Chamberlin–Courant system.

For each data set and for each algorithm we ran experiments for three independent settings with different values of the parameters describing the elections: (1) $m = 100$, $K = 20$, (2) $m = 100$, $K = 10$, and (3) $m = 20$, $K = 4$. For each setting we ran the experiments for the values of P (the number of known positions) varying between 1 and m.

For each algorithm, data set, setting and each value of P we ran 50 independent experiments in the following way. From a data set we sampled a sub-profile of the appropriate size $n \times m$. We truncated this profile to the P first positions. We ran the algorithm for the truncated profile and calculated the quality ratio C/C_{ideal}. When calculating C/C_{ideal} we assumed the worst case scenario, i.e., that the satisfaction of each agent from an alternative outside of her first P positions is equal to 0. In other words, we used the positional scoring function described by the following vector: $\langle m-1, m-2, \dots, m-P, 0, \dots, 0\rangle$. Next, we averaged the values of C/C_{ideal} over all 50 experiments.

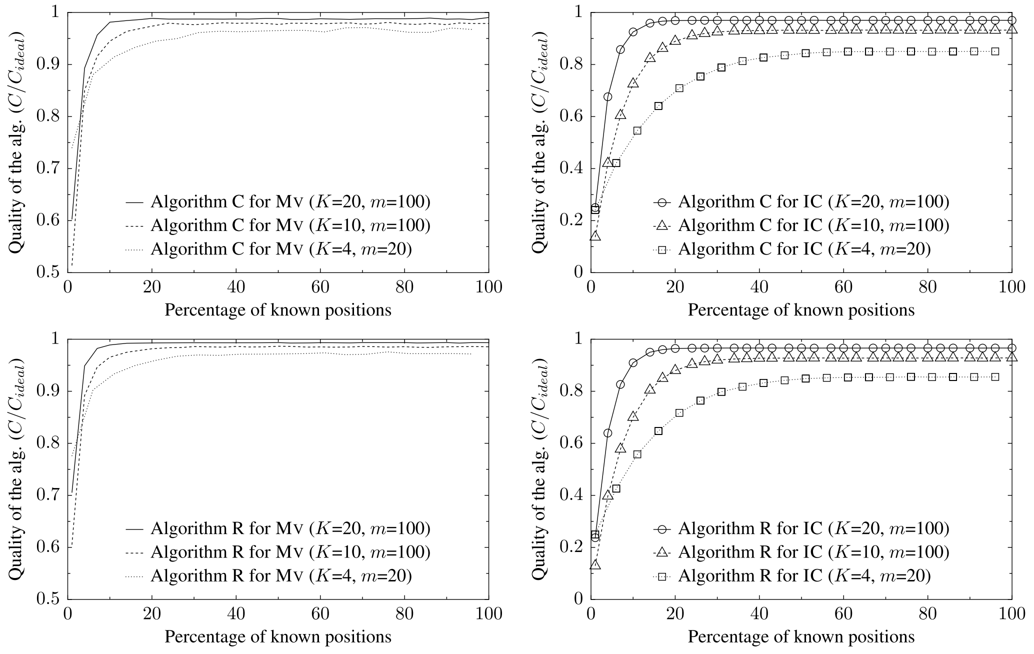


Fig. 19. The relation between the number of known positions P and the quality of the algorithm C/C_{ideal} for Algorithms C and R for the Chamberlin–Courant system. Each row of the plots describes one algorithm; each column describes one data set; $n = 1000$. (Results for the Mallows model are similar to those for the urn model and are omitted for clarity.)

The relation between the percentage of the known positions in the preference profile and the average quality of the algorithm for the Monroe and Chamberlin–Courant systems are plotted in Figs. 18 and 19, respectively. We omit the plots for the Urn model and for the Mallow's model, as in this case the results are very similar. We have the following conclusions.

1. All the algorithms require only small number of the top positions to achieve their best quality. Here, the deterministic algorithms are superior.
2. The small elections with synthetic distributions appear to be the worst case scenario (we also run our experiments for Urn model and for Mallow's model: for these two models our algorithms gave slightly better results than for the impartial culture model)—in such cases we require the knowledge of about 40% of the top positions to obtain the highest approximation ratios of the algorithms. In the case of the NetFlix data set, even on small instances the deterministic algorithms require only about 8% of the top positions to get their best quality (however the quality is already high for 3–5% of the top positions). For the larger number of the alternatives, the algorithms do not require more than 3% of the top positions to reach their top results.
3. Algorithm C gives the best solutions and is also the most immune one to the lack of knowledge. These results are more evident for the case of the Monroe system.

5.5. *Running time*

In our final set of experiments, we measured running times of our algorithms on the data set Mv. We used a machine with Intel Pentium Dual T2310 1.46 GHz processor and 1.5 GB of RAM. In Fig. 20 we show the running times of the GLPK ILP solver for the Monroe and for Chamberlin–Courant rules. These running times are already large for small instances and they are increasing exponentially with the number of voters. For the Monroe rule, even for $K = 9$, $m = 30$, $n = 100$ some of the experiments timed out after 1 hour, and for $K = 9$, $m = 30$, $n = 200$ none of the experiments finished within one day. Thus we conclude that the real application of the ILP-based algorithm is very limited.

Example running times of the other algorithms for some combinations of n, m, and K are presented in Table 5. For the case of CC, essentially all the algorithms are very fast and the quality of computed solutions is the main criterion in choosing among them. For the case of Monroe, the situation is more complicated. While for small elections all the algorithms are practical, for elections with thousands of voters, using Algorithm GM becomes problematic. Indeed, even Algorithm C can be seen as a bit too slow if one expects immediate results. On the other hand, Algorithms A and B seem perfectly practical and, as we have seen in the previous experiments, give high-quality results.

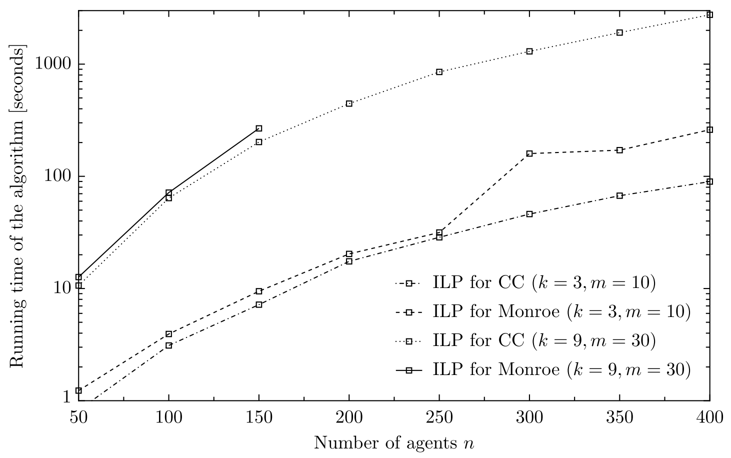


Fig. 20. The running time of the standard ILP solver for the Monroe and for the Chamberlin–Courant systems. For the Monroe system, for $K = 9, m = 30$, and for $n \geq 200$ none of the single algorithm execution finished within 1 day.

Table 5
Example running times of the algorithms [in seconds].

		$m = 10,\ K = 3$			$m = 10,\ K = 6$		
	$n =$	2000	6000	10 000	2000	6000	10 000
Monroe	A	0.01	0.03	0.05	0.01	0.04	0.07
	B	0.08	0.9	2.3	0.2	1.4	3.6
	C	1.1	8	22	2.1	16	37
	GM	0.8	7.3	20	1.9	13	52
	R	7.6	50	180	6.5	52	140
CC	C	0.02	0.07	0.12	0.05	0.14	0.26
	GM	0.003	0.009	0.015	0.003	0.01	0.018
	P	0.009	0.032	0.05	0.008	0.02	0.05
	R	0.014	0.04	0.065	0.02	0.06	0.11

		$m = 100,\ K = 30$			$m = 100,\ K = 60$		
	$n =$	2000	6000	10 000	2000	6000	10 000
Monroe	A	0.5	1.6	2.8	0.9	2.8	4.9
	B	0.8	4	9.5	1.7	8	18
	C	38	140	299	64	221	419
	GM	343	2172	5313	929	5107	13420
	R	41	329	830	88	608	1661
CC	C	4.3	11	19	7.5	19	31
	GM	0.06	0.2	0.4	0.09	0.3	0.7
	P	0.03	0.1	0.26	0.03	0.1	0.2
	R	0.06	0.24	0.45	0.1	0.4	0.8

6. Summary

We have studied approximability of finding winners under the Monroe and Chamberlin–Courant multiwinner voting rules. We have shown that if we try to optimize agents' dissatisfaction, then our problems are hard to approximate up to any constant factor. The same holds for the case where we focus on the satisfaction of the least satisfied agent. However, for the case of optimizing the sum of agents' satisfactions, we found good approximation algorithms. In particular, for the Monroe system we suggest a randomized algorithm that for the Borda score achieves an approximation ratio arbitrarily close to 0.715 (and much better ratios in many real-life settings), and $(1 - \frac{1}{e})$-approximation algorithm for arbitrary positional scoring function. For the Chamberlin–Courant system, we have shown a polynomial-time approximation scheme (PTAS).

In Table 6 we present the summary of our (in)approximability results. In Table 7 we present specific results regarding our approximation algorithms for the utilitarian satisfaction-based framework. In particular, the table clearly shows that for the case of Monroe, Algorithms B and C are not much slower than Algorithm A but offer a chance of improved performance. Algorithm GM is intuitively even more appealing, but achieves this at the cost of high time complexity. For the case of Chamberlin–Courant rule, theoretical results suggest using Algorithm P (however, see below).

We have provided experimental evaluation of the algorithms for computing the winner sets both for the Monroe and Chamberlin–Courant rules. While finding solutions for these rules is computationally hard in the worst case, it turned out

Table 6
Summary of approximability results for the Monroe and Chamberlin–Courant multiwinner voting systems and for the general resource allocation problem (see Appendix A for the general resource allocation model and Appendix C for the egalitarian variants of our rules).

	Monroe		Chamberlin and Courant		General model	
	dissat.	satisfaction	dissat.	satisfaction	dissat.	satisfaction
util.	Inapprox. Theorem 1	Good approx.	Inapprox. Theorem 2	Good approx.	Inapprox. Theorem 1 Theorem 2	Open problem
egal.	Inapprox. Theorem 15	Inapprox. Theorem 17	Inapprox. Theorem 16	Inapprox. Theorem 19	Inapprox. Theorem 15 Theorem 16	Inapprox. Theorem 17 Theorem 19

Table 7
A summary of the algorithms studied in this paper. The top of the table regards algorithms for the Monroe rule and the bottom for the Chamberlin–Courant rule. In column “Approximation” we give currently known approximation ratio for the algorithm under Borda PSF, on profiles with m candidates and where the goal is to select a committee of size K. Here, $O(\Phi^S) = O(n^2(K + \log n))$ is the complexity of finding a partial representation function with the algorithm of Betzler et al. [9]. w(·) denotes Lambert's W-function.

	Algorithm	Approximation	Runtime	Reference
Monroe	A	$1 - \frac{K-1}{2(m-1)} - \frac{H_K}{K}$	Kmn	Lemma 4
	B	as in Algorithm A	$Kmn + O(\Phi^S)$	Lemma 4
	C	as in Algorithm A	$dKmn + dO(\Phi^S)$	Lemma 4
	GM	as in Alg. A for Borda PSF; $1 - \frac{1}{e}$ for others	$KmO(\Phi^S)$	Theorem 10
	R	$\frac{1}{2}(1 + \frac{K}{m} - \frac{K^2m-K^3}{m^3-m^2})$	$\frac{\lvert\log(1-\lambda)\rvert}{K\epsilon^2}O(\Phi^S)$	Lemma 7
	AR	0.715	max(A, R)	Theorem 8
CC	PTAS			Corollary 13
	P	$1 - \frac{2\mathrm{w}(K)}{K}$	$nm\mathrm{w}(K)$	Theorem 11
	GM	$1 - \frac{1}{e}$	Kmn	Lu and Boutilier [32]
	C	as in Algorithm GM	$dKm(n + \log dm)$	Lu and Boutilier [32]
	R	$(1 - \frac{1}{K+1})(1 + \frac{1}{m})$	$\frac{\lvert\log(1-\lambda)\rvert}{\epsilon^2}n$	Oren [45]

that in practice we can obtain very high quality solutions using very simple approximation algorithms. Indeed, both for the Monroe and Chamberlin–Courant rules we recommend using Algorithm C (or Algorithm A on very large Monroe elections). Our experimental evaluation confirms that the algorithms work very well in the case of truncated ballots. We believe that our results mean that (approximations of) the Monroe and Chamberlin–Courant rules can be used in practice.

Our work leads to a number of further research directions. First, it would be interesting to find a better upper bound on the quality of solutions for the (satisfaction-based) Monroe and Chamberlin–Courant systems (with Borda PSF) than the simple $n(m-1)$ bound that we use (where n is the number of voters and m is the number of candidates). We use a different approach in our randomized algorithm, but it would be much more interesting to find a deterministic algorithm that beats the approximation ratios of our algorithms. One of the ways of seeking such a bound would be to consider the Monroe rule with “exponential” Borda PSF, that is, with PSF of the form, e.g., $(2^{m-1}, 2^{m-2}, \dots, 1)$. For such PSF our approach in the proof of Lemma 4 would not give satisfactory results and so one would be forced to seek different attacks. In a similar vein, it would be interesting to find out if there is a PTAS for the Monroe system.

In our work, we have focused on PSFs that are strictly increasing/decreasing. It would also be interesting to study PSFs which increase/decrease but not strictly, that is, PSFs which allow some equalities. We have started to work in this direction by considering agents with dichotomous preferences. Formally, we consider t-approval PSF's α_t, which (in the satisfaction-based variant) are defined as follows: $\alpha_t(i) = 1$ if $i \le t$ and otherwise $\alpha_t(i) = 0$. Results for this case for the Chamberlin–Courant rule are presented in the paper of Skowron and Faliszewski [57]. (As a side comment, we mention that there are also other interesting multiwinner voting rules that focus on dichotomous preferences; we point the reader to the overview of Kilgour [29] and to algorithmic studies of, for example, LeGrand et al. [31], Caragiannis et al. [12], and Aziz et al. [3].)

On a more practical side, it would be interesting to develop our study of truncated ballots. Our results show that we can obtain very high approximation ratios even when voters rank only relatively few of their top candidates. For example, to achieve 90% approximation ratio for the satisfaction-based Monroe system in Polish parliamentary election ($K = 460$, $m = 6000$), each voter should rank about 8.7% of her most-preferred candidates. However, this is still over 500 candidates. It is unrealistic to expect that the voters would be willing to rank this many candidates. Thus, the obvious question is how should one organize Monroe-based elections in practice, to balance the amount of effort required from the voters and the quality of the results?

Acknowledgements

This paper is based on two extended abstracts, presented at IJCAI-2013 and AAMAS-2012. We would like to thank the reviewers from these two conferences for very useful feedback. The authors were supported in part by Poland's National Science Centre grants UMO-2012/06/M/ST1/00358, DEC-2011/03/B/ST6/01393, and by AGH University of Science and Technology grant 11.11.230.124. Piotr Skowron was also supported by EU's Human Capital Program "National PhD Programme in Mathematical Sciences" carried out at the University of Warsaw and Polands National Science Center grant UMO-2013/09/N/ST6/03661.

Appendix A. The Monroe and Chamberlin–Courant rules as resource allocation problems

One of the ways of looking at the Monroe and Chamberlin–Courant rules is to consider them through the lenses of resource allocation. In this section we describe this more general view.

As before, we assume that we have a set $N = [n]$ of agents and a set $A = \{a_1, \dots, a_m\}$ of items. In the voting setting the items were the candidates participating in election; now they can be other entities as well (see the example later). Each agent has a preference order over A. The understanding is that each agent will receive access to one of the items and in the preference order each agent ranks the items from the most desired one to the least desired one. Additionally, each item $a \in A$ has the *capacity* $\mathrm{cap}_a \in \mathbb{N}$. At most cap_a agents can be granted access to item a. A capacitated K-assignment function Φ is a K-assignment that satisfies the constraint that for each item a, $\|\Phi^{-1}(a)\| \le \mathrm{cap}_a$.

Given this setup, our resource allocation problem is the following (for the dissatisfaction-based case and for the satisfaction-based case).

Definition 7. Let α be a normal IPSF. An instance of α-Capacitated-DisWinner problem consists of a set of agents $N = [n]$, a set of items $A = \{a_1, \dots, a_m\}$, a preference profile V of the agents, a sequence $(\mathrm{cap}_{a_1}, \dots, \mathrm{cap}_{a_m})$ of items' capacities, and a positive integer K. We ask for a capacitated K-assignment function Φ such that $\ell^{\alpha}_{\mathrm{sum}}(\Phi)$ is minimized.

Definition 8. Let α be a normal DPSF. α-Capacitated-SatWinner is defined in the same way as α-Capacitated-DisWinner except that we ask to maximize $\ell^{\alpha}_{\mathrm{sum}}(\Phi)$.

Now it is easy to see that for each IPSF α (DPSF α), α-CC-DisWinner (α-CC-SatWinner) is a special case of α-Capacitated-DisWinner (α-Capacitated-SatWinner): It suffices that the capacities are set to be $\|N\|$. Similarly, if K divides $\|N\|$ and we set the all the capacities to $\frac{\|N\|}{K}$, then we obtain the variants of the problem for the Monroe rule.

Nonetheless, Capacitated-DisWinner (Capacitated-SatWinner) is a far more general problem and we believe that it is interesting in its own right. For example, consider a company that wants to pay for sport classes for its employees. Naturally, each class can fit only a given number of employees and, due to limited budget, the company wants to pay for K different classes only. However, it also wants the employees to be as happy with the provided classes as possible and, so, it asks each employee to rank the classes. Our problem models exactly the issue of choosing which classes to offer and which employees should be assigned to which one of them.

While all the negative results for the case of the Monroe and Chamberlin–Courant rules translate directly to our resource allocation problem, we do not have any positive results for it. We believe that finding approximation algorithms for it (in the satisfaction-based setting) is a very interesting, challenging, open problem.

Appendix B. ILP formulation for the Monroe and Chamberlin–Courant rules

To experimentally measure the quality of our approximation algorithms, we compare the results against optimal solutions that we obtain using integer linear programs (ILPs) that solve the Monroe and Chamberlin–Courant winner determination problem. An ILP for the Monroe rule was provided by Potthoff and Brams [47], Lu and Boutilier [32] adapted it also for the Chamberlin–Courant rule with arbitrary PSF α. For the sake of completeness, below we recall the ILP whose optimal solutions correspond to α-Monroe-SatWinner winner sets for the given election (we also indicate which constraints to drop to obtain an ILP for finding α-CC-SatWinner winner sets):

1. For each i, $1 \le i \le n$, and each j, $1 \le j \le m$ we have a 0/1 variable a_{ij} indicating whether alternative a_j represents agent i. For each j, $1 \le j \le m$, we have a 0/1 variable x_j indicating whether alternative a_j is included in the set of winners.
2. Our goal is to maximize the value $\sum_{i=1}^{n} \alpha(pos_i(a_j))a_{ij}$ subject to the following constraints:
   (a) For each i and j, $1 \le i \le n, 1 \le j \le m$, $0 \le a_{ij} \le x_j$ (alternative a_j can represent agent i only if a_j belongs to the set of winners).
   (b) For each i, $1 \le i \le n$, $\sum_{1 \le j \le m} a_{ij} = 1$ (every agent is represented by exactly one alternative).

(c) For each j, $1 \le j \le m$, $x_j \lfloor \frac{n}{K} \rfloor \le \sum_{1 \le i \le n} a_{ij} \le x_j \frac{n}{K}$ (each alternative either does not represent anyone or represents between $\lfloor \frac{n}{K} \rfloor$ and $\frac{n}{K}$ agents; if we remove these constraints then we obtain an ILP for the Chamberlin–Courant rule).
(d) $\sum_{j=1}^{n} x_j \le K$ (there are exactly K winners).[11]

We used the GLPK 4.47 package (GNU Linear Programming Kit, version 4.47) to solve these ILPs, whenever it was possible to do so in reasonable time.

Appendix C. Egalitarian cases: hardness results

In this section we provide our results for the egalitarian variants of the Monroe and Chamberlin–Courant rules. We start by providing formal definitions of our problems.

Definition 9. Let α be a normal IPSF. Problems α-EgalitarianCC-DisWinner and α-EgalitarianMonroe-DisWinner are defined in the same way as α-CC-DisWinner and α-Monroe-DisWinner, respectively, except that we seek to minimize $\ell^{\alpha}_{\max}(\Phi)$ (instead of $\ell^{\alpha}_{\mathrm{sum}}(\Phi)$).

Definition 10. Let α be a normal DPSF. Problems α-EgalitarianCC-SatWinner and α-EgalitarianMonroe-SatWinner are defined in the same way as α-CC-SatWinner and α-Monroe-SatWinner, respectively, except that we seek to maximize $\ell^{\alpha}_{\min}(\Phi)$ (instead of $\ell^{\alpha}_{\mathrm{sum}}(\Phi)$).

In other words, the egalitarian variants of our problems are defined in the same way as the utilitarian variants, but the aggregated satisfaction of the agents (the aggregated dissatisfaction of the agents) is understood as that of the worst-off one of them.

We define the notion of approximation algorithms for the egalitarian variants of the problems in the same way as in the utilitarian case. We show that there are no constant-factor approximation algorithms for the egalitarian variants of the Monroe and Chamberlin–Courant rules.

Theorem 15. *For each normal IPSF α and each constant $r > 1$, there is no polynomial-time r-approximation algorithm for α-*Monroe-DisWinner *unless* P = NP.

Proof. The proof of Theorem 1 applies to this case as well. In fact, it even suffices to take $m' = \|A_2\| = \frac{1}{2}nr \cdot \alpha(3)$. □

Theorem 16. *For each normal IPSF α and each constant factor $r > 1$, there is no polynomial-time r-approximation algorithm for α-*EgalitarianCC-DisWinner *unless* P = NP.

Proof. The proof of Theorem 2 is applicable in this case as well. In fact, it even suffices to take the m groups of alternatives, $A_1, \dots, A_m$, to contain $\alpha(2) \cdot r$ alternatives each. □

Theorem 17. *For each normal DPSF α (where each entry is polynomially bounded in the number of alternatives) and each constant factor r, with $0 < r \le 1$, there is no r-approximation algorithm for α-*EgalitarianMonroe-SatWinner *unless* P = NP.

Proof. Let us fix a DPSF $\alpha = (\alpha^m)_{m \in \mathbb{N}}$, where each entry α^m is polynomially bounded in the number of alternatives m. For the sake of contradiction, let us assume that for some r, $0 < r \le 1$, there is a polynomial-time r-approximation algorithm $\mathcal{A}$ for α-EgalitarianMonroe-SatWinner. We will show that the existence of this algorithm implies that X3C is solvable in polynomial time.

Let I be an X3C instance with ground set $U = \{1, 2, \dots, n\}$ and collection $\mathcal{F} = \{F_1, \dots, F_m\}$ of subsets of U. Each set in $\mathcal{F}$ has cardinality three. Further, without loss of generality, we can assume that n is divisible by three and that each $i \in U$ appears in at most three sets from $\mathcal{F}$. Given I, we form an instance I_M of α-EgalitarianMonroe-SatWinner as follows. Let $n' = 3 \cdot (\alpha^{m+1}(1) \cdot \lceil \frac{1-r}{r} \rceil + 3)$. The set N of agents is partitioned into two subsets, N_1 and N_2. N_1 contains n agents (intuitively, corresponding to the elements of the ground set U) and N_2 contains n' agents (used to enforce certain properties of the solution). The set of alternatives A is partitioned into two subsets, A_1 and A_2. We set $A_1 = \{a_1, \dots, a_m\}$ (members of A_1 correspond to the sets in $\mathcal{F}$), and we set $A_2 = \{b_1, \dots, b_{m'}\}$, where $m' = \frac{n'}{3}$.

For each j, $1 \le j \le n$, we set $M_f(j) = \{a_i \mid j \in F_i\}$. For each j, $1 \le j \le n$, we set the preference order of the j-th agent in N_1 to be of the form

$$M_f(j) \succ A_2 \succ A_1 - M_f(j).$$

[11] For the Monroe framework inequality here is equivalent to equality. We use the inequality so that deleting constraints from item (2c) leads to an ILP for the Chamberlin–Courant rule.

Note that by our assumptions, $\|M_f(j)\| \le 3$. For each j, $1 \le j \le n'$, we set the preference order of the j-th agent in N_2 to be of the form

$$b_{\lceil \frac{j}{3} \rceil} \succ A_2 - \{b_{\lceil \frac{j}{3} \rceil}\} \succ A_1.$$

Note that each agent in N_2 ranks the alternatives from A_1 in positions $m'+1, \dots, m'+m$. Finally, we set the number of candidates that can be selected to be $K = \frac{n+n'}{3}$.

Now, consider the solution Φ returned by $\mathcal{A}$ on I_M. We will show that $\ell_{\max}^{\alpha^{m+m'}}(\Phi) \le r\alpha^{m+m'}(3)$ if and only if I is a *yes*-instance of X3C.

($\Leftarrow$) If there exists an exact set cover of U with sets from $\mathcal{F}$, then it is easy to construct a solution for I_M where the satisfaction of each agent is greater or equal to $r \cdot \alpha^{m+m'}(3)$. Let $I \subseteq \{1, \dots, m\}$ be a set such that $\bigcup_{i \in I} F_i = U$ and $\|I\| = \frac{n}{3}$. We assign each agent j from N_1 to the alternative a_i such that (a) $i \in I$ and (b) $j \in F_i$, and we assign each agent from N_2 to her most preferred alternative. Thus, Algorithm $\mathcal{A}$ has to return an assignment with the minimal satisfaction greater or equal to $r \cdot \alpha^{m+m'}(3)$.

($\Rightarrow$) For the other direction, we first show that $r \cdot \alpha^{m+m'}(3) \ge \alpha^{m+m'}(m')$. Since DPSFs are strictly decreasing, it holds that:

$$r \cdot \alpha^{m+m'}(3) \ge r \cdot \left(\alpha^{m+m'}\left(m'\right) + m' - 3\right). \tag{1}$$

Then, by the definition of DPSFs, it holds that:

$$\alpha^{m+m'}\left(m'\right) = \alpha^{m+1}(1). \tag{2}$$

Using the fact that $m' = (\alpha^{m+1}(1) \cdot \lceil \frac{1-r}{r} \rceil + 3)$ and using (2), we can transform inequality (1) to obtain the following:

$$\begin{aligned}
r \cdot \alpha^{m+m'}(3) &\ge r \cdot \left(\alpha^{m+m'}\left(m'\right) + m' - 3\right) \\
&= r \cdot \left(\alpha^{m+m'}\left(m'\right) + \left(\alpha^{m+1}(1) \cdot \left\lceil \frac{1-r}{r} \right\rceil + 3\right) - 3\right) \\
&\ge r \cdot \alpha^{m+m'}\left(m'\right) + (1-r) \cdot \alpha^{m+1}(1) \\
&= r \cdot \alpha^{m+m'}\left(m'\right) + (1-r) \cdot \alpha^{m+m'}\left(m'\right) = \alpha^{m+m'}\left(m'\right).
\end{aligned}$$

This means that if the minimal satisfaction of an agent is at least $r \cdot \alpha^{m+m'}(3)$, then no agent was assigned to an alternative that she ranked beyond position m'. If some agent j from N_1 were assigned to an alternative from A_2, then, by the pigeonhole principle, some agent from N_2 would be assigned to an alternative from A_1. However, each agent in N_2 ranks the alternatives from A_1 beyond position m' and thus such an assignment is impossible. In consequence, it must be that each agent in j was assigned to an alternative that corresponds to a set F_i in $\mathcal{F}$ that contains j. Such an assignment directly leads to a solution for I. □

Let us now move on to the case of EgalitarianCC-SatWinner family of problems. Unfortunately, in this case our inapproximability argument holds for the case of Borda DPSF only (though we believe that it can be adapted to other DPSFs as well). Further, in our previous theorems we were showing that existence of a respective constant-factor approximation algorithm implies that NP collapses to P. In the following theorem we will show a seemingly weaker collapse of W[2] to FPT.

To prove hardness of approximation for $\alpha_{\mathrm{B,dec}}$-EgalitarianCC-SatWinner, we first prove the following simple lemma.

Lemma 18. *Let K, p, l be three positive integers and let X be a set of cardinality lpK. There exists a family $\mathcal{S} = \{S_1, \dots, S_{\binom{lK}{K}}\}$ of pK-element subsets of X such that for each K-element subset B of X, there is a set $S_i \in \mathcal{S}$ such that $B \subseteq S_i$.*

Proof. Set $X' = [lK]$ and let $\mathcal{S}'$ be a family of all K-element subsets of X'. Replace each element i of X' with p new elements (at the same time replacing i with the same p elements within each set in $\mathcal{S}'$ that contains i). As a result we obtain two new sets, X and $\mathcal{S}$, that satisfy the statement of the theorem (up to the renaming of the elements). □

Theorem 19. *Let $\alpha_{\mathrm{B,dec}}^m$ be the Borda DPSF ($\alpha_{\mathrm{B,dec}}^m(i) = m - i$). For each constant factor r, $0 < r \le 1$, there is no r-approximation algorithm for $\alpha_{\mathrm{B,dec}}^m$-EgalitarianCC-SatWinner unless* FPT = W[2].

Proof. For the sake of contradiction, let us assume that there is some constant r, $0 < r \le 1$, and a polynomial-time r-approximation algorithm $\mathcal{A}$ for $\alpha_{\mathrm{B,dec}}^m$-EgalitarianCC-SatWinner. We will show that the existence of this algorithm implies that Set-Cover is fixed-parameter tractable for the parameter K (since Set-Cover is known to be W[2]-complete for this parameter, this will imply FPT = W[2]).

Let I be an instance of SET-COVER with ground set $U = [n]$ and family $\mathcal{F} = \{F_1, F_2, \dots, F_m\}$ of subsets of U. Given I, we build an instance I_{CC} of $\alpha^m_{\mathrm{B,dec}}$-EGALITARIANCC-SATWINNER as follows. The set of agents N consists of n subsets of agents, $N_1, \dots, N_n$, where each group N_i contains exactly $n' = \binom{\lceil \frac{2}{r} \rceil K}{K}$ agents. Intuitively, for each i, $1 \le i \le n$, the agents in the set N_i correspond to the element i in U. The set of alternatives A is partitioned into two subsets, A_1 and A_2, such that: (1) $A_1 = \{a_1, \dots, a_m\}$ is a set of alternatives corresponding to the sets from the family $\mathcal{F}$, and (2) A_2, $\|A_2\| = \lceil \frac{2}{r} \rceil \lceil \frac{m(1+r)}{K} \rceil K$, is a set of dummy alternatives needed for our construction. We set $m' = \|A\| = m + \|A_2\|$.

Before we describe the preference orders of the agents in N, we form a family $R = \{r_1, \dots, r_{n'}\}$ of preference orders over A_2 that satisfies the following condition: For each K-element subset B of A_2, there exists r_j in R such that all members of B are ranked among the bottom $\lceil \frac{m(1+r)}{K} \rceil K$ positions in r_j. By Lemma 18, such a construction is possible (it suffices to take $l = \lceil \frac{2}{r} \rceil$ and $p = \lceil \frac{m(1+r)}{K} \rceil$); further, the proof of the lemma provides an algorithmic way to construct R.

We form the preference orders of the agents as follows. For each i, $1 \le i \le n$, set $M_f(i) = \{a_t \mid i \in F_t\}$. For each i, $1 \le i \le n$, and each j, $1 \le j \le n'$, the j-th agent from N_i has preference order of the form:

$$M_f(i) \succ r_j \succ A_1 - M_f(i)$$

(we pick any arbitrary, polynomial-time computable order of candidates within $M_f(i)$ and $A_1 - M_f(i)$).

Let Φ be an assignment computed by $\mathcal{A}$ on I_M. We will show that $\ell_{\max}^{\alpha^{m'}_{\mathrm{B,dec}}}(\Phi) \ge r \cdot (m' - m)$ if and only if I is a *yes*-instance of SET-COVER.

($\Leftarrow$) If there exists a solution for I (i.e., a cover of U with K sets from $\mathcal{F}$), then we can easily show an assignment where each agent is assigned to an alternative that he or she ranks among the top m positions (namely, for each j, $1 \le j \le n$, we assign all the agents from the set N_j to the alternative $a_i \in A_1$ such that $j \in F_i$ and F_i belongs to the alleged K-element cover of U). Under this assignment, the least satisfied agent's satisfaction is at least $m' - m$ and, thus, $\mathcal{A}$ has to return an assignment Φ where $\ell_{\max}^{\alpha^{m'}_{\mathrm{B,dec}}}(\Phi) \ge r \cdot (m' - m)$.

($\Rightarrow$) Let us now consider the opposite direction. We assume that $\mathcal{A}$ found an assignment Φ such that $\ell_{\max}^{\alpha^{m}_{\mathrm{B,dec}}}(\Phi) \ge r \cdot (m' - m)$ and we will show that I is a *yes*-instance of SET-COVER. We claim that for each i, $1 \le i \le n$, at least one agent j in N_i were assigned to an alternative from A_1. If all the agents in N_i were assigned to alternatives from A_2, then, by the construction of R, at least one of them would have been assigned to an alternative that she ranks at a position greater than $\|A_2\| - \lceil \frac{m(1+r)}{K} \rceil K = \lceil \frac{2}{r} \rceil \lceil \frac{m(1+r)}{K} \rceil K - \lceil \frac{m(1+r)}{K} \rceil K$. For $x = \lceil \frac{m(1+r)}{K} \rceil K$ we have:

$$\left\lceil \frac{2}{r} \right\rceil x - x \ge m' - m'r + mr$$

(we skip the straightforward calculation) and, thus, this agent would have been assigned to an alternative that she ranks at a position greater than $m' - m'r + mr$. As a consequence, this agent's satisfaction would be lower than $(m' - m)r$. Similarly, no agent from N_i can be assigned to an alternative from $A_1 - M_f(i)$. Thus, for each i, $1 \le i \le n$, there exists at least one agent $j \in N_i$ that is assigned to an alternative from $M_f(i)$. In consequence, the covering subfamily of $\mathcal{F}$ consists simply of those sets F_t, for which some agent is assigned to alternative $a_t \in A_1$.

The presented construction gives the exact algorithm for SET-COVER problem running in time $f(K)(n+m)^{O(1)}$, where $f(K)$ is polynomial in $\binom{\lceil \frac{2}{r} \rceil}{K}$. The existence of such an algorithm means that SET-COVER is in FPT. On the other hand, we know that SET-COVER is W[2]-complete, and thus if $\mathcal{A}$ existed then FPT = W[2] would hold. □

Appendix D. Other proofs omitted from the main text

Proof of Theorem 2. Let us fix a normal IPSF α. For the sake of contradiction, let us assume that there is some constant $r > 1$, and a polynomial-time r-approximation algorithm $\mathcal{A}$ for α-CC-DISWINNER. We will show that it is possible to use $\mathcal{A}$ to solve the NP-complete VERTEX-COVER problem.

Let $I = (U, \mathcal{F}, K)$ be an instance of VERTEX-COVER, where $U = [n]$ is the ground set, $\mathcal{F} = \{F_1, \dots, F_m\}$ is a family of subsets of U (where each member of U belongs to exactly two sets in $\mathcal{F}$), and K is a positive integer.[12]

Given I, we construct an instance I_{CC} of α-CC-DISWINNER as follows. The set of agents is $N = U$ and the set of alternatives is $A = \bigcup_{j=1}^m A_j$, where each A_j contains exactly $\alpha(2) \cdot r \cdot n$ (unique) alternatives. Our intention is that for each $j \in [m]$ the alternatives in A_j correspond to the set F_j. For each A_j, $1 \le j \le m$, we pick one alternative, which we denote a_j. For each agent $i \in N$, we set i's preference order as follows: Let F_j and F_t, $j < t$, be the two sets that contain i. Agent i's preference order is of the form $a_j \succ_i a_t \succ_i A_t - \{a_t\} \succ_i A - (A_t \cup \{a_j, a_t\})$ (the particular order of alternatives in the sets

[12] In the typical view of the VERTEX-COVER problem we speak of a graph where each vertex corresponds to a set in $\mathcal{F}$, where each edge corresponds to a member of U, and where the goal is to pick K vertices that cover all the edges. The interpretation of the set U is as follows: If an edge $k \in U$ belonged to sets F_i and F_j, this means that this edge connects vertices i and j.

$A_t - \{a_t\}$ and $A - (A_t \cup \{a_j, a_t\})$ is irrelevant for the construction). We ask for an assignment of the agents to at most K alternatives.

Let us consider a solution Φ returned by $\mathcal{A}$ on input I_{CC}. We claim that $\ell^{\alpha}_{\mathrm{sum}}(\Phi) \le nr \cdot \alpha(2)$ if and only if I is a *yes*-instance of VERTEX-COVER.

($\Leftarrow$) If I is a *yes*-instance then each agent i can be assigned to one of the top two alternatives in his or her preference order (if there is a size-K cover, then this assignment selects at most K candidates). Thus the total dissatisfaction of an optimal assignment is at most $n \cdot \alpha(2)$. As a result, the solution Φ returned by $\mathcal{A}$ has total dissatisfaction at most $nr \cdot \alpha(2)$.

($\Rightarrow$) If $\mathcal{A}$ returns an assignment with total dissatisfaction no greater than $nr \cdot \alpha(2)$, then, by the construction of agents' preference orders, we see that each agent i was assigned to an alternative from a set A_j such that $i \in F_j$. Since the assignment can use at most K alternatives, this directly implies that there is a size-K cover of U with sets from $\mathcal{F}$. □

Proof of Proposition 6. We use the same approach as in the proof of Lemma 4, except that we adjust our estimates of voters' satisfaction. Consider a situation after some i-th iteration of the algorithm's outer loop ($i = 0$ if we are before the first iteration). If $i + \frac{m-i}{K-i} \le P$, then we can use the same lower bound for the satisfaction of the agents assigned in the $(i+1)$-th iteration as in the proof of Lemma 4. That is, the agents assigned in the $(i+1)$-th iteration will have satisfaction at least $r_1(i) = \frac{n}{K} \cdot (m - i - \frac{m-i}{K-i})$.

For the case where $i + \frac{m-i}{K-i} > P$, the bound from Lemma 4 does not hold, but we can use a similar approach to find a different one. Let $P_x \le P$ be some positive integer. We are interested in the number x of not-yet assigned agents who rank some not-yet-selected alternative among their top P_x positions (after the i-th iteration). Similarly as in the proof of Lemma 4, using the pigeonhole principle we note that:

$$x \ge \frac{1}{m-i}\left(n - i\frac{n}{K}\right)(P_x - i) = \frac{n}{K} \cdot \frac{(K-i)(P_x-i)}{m-i}.$$

Thus, the satisfaction of the agents assigned in the $(i+1)$-th iteration is at least:

$$\min\left(x, \frac{n}{K}\right)(m - P_x) = \frac{n}{K} \cdot (m - P_x) \min\left(\frac{(K-i)(P_x-i)}{m-i}, 1\right). \tag{3}$$

The case $\frac{(K-i)(P_x-i)}{m-i} \ge 1$ (or, equivalently, $i + \frac{m-i}{K-i} \le P_x$) implies that $i + \frac{m-i}{K-i} \le P$ and for this case we lower-bound agents' satisfaction by $r_1(i)$. For the case where $\frac{(K-i)(P_x-i)}{m-i} \le 1$, i.e. where $i + \frac{m-i}{K-i} \ge P_x$, Eq. (3) simplifies to:

$$\frac{n}{K} \cdot (m - P_x) \cdot \frac{(K-i)(P_x-i)}{m-i}. \tag{4}$$

We use the estimate given by Eq. (4) to bound the satisfaction of the agents assigned in the $(i+1)$-th iteration for the cases where (a) $i + \frac{m-i}{K-i} \ge \frac{m+i}{2}$ and (b) $\frac{m+i}{2} \le P$ (or, equivalently, $(2P - m) \ge i \ge (K-2)$): We estimate Eq. (4) as follows:

$$\begin{aligned}\frac{n}{K} \cdot (m - P_x) \cdot \frac{(K-i)(P_x-i)}{m-i} &\ge \frac{n}{K} \cdot \left(m - \frac{m+i}{2}\right) \cdot \frac{(K-i)(\frac{m+i}{2} - i)}{m-i} \\ &= \frac{n}{K} \cdot \frac{(K-i)(m-i)^2}{4(m-i)} = \frac{n}{K} \cdot \frac{(K-i)(m-i)}{4}.\end{aligned}$$

For the remaining cases (corresponding to the "otherwise" clause in the definition of $r(i)$), we set $P_x = P$ and (4) becomes:

$$\frac{n}{K} \cdot \frac{(m-P)(K-i)(P-i)}{m-i}.$$

Naturally, we replace our estimates by 0 whenever they become negative.

To complete the proof, it suffice to, as in the proof of Lemma 4, note that $(m-1)n$ is an upper bound on the satisfaction achieved by the optimal solution. □

Proof of Theorem 9. The proof follows by applying the powerful result of Nemhauser et al. [43], which says that greedy algorithms achieve $(1 - \frac{1}{e})$-approximation ratio when used to optimize nondecreasing submodular functions (we explain these notions formally below). The main challenge in the proof is to define a function that, on one hand, satisfies the conditions of Nemhauser et al.'s result, and, on the other hand, models solutions for WEIGHTED-B-MATCHING.

Let $U \cup V$ be the set of vertices and E be the set of edges of a given bipartite graph. For each edge $\{u, v\} \in E$, let $w_{(u,v)}$ denote its weight. Without loss of generality, we assume that each vertex from U is connected with each vertex from V. Indeed, it is always possible to modify the original graph by introducing the missing edges and setting their weight to 0, so that the optimal solutions for the modified graph are the same as for the original one. Let $s : U \to \mathbb{N}$ be a function assigning to the vertices from U their capacities, and let K denote the number of vertices from U that can be used in a B-matching. For each $S \subseteq U$ such that $\|S\| \le K$, by $\Phi^S : V \to U$ we denote the optimal B-matching in the graph induced by $S \cup V$; by

$\ell(\Phi^S)$ we denote the total weight of Φ^S. It is easy to verify that both Φ^S and $\ell(\Phi^S)$ are computable in polynomial time. We define function z, $z: 2^U \to \mathbb{R}_+$, so that for each $S \subseteq U$, $\|S\| \le K$, we have $z(S) = \ell(\Phi^S)$. Clearly, $z(S)$ is nondecreasing (that is, for each two sets A and B, if $A \subseteq B \subseteq U$ and $\|B\| \le K$, then $z(A) \le z(B)$). Let $S_{\max} = \mathrm{argmax}_{S \subset U, \|S\|=K} z(S)$. We see that each optimal B-matching for the graph induced by $S_{\max} \cup V$ is also optimal for the original graph. Thus, if z were submodular, then by the result of Nemhauser et al. [43] we would get that the greedy algorithm from the theorem statement is a $(1 - \frac{1}{e})$-approximation algorithm for Weighted-B-Matching (for bipartite graphs). So, our goal is to show that z is submodular.

Formally, function z is submodular if for each two sets S and T, $S \subset T \subset U$, and each vertex $u \in U \setminus T$ it holds that:

$$z(S \cup \{u\}) - z(S) \ge z(T \cup \{u\}) - z(T).$$

We will show a slightly stronger variant of submodularity. However, first we introduce some additional notation for capacity functions. For each $u \in U$, we write $s \cup \{u\}$ to denote the function such that $(s \cup \{u\})(u) = s(u) + 1$ and $\forall_{u' \in S \setminus \{u\}} (s \cup \{u\})(u') = s(u')$. (Analogously, we interpret $s \setminus \{u\}$ as subtracting one from the capacity for u, provided the capacity is nonzero.) Also, if s and t are two capacity functions, then by $s \le t$ we mean that $\forall_{u \in U} s(u) \le t(u)$. We extend our function z to allow subsets of the vertices from V and arbitrary capacity functions: For each subset $V' \subseteq V$ and each capacity function s, we define $z_{V'}(s)$ to be the weight of the optimal B-matching for a graph induced by $U \cup V'$, such that the capacities of the vertices from U are given by the capacity function s. We will now prove a stronger variant of submodularity for our extended z. That is, we will show that for each two capacity functions s and t, each $V' \subseteq V$, and each vertex $u \in U$ it holds that:

$$s \le t \Rightarrow z_{V'}(s \cup \{u\}) - z_{V'}(s) \ge z_{V'}(t \cup \{u\}) - z_{V'}(t). \tag{5}$$

We will also extend notation for optimal matchings: Let $\Phi^s_{V'} : V' \to U$ denote the optimal B-matching for a graph induced by $U \cup V'$, and capacity function s.

Our proof is by induction on V'. Clearly, Eq. (5) holds for $V' = \emptyset$. Now, assuming that Eq. (5) holds for every $V' \subset V''$ we will prove its correctness for V''. Let us fix some vertex $u \in U$ and let $v \in V''$ denote a vertex such that $\Phi^{t \cup \{u\}}_{V''}(v) = u$ (if there is no such vertex v then clearly the equation holds). Let $u_s = \Phi^s_{V''}(v)$ and $u_t = \Phi^t_{V''}(v)$ (technically, it is possible that either of these vertices does not exist; in such a case, in the reasoning below we treat this vertex as a null one and we assume the weight of each edge incident to the null vertex is zero). We have:

$$z_{V''}(t \cup \{u\}) - z_{V''}(t) = w_{(u,v)} + z_{V'' \setminus \{v\}}(t) - w_{(u_t,v)} - z_{V'' \setminus \{v\}}(t \setminus \{u_t\}).$$

We also have:

$$z_{V''}(s \cup \{u\}) - z_{V''}(s) \ge w_{(u,v)} + z_{V'' \setminus \{v\}}(s) - w_{(u_s,v)} - z_{V'' \setminus \{v\}}(s \setminus \{u_s\}).$$

Since $\Phi^t_{V''}$ describes an optimal B-matching under the capacity restriction t, we have that:

$$w_{(u_t,v)} + z_{V'' \setminus \{v\}}(t \setminus u_t) \ge w_{(u_s,v)} + z_{V'' \setminus \{v\}}(t \setminus \{u_s\}).$$

Finally, from the inductive hypothesis for $V' = V'' \setminus \{v\}$ we have:

$$z_{V'' \setminus \{v\}}(s) - z_{V'' \setminus \{v\}}(s \setminus \{u_s\}) \ge z_{V'' \setminus \{v\}}(t) - z_{V'' \setminus \{v\}}(t \setminus \{u_s\}).$$

By combining these inequalities we get:

$$\begin{aligned} z_{V''}(s \cup \{u\}) - z_{V''}(s) &\ge w_{(u,v)} + z_{V'' \setminus \{v\}}(s) - \left(w_{(u_s,v)} + z_{V'' \setminus \{v\}}(s \setminus \{u_s\})\right) \\ &\ge w_{(u,v)} - w_{(u_s,v)} + z_{V'' \setminus \{v\}}(t) - z_{V'' \setminus \{v\}}(t \setminus \{u_s\}) \\ &\ge w_{(u,v)} + z_{V'' \setminus \{v\}}(t) - w_{(u_t,v)} - z_{V'' \setminus \{v\}}(t \setminus \{u_t\}) \\ &= z_{V''}(t \cup \{u\}) - z_{V''}(t). \end{aligned}$$

This completes the proof. □

Proof of Proposition 14. Let n_i denote the number of the agents not-yet-assigned until the $(i+1)$-th iteration of the algorithm. Using the same reasoning as in Theorem 11 we show that $n_i \le n(1 - \frac{Q}{m})^i$. As before, our proof proceeds by induction on i. It is evident that the hypothesis is correct for $i = 0$. Now, assuming that $n_i \le n(1 - \frac{Q}{m})^i$, we assess n_{i+1} as follows:

$$n_{i+1} \le n_i - \frac{n_i Q}{m - i} \le n_i\left(1 - \frac{Q}{m}\right) \le n\left(1 - \frac{Q}{m}\right)^{i+1}.$$

This proves the hypothesis. Thus, we can bound n_K:

$$n_K \le n\left(1-\frac{Q}{m}\right)^K \le n\left(\frac{1}{e}\right)^{\frac{QK}{m}}.$$

This means that the satisfaction of the assignment Φ returned by our algorithm is at least:

$$\ell_{\mathrm{sum}}^{\alpha_{\mathrm{B,dec}}}(\Phi) \ge (n-n_K)(m-Q) \ge n(m-Q)\left(1-e^{-\frac{QK}{m}}\right).$$

In effect, it holds that:

$$\frac{\ell_{\mathrm{sum}}^{\alpha_{\mathrm{B,dec}}}(\Phi)}{\mathrm{OPT}} \ge \frac{n(m-Q)(1-e^{-\frac{QK}{m}})}{n(m-1)} \ge \frac{m-Q}{m-1}\left(1-e^{-\frac{QK}{m}}\right).$$

This completes the proof. □

References

[1] N. Ailon, M. Charikar, A. Newman, Aggregating inconsistent information: ranking and clustering, J. ACM 55 (5) (2008) 23, Article 23.
[2] S. Airiau, U. Endriss, Multiagent resource allocation with sharable items: simple protocols and Nash equilibria, in: Proceedings the 9th International Conference on Autonomous Agents and Multiagent Systems, May 2010, pp. 167–174.
[3] H. Aziz, S. Gaspers, J. Gudmundsson, S. Mackenzie, N. Mattei, T. Walsh, Computational aspects of multi-winner approval voting, in: 8th Multidisciplinary Workshop on Advances in Preference Handling, MPREF-14, 2014.
[4] J. Bartholdi III, C. Tovey, M. Trick, Voting schemes for which it can be difficult to tell who won the election, Soc. Choice Welf. 6 (2) (1989) 157–165.
[5] D. Baumeister, P. Faliszewski, J. Lang, J. Rothe, Campaigns for lazy voters: truncated ballots, in: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, June 2012, pp. 577–584.
[6] S. Berg, Paradox of voting under an urn model: the effect of homogeneity, Public Choice 47 (1985) 377–387.
[7] N. Betzler, M. Fellows, J. Guo, R. Niedermeier, F. Rosamond, Fixed-parameter algorithms for Kemeny scores, Theor. Comput. Sci. 410 (45) (2009) 4554–4570.
[8] N. Betzler, J. Guo, R. Niedermeier, Parameterized computational complexity of Dodgson and Young elections, Inf. Comput. 208 (2) (2010) 165–177.
[9] N. Betzler, A. Slinko, J. Uhlmann, On the computation of fully proportional representation, J. Artif. Intell. Res. 47 (2013) 475–519.
[10] I. Caragiannis, J. Covey, M. Feldman, C. Homan, C. Kaklamanis, N. Karanikolas, A. Procaccia, J. Rosenschein, On the approximability of Dodgson and Young elections, Artif. Intell. 187 (2012) 31–51.
[11] I. Caragiannis, C. Kaklamanis, N. Karanikolas, A. Procaccia, Socially desirable approximations for Dodgson's voting rule, ACM Trans. Algorithms 10 (2) (2014), Article 6.
[12] I. Caragiannis, D. Kalaitzis, E. Markakis, Approximation algorithms and mechanism design for minimax approval voting, in: Proceedings of the 24th AAAI Conference on Artificial Intelligence, AAAI Press, July 2010, pp. 737–742.
[13] B. Chamberlin, P. Courant, Representative deliberations and representative decisions: proportional representation and the borda rule, Am. Polit. Sci. Rev. 77 (3) (1983) 718–733.
[14] Y. Chevaleyre, P. Dunne, U. Endriss, J. Lang, M. Lemaître, N. Maudet, J. Padget, S. Phelps, J. Rodríguez-Aguilar, P. Sousa, Issues in multiagent resource allocation, Informatica 30 (2006) 3–31.
[15] V. Conitzer, M. Rognlie, L. Xia, Preference functions that score rankings and maximum likelihood estimation, in: Proceedings of the 21st International Joint Conference on Artificial Intelligence, July 2009, pp. 109–115.
[16] D. Coppersmith, L. Fleisher, A. Rurda, Ordering by weighted number of wins gives a good ranking for weighted tournaments, ACM Trans. Algorithms 6 (3) (2010), Article 55.
[17] A. Darmann, E. Elkind, S. Kurz, J. Lang, J. Schauer, G. Woeginger, Group activity selection problem, in: Proceedings of the 8th International Workshop on Internet and Network Economics, 2012, pp. 156–169.
[18] R.G. Downey, M.R. Fellows, Parameterized Complexity, Springer, 1999.
[19] E. Elkind, P. Faliszewski, P. Skowron, A. Slinko, Properties of multiwinner voting rules, in: Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems, May 2014, pp. 53–60.
[20] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, Multimode control attacks on elections, J. Artif. Intell. Res. 40 (2011) 305–351.
[21] M.R. Fellows, H. Fernau, Facility location problems: a parameterized view, Discrete Appl. Math. 159 (11) (2011) 1118–1130.
[22] J. Flum, M. Grohe, Parameterized Complexity Theory, Springer-Verlag, 2006.
[23] M. Garey, D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, 1979.
[24] E. Hemaspaandra, L. Hemaspaandra, J. Rothe, Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP, J. ACM 44 (6) (1997) 806–825.
[25] E. Hemaspaandra, H. Spakowski, J. Vogel, The complexity of Kemeny elections, Theor. Comput. Sci. 349 (3) (2005) 382–391.
[26] C. Homan, L. Hemaspaandra, Guarantees for the success frequency of an algorithm for finding Dodgson-election winners, J. Heuristics 15 (4) (2009) 403–423.
[27] T. Kamishima, Nantonac collaborative filtering: recommendation based on order responses, in: Proceedings of KDD-03, 2003, pp. 583–588.
[28] C. Kenyon-Mathieu, W. Schudy, How to rank with few errors, in: Proceedings of the 39th Annual ACM Symposium on Theory of Computing, June 2007, pp. 95–103.
[29] D. Kilgour, Approval balloting for multi-winner elections, in: J. Laslier, R. Sanver (Eds.), Handbook on Approval Voting, Springer, 2010, pp. 105–124.
[30] M. Lackner, Incomplete preferences in single-peaked electorates, in: Proceedings of the 28th AAAI Conference on Artificial Intelligence, 2014, pp. 742–748.
[31] R. LeGrand, E. Markakis, A. Mehta, Some results on approximating the minimax solution in approval voting, in: Proceedings of the 6th International Conference on Autonomous Agents and Multiagent Systems, May 2007, pp. 1193–1195.
[32] T. Lu, C. Boutilier, Budgeted social choice: from consensus to personalized decision making, in: Proceedings of IJCAI-2011, 2011, pp. 280–286. See also a version that appeared in COMSOC-2010.
[33] T. Lu, C. Boutilier, Learning Mallows models with pairwise preferences, in: Proceedings of ICML-11, June 2011, pp. 145–152.
[34] C.L. Mallows, Non-null ranking models. I, Biometrika 44 (1–2) (1957) 114–130.
[35] N. Mattei, Empirical evaluation of voting rules with strictly ordered preference data, in: Proceedings of the Second International Conference on Algorithmic Decision Theory, 2011, pp. 165–177.
[36] N. Mattei, J. Forshee, J. Goldsmith, An empirical study of voting rules and manipulation with large datasets, Presented at the 4th International Workshop on Computational Social Choice COMSOC, 2012.

[37] N. Mattei, T. Walsh, PrefLib: a library of preference data, in: Proceedings of Third International Conference on Algorithmic Decision Theory, 2013, pp. 259–270.
[38] J. McCabe-Dansted, G. Pritchard, A. Slinko, Approximability of Dodgson's rule, Soc. Choice Welf. 31 (2) (2008) 311–330.
[39] J. McCabe-Dansted, A. Slinko, Exploratory analysis of similarities between common social choice rules, Group Decis. Negot. 15 (1) (2006) 77–107.
[40] B. Monroe, Fully proportional representation, Am. Polit. Sci. Rev. 89 (4) (1995) 925–940.
[41] L. Naamani-Dery, M. Kalech, L. Rokach, B. Shapira, Preference elicitation for narrowing the recommended list for groups, in: Proceedings of the 8th ACM Conference on Recommender Systems, October 2014, pp. 333–336.
[42] N. Narodytska, T. Walsh, The computational impact of partial votes on strategic voting, in: Proceedings of 21st European Conference on Artificial Intelligence, IOS Press, August 2014, pp. 657–662.
[43] G. Nemhauser, L. Wolsey, M. Fisher, An analysis of approximations for maximizing submodular set functions, Math. Program. 14 (1) (1978) 265–294.
[44] R. Niedermeier, Invitation to Fixed-Parameter Algorithms, Oxford University Press, 2006.
[45] J. Oren, Personal communication, 2012.
[46] A. Popova, M. Regenwetter, N. Mattei, A behavioral perspective on social choice, Ann. Math. Artif. Intell. 68 (1–3) (2013) 5–30.
[47] R. Potthoff, S. Brams, Proportional representation: broadening the options, J. Theor. Polit. 10 (2) (1998) 147–178.
[48] A. Procaccia, J. Rosenschein, A. Zohar, On the complexity of achieving proportional representation, Soc. Choice Welf. 30 (3) (2008) 353–362.
[49] F. Pukelsheim, Proportional Representation: Apportionment Methods and Their Applications, Springer, 2014.
[50] M. Regenwetter, B. Grofman, A. Marley, I. Tsetlin, Behavioral Social Choice: Probabilistic Models, Statistical Inference, and Applications, Cambridge University Press, 2006.
[51] J. Rothe, H. Spakowski, J. Vogel, Exact complexity of the winner problem for Young elections, ACM Trans. Comput. Syst. 36 (4) (2003) 375–386.
[52] F. Shepherd, A. Vetta, The Demand-Matching Problem, Math. Oper. Res. 32 (2007) 563–578.
[53] D. Shiryaev, L. Yu, E. Elkind, On elections with robust winners, in: Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems, 2013, pp. 415–422.
[54] Y. Shoham, K. Leyton-Brown, Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations, Cambridge University Press, 2009.
[55] H.A. Simon, Rational choice and the structure of the environment, Psychol. Rev. 63 (1956) 129–138.
[56] M. Singh, H. Wu, Nearly tight linear programming bounds for demand matching in bipartite graphs, Manuscript, 2012.
[57] P. Skowron, 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 Intelligence, January 2015, in press, http://arxiv.org/abs/1309.4405.
[58] P. Skowron, L. Yu, P. Faliszewski, E. Elkind, The complexity of fully proportional representation for single-crossing electorates, Theor. Comput. Sci. 569 (2015) 43–57.
[59] N. Tideman, The single transferable vote, J. Econ. Perspect. 9 (1995) 27–38.
[60] T. Walsh, Where are the hard manipulation problems?, J. Artif. Intell. Res. 42 (2011) 1–29.
[61] L. Yu, H. Chan, E. Elkind, Multiwinner elections under preferences that are single-peaked on a tree, in: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, 2013, pp. 425–431.