Condorcet Consistent Bundling with Social Choice

Shreyas Sekar
Rensselaer Polytechnic Institute
Computer Science Department
sekars@rpi.edu

Sujoy Sikdar
Rensselaer Polytechnic Institute
Computer Science Department
sikdas@rpi.edu

Lirong Xia
Rensselaer Polytechnic Institute
Computer Science Department
xial@cs.rpi.edu

ABSTRACT

We study the problem of computing optimal bundles given agents' preferences over individual items when agents derive satisfaction from the entire bundle under constraints on the size k of the bundle. Building on the notion of Condorcet winning sets by Gehrlein [16], we extend common Condorcet consistent voting rules from the single winner voting setting to that of forming bundles of size k. Our main technical contribution involves designing efficient algorithms for computing (approximately)-optimal bundles for multi-winner extensions of the following voting rules: Copeland, Minimax, Ranked Pairs, and Schulze.

1. INTRODUCTION

Product bundling, or more generally, the practice by which a central designer combines multiple independent entities to offer a single discernible bundle is a notion that is frequently encountered in a number of environments. For instance, cable television companies often bundle channels together, publishers offer a platter of academic journals, and insurance policies are typically a combination of several benefits. A natural problem that the 'seller' faces in such scenarios is that of bundle selection, i.e., how to choose a finite set of items from a large candidate pool.

Condorcet Consistency for Sets of Candidates. The current work is motivated by the following behavioral considerations that constrain the selection problem: (i) agents find it natural to express preferences over individual items and not combinations of items, (ii) agents derive satisfaction from all items in the bundle as opposed to just one representative member, and (iii) agents evaluate the bundle by comparing its members to items that were not selected for the bundle. In this work, we address these concerns by designing social choice mechanisms to construct bundles of exactly k candidates based on natural generalizations of the notion of *Condorcet consistency*.

The driving force behind our social choice mechanisms is Gehrlein's generalization of Condorcet winners to k-sized committees such that "*each member of the committee defeats every non-committee candidate on the basis of a simple majority rule*" [16]. We provide an alternative interpretation for this property in terms of comparisons between *neighboring bundles* that allows us to define a majority graph on sets of alternatives. Based on this, we present a framework to derive natural relaxations of Condorcet consistent mechanisms that depend only on (weighted) tournament graphs to multi-winner settings including *Copeland*, *Maximin*, *Ranked Pairs*, and *Schulze*.

Gehrlein's [16] notion of Condorcet consistency has been the subject of an extensive body of literature [5, 8, 18, 25]. This has resulted in extensions of well-studied Condorcet consistent mechanisms such as the Dodgson and Kemeny rules [25], as well as the Copeland and Maximin rules [8] (see [18] for a recent survey); the latter two rules are also the subject of this work. Our framework allows us to derive the extensions of Copeland and Maximin studied in [8] as well as the first extensions of the Ranked Pairs and Schulze rules for selecting k-sized bundles. Moreover, *all of the above papers* focus mainly on studying the properties of Condorcet sets and their various relaxations and not on the computation of Condorcet winning bundles. Unlike the single-winner setting, computing optimal bundles of size k is a non-trivial algorithmic problem, and in some cases, even NP-Hard. Bearing this in mind, we approach this problem via a computational lens and develop the first efficient algorithms for selecting (approximately-)optimal bundles for various Condorcet consistent bundling mechanisms.

1.1 Our contributions

The problem studied in this work is that of selecting a set of k items or candidates based only on agents' preferences over individual candidates. Conceptually, the main contribution of this work is a new framework for extending Condorcet consistent mechanisms from the single-winner to the bundling setting in a manner that is compatible with Gehrlein's [16] definition. Our approach allows us to generalize tournament graph based voting rules such as *Copeland, Maximin, Ranked Pairs, and Schulze* to the problem of selecting k-sized bundles when Condorcet winners do not exist.

Since the number of bundles of size k is exponential in k, our definitions do not lead to any computational insights. Therefore, we supplement our black-box approach by presenting efficient algorithms for computing winning bundles as well as hardness results for some rules. Our main technical contributions are as follows.

1. **Copeland:** Efficient algorithms for computing the winning bundle when there are no ties in the tournament graph. In the case of ties, we consider the extension Copeland$^\alpha$ [12] and present NP-Hardness proofs for

Appears in: *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

$\alpha = \{0, 1\}$. We also provide approximation algorithms for the problem of computing bundles with the highest Copeland$^\alpha$ score: specifically, a $\frac{1}{2}$-approximation algorithm for all $\alpha \in [0, 1]$.

2. **Maximin**: we show that the decision version corresponding to the problem of computing a Maximin winning bundle is NP-Complete. For any given instance, we then relax the size of the bundle and provide an approximation algorithm, which returns a bundle of size at most $2k$ that is as good as the optimum Maximin bundle of size k, i.e., its Maximin objective is at least that of the optimum Maximin bundle of size k.

3. **Ranked Pairs:** we give an efficient algorithm for computing the winning bundle.

4. **Schulze:** an efficient algorithm for computing the winning bundle.

Discussion of Related Work

Our problem is fundamentally different from *proportional representation* [10, 21, 24]. In that literature, an agent's satisfaction with a committee is determined *only* by the rank of her favorite candidate in that committee, i.e., a representative elected candidate. On the contrary, our model is motivated by applications where agents care about all members of the bundle or committee. Therefore, we are interested in computing bundles where every item is preferred by a large number of agents and each individual agents influences the selection of more than just one item in the bundle.

A more recent line of work has examined multi-winner voting through the lens of utility maximization, where each agent derives some utility from each item in the bundle and the goal is to select bundles to maximize the aggregate utility. These include generalizations of specific single-winner voting rules such as Borda [10], budgeted approaches towards proportional representation [21, 23], and the highly general proportional multi-representation or group recommendation framework [27]. All of these mechanisms are in some senses generalizations of positional scoring rules, whereas we are interested in Condorcet consistent rules. Another popular line of research concerns the design of multi-winner elections based on approval voting [19, 4, 3], where agent have 0-1 preferences over the items.

Our setting is also different from generalizations of Condorcet consistency for committees based on proportional representation [11] where a majority of the agents prefer (only) their favorite item in the bundle to each item not present in the bundle.

Our approach bears similarities to the extensive literature on combinatorial voting [20]. However, it is not always feasible for agents to express preferences over every potential bundle of items. In the same vein as the literature on compact preference representation languages, we take the approach of eliciting agents' preferences over individual items and extend them to preferences over bundles. At the core of our approach is the notion of 'local dominance', i.e., the mechanism cannot improve upon the outcome by switching an item in the bundle with another item that is not in the bundle, from the perspective of a majority of the agents. Such a notion has also been explored in the combinatorial voting literature as local Condorcet winners [9, 28] that beat "neighboring" alternatives by pairwise majority.

Gehrlein's notion of a Condorcet set has received considerable attention over the years. Ratliff [25] provided an equivalent definition of Condorcet committees and defined Dodgson and Kemeny committees as extensions to their single winner counterparts. In [8], the authors propose a slight relaxation of Gehrlein's definition, known as weak Condorcet consistency where no member of the committee is defeated by a member outside the committee by pairwise majority as well as extensions to the Copeland and Maximin rules to selecting committees of size k under this definition. Subsequent work [5, 18] has studied the stability of these rules as screening rules. Our work builds on the model in [8] and provides the first computational results for the extensions of Copeland and Maximin rules proposed in that work by presenting an efficient algorithm for computing the optimal Copeland bundle and an approximation algorithm for Maximin.

2. PRELIMINARIES

In our setting, $\mathcal{N}$ is a set of agents and $\mathcal{A}$ is a set of candidates. Formally, we are given a *preference profile* $\vec{P} = (P_i)_{i \in \mathcal{N}}$, where each P_i is a strict ranking over the items in $\mathcal{A}$. A *bundle* B of size k is any subset $B \subseteq \mathcal{A}, |B| = k$ of candidates. We use $\mathcal{A}_k$ to denote the set of all bundles of size k. Our objective is to select from $\mathcal{A}_k$ an optimal consensus bundle of size k in polynomial time; to achieve this, we will extend several Condorcet consistent mechanisms from single-winner elections to the multi-winner setting.

The Condorcet criterion for single-winner elections can be described through the notion of a *tournament graph* on the alternatives $\mathcal{A}$. The tournament graph T contains the directed edge (x, y) if a majority of agents prefer alternative x to y. Based on this definition, an alternative $x \in \mathcal{A}$ is said to be a Condorcet winner for single-winner elections if and only if it has outgoing edges to every other alternative y in the tournament graph T, i.e., for every alternative $y \neq x$, more users prefer x to y. We also define W to be the weighted complete graph over candidates such that the weight on the edge (x, y) is $N(x, y)$ that denotes the number of agents who prefer x to y.

2.1 Condorcet Consistent Bundling

How do we go about extending the idea of a Condorcet winner when we require a bundle of alternatives? One such paradigm was proposed by Fishburn [14, 15] who defined a Condorcet committee to be a bundle B such that for every other bundle B' of the same size, a majority of the users prefer B to B'. Unfortunately, such a direct comparison between bundles may be incompatible with item preferences. For example, suppose that in a given instance, agent i's preferences are $x \succ y \succ a \succ b$. Such information is often insufficient to conclude whether this agent prefers bundle $\{x, b\}$ or $\{y, a\}$. Moreover, comparing a given bundle B with *every other bundle* of the same size is perhaps inconsistent with how real users behave. With this in mind, we ask the following question:

> *Using only item preferences, when does a majority of the agents prefer one bundle to another?*

One simple and intuitive answer to this question is 'when the bundles are *neighbors*', i.e., the bundles differ only in one item. Formally, given two bundles B and B' that differ only in one item, we say that B *locally dominates* B' if a majority of the users prefer the item $(B \setminus B')$ to the item $(B' \setminus B)$. We use $N_B(B, B')$ to denote the number of agents who prefer B to its neighbor B'.

Using local dominance as a basis, we can now define a majority graph that leads to a natural extension of Condorcet winners. Specifically, let $T^{(k)}$ denote the unweighted, directed local majority graph whose vertex set is $\mathcal{A}_k$, i.e., the set of all bundles of size k. For every pair of neighboring bundles B, B' such that B locally dominates B', $T^{(k)}$ comprises of a directed edge from B to B'. For convenience one can interpret $T^{(k)}$ to be the bundling analog of the tournament graph T on alternatives. However, unlike the tournament graph, $T^{(k)}$ is not complete and only comprises of edges between bundles that differ in exactly one item. We now define our notion of Condorcet winners.

Definition 1. *Given an input parameter k, a bundle B of size k is said to be a Condorcet winning bundle if and only if B has an outgoing edge to every neighboring bundle B' in $T^{(k)}$.*

It is not hard to see that (i) such a Condorcet winning bundle may not always exist and *is unique* when it does, and (ii) when $k = 1$, this reduces to the classic definition of a Condorcet winner. One popular definition of a Condorcet winning set in the literature was proposed by Gehrlein [16]; according to this definition, *a set $B \subseteq \mathcal{A}$ is said to be a Condorcet winning committee if $\forall x \in B$ and any given $y \notin B$, a majority of the users prefer x to y.* Our notion turns out to be equivalent to Gehrlein's classic definition.

Proposition 1. *If B is a Condorcet winning bundle of size k, then for every $x \in B$ and $y \notin B$, a majority of the agents prefer x to y. Therefore, Definition 1 is equivalent to the definition of a Condorcet committee proposed in [16].*

Note that although the definition of Condorcet consistency considered in this paper was already known from [16], our main contribution can be viewed as providing a convenient interpretation for this notion, that allows us to provide natural extensions to well-studied Condorcet consistent mechanisms.

2.2 Condorcet Consistent Mechanisms for Bundling

Recall the definition of the local majority graph for bundles of size k, $T^{(k)}$. In a similar spirit, we define a weighted majority graph $W^{(k)}$, which is a directed graph whose vertex set is $\mathcal{A}_k$, and where for every pair of neighboring bundles, there are edges (B, B') and (B', B) with weights $N_B(B, B'), N_B(B', B)$ respectively. We now define a rule-of-thumb for extending many Condorcet consistent rules to the multi-winner setting.

Generic Template for Extending Condorcet Consistent Rules.

The precise definition of a Condorcet consistent voting rule is 'a rule that outputs a Condorcet winning alternative when it exists'. However, a number of Condorcet mechanisms are essentially functions whose output depends only on the (weighted) tournament graph. Consider any single-winner mechanism whose output depends only on T (equivalently, the weighted tournament graph W): *a natural extension of such rules to the multi-winner setting is to consider the exact same definition as before but applied on the (weighted) majority graph for bundles $T^{(k)}$ (equivalently $W^{(k)}$).* If the original rule is Condorcet consistent, then such an extension also selects a Condorcet winning bundle when one exists. This is the generic template that we will use to define extensions of popular voting rules such as Copeland, Minimax, Ranked Pairs, and Schulze.

Definition 2. *(Copeland^k) The Copeland^k rule selects a bundle B^* (of size k) with the maximum number of outgoing edges in $T^{(k)}$.*

Definition 3. *(Maximin^k) The Maximin^k rule selects a bundle B^* (of size k) that maximizes the minimum weight of any outgoing edge in $W^{(k)}$, over all bundles of size k, i.e.,*

$$B^* \in \arg\min_{B \in \mathcal{A}_k} \max_{(B', B) \in W^{(k)}} N_B(B', B).$$

We remark that our definitions of Copeland^k and Maximin^k were previously defined in [8, 18], where they were referred to as the NED and SEO rules respectively.

Definition 4. *(Ranked Pairs^k) The winner of this mechanism is determined by the following procedure: sort the edges of $W^{(k)}$ by decreasing order of their weights. Construct a directed graph $RP^{(k)}$ with the same vertex set and whose edge set is initially empty. Go over the sorted edges of $W^{(k)}$ and add them to $RP^{(k)}$ as long as it does not create a cycle. Once this concludes, the winner is the bundle in $RP^{(k)}$ that has no incoming edges.*

We conclude by pointing out that all the three mechanisms above reduce to their well-studied single-winner counterparts when $k = 1$. When there exists a Condorcet winner, all of these mechanisms return only that bundle, i.e., they are consistent with the Condorcet criterion for bundling as per Definition 1. Finally, while Copeland^k is not well-defined when there are ties among candidates (an equal number of users preferring two candidates), the presence of ties does not in any affect our Maximin and Ranked Pairs definitions. In Section 3.1, we propose a parameterized generalization of the Copeland bundling rule that allows us to explicitly factor in ties. The Schulze extension is provided in Section 3.4

Computation of Condorcet Winning Bundles.

Even though the number of bundles of size k is exponential in k, we show how to compute the Condorcet winning bundle (when it exists) via a simple algorithm, specifically the algorithm presented in Theorem 1 for computing optimal Copeland bundles. Since the algorithm is Condorcet consistent, it always results in a Condorcet winning bundle when it exists.

3. EFFICIENT ALGORITHMS FOR COMPUTING OPTIMAL BUNDLES

In this section, we present our main technical results, polynomial time algorithms for computing optimal and approximately-optimal bundles based on the Copeland, Maximin, Ranked Pairs, and Schulze rules.

3.1 Copeland and Copeland$^k(\alpha)$

Copeland is among the most popular of Condorcet consistent mechanisms; we are able to provide an efficient algorithm for computing the Copeland^k bundle as per Definition 2 as long as the tournament graph T does not have any ties, i.e., there exist no pair of items such that an equal number of agents prefer both these alternatives. A trivial condition that ensures the absence of ties is the presence of an odd number of agents in $\mathcal{N}$.

When the tournament graph contains ties, the Copeland^k mechanism is no longer well defined. To handle such instances, we consider a generalization of *Copeland*^{k} referred to as the Copeland$^k(\alpha)$ voting rule, originally defined in [12] for the singer-winner setting, where $\alpha \in [0,1]$ denotes the 'weight' given to ties. We extend the Copeland$^k(\alpha)$ definitions to the bundling setting and present algorithms to compute approximately optimal bundles of size k.

We begin with a simple claim that provides an alternative characterization of the optimal Copeland^k bundle and then move on to our main computational results.

Proposition 2. *A bundle B^* is a Copeland^k winner if and only it maximizes the number of pairwise defeats of the items that are not present in the bundle i.e., $B^* \in \arg\max_{B \in \mathcal{A}_k} |x \in B, y \notin B : (x,y) \in T|$.*

It follows from Proposition 2 that the problem of finding a Copeland^k winner can be mathematically formulated as the problem of finding the maximum sized cut $(B, \mathcal{A} \setminus B)$ in the tournament graph T with $|B| = k$, i.e., the cut that maximizes the number of edges going from B to $\mathcal{A} \setminus B$.

Pertinent Notation for Computational Results Formally, given a bundle B, let $\bar{B} = (\mathcal{A} \setminus B)$. Let $out(i)$ denote the outdegree of a node i in T. Let $out(B)$ denote the number of outgoing edges from B in the graph $T^{(k)}$ and suppose that $\tau(i)$ is the number of candidates tied with i and similarly, $\tau(B)$ is the number of neighboring bundles B' of B such that an equal number of users prefer $B \setminus B'$ to $B' \setminus B$, i.e., $\tau(B)$ is the number of ties containing B. Then, for any $0 \leq \alpha \leq 1$, the Copeland(α) score of a candidate i is $out(i) + \alpha\tau(i)$ [12], and we define the Copeland$^k(\alpha)$ score of a bundle B to be $out(B) + \alpha\tau(B)$.

We now present a surprisingly simple algorithm (Algorithm 1) that (i) returns a Copeland^k winning bundle when there are no ties (Theorem 1), (ii) returns a Copeland$^k(\alpha)$ winning bundle for $\alpha = 0.5$, and (iii) is a $\min(2\alpha, \frac{1}{2\alpha})$-approximation to the optimum Copeland$^k(\alpha)$ bundle of size k for any $\alpha \in [0,1]$. (Theorem 3).

Algorithm 1 CopelandWinner

1: **Input:** A profile $\vec{P}$ over $\mathcal{A}$, parameter k.
2: Let T be the tournament graph whose nodes are the candidates in $\mathcal{A}$.
3: Arrange the candidates in decreasing order of their Copeland(α) score $out(i) + \alpha * \tau(i)$.
4: Pick the first k candidates with the largest Copeland(α) score to form bundle B^*.
5: **Output:** The bundle B^*.

Theorem 1. *CopelandWinner (Algorithm 1) computes the Copeland^k winning bundle in polynomial time when there are* no ties *in the tournament graph.*

Observe that when there are no ties in the tournament graph T, the algorithm returns the same bundle B^* irrespective of the value of α.

Proof. Let $\delta(A, B)$ denote the number of (cut) edges going from A to B in the tournament graph T. We prove that the bundle returned by CopelandWinner for profile $\vec{P}$, i.e., B^*, maximizes $\delta(B^*, \bar{B}^*)$ as long as T does not contain any ties. The proof of correctness then follows from Proposition 2. Now, for any bundle B of fixed size k, we have,

$$\sum_{i \in B} out(i) = \delta(B, \bar{B}) + \delta(B, B) = \delta(B, \bar{B}) + \binom{k}{2}.$$

The last term equals $\binom{k}{2}$ as there are exactly that many pairs of candidates within a set of k nodes and for every $i, j \in B$, exactly one of (i,j) or (j,i) belongs to the tournament graph. Therefore, for any B, $\delta(B, \bar{B}) = \sum_{i \in B} out(i) - \binom{k}{2}$. As the latter term is a constant, we can conclude that any set that maximizes the total out-degree of its members must also maximize the number of cut edges. □

3.1.1 Computational Results for Copeland$^k(\alpha)$

Building on our proof of Theorem 1, it is not hard to see that one can efficiently compute a Copeland$^k(0.5)$ winning bundle using Algorithm 1 for $\alpha = 0.5$ when there are ties. Unfortunately, as the following claim indicates, we are not so lucky for other cases. In particular, we show that the decision version of Copeland$^k(\alpha)$ is NP-Complete for $\alpha \in \{0,1\}$.

Claim 2. *For $\alpha \in \{0,1\}$, it is NP-Complete to determine if there exists a bundle of size k whose Copeland$^k(\alpha)$ score is at least l, for some input parameter l.*

Proof. We begin with Copeland$^k(\alpha)$ given $\alpha = 0$, for which the reduction follows from the well known Max-DICUT problem [1] (with given size of parts), where given a directed graph G, and a number ℓ, we are asked if there exists a set B of k nodes such that $\delta(B, \bar{B})$ is at least ℓ. For any given instance of this problem with unweighted graph G, we can use the classic result of McGarvey [6] to construct a polynomially large profile $\vec{P}$ whose tournament graph corresponds to G. After this, it is not hard to see that the Copeland$^k(0)$ winner must also maximize the cut value.

Moving on to Copeland$^k(\alpha)$ for $\alpha = 1$, we turn to the *oneway bisection* problem [13], where we are given a directed graph G with N vertices, and we are asked whether there exists a set S of size $\frac{N}{2}$ with no incoming cut edges, i.e., $\delta(\bar{S}, S) = 0$. Once again, given G, we reduce it to the Copeland$^{\frac{N}{2}}(\alpha)$ problem with the same tournament graph, where for some number l, we are asked whether there is a bundle B of size $\frac{N}{2}$ such that $\delta(B, \bar{B}) + \tau(B, \bar{B})$ is at least l. It is not hard to verify the following statement: there exists a set S of size $\frac{N}{2}$ with no incoming cut edges if and only if the value of the Copeland$^{\frac{N}{2}}(1)$ objective function for the winning bundle is exactly $(\frac{N}{2})^2$.

We conclude by observing that given a bundle, it is easy to compute its Copeland$^k(\alpha)$ score. □

Since we cannot efficiently compute the winning bundle for the above values of α, it is natural to ask whether we can compute bundles that are approximately as good as the optimum bundles. In Theorem 3, we provide two approximation algorithms for this task: the first algorithm is somewhat involved but provides a consistent $\frac{1}{2}$-approximation irrespective of the value of α, whereas the second approach is much simpler and results in improved approximations as long as $\alpha \in (0.25, 1)$. Formally, a bundle B of size k is said to be a c-approximation for $c \leq 1$ to the optimum Copeland$^k(\alpha)$ bundle B^* if

$$out(B) + \alpha\tau(B) \geq c\,(out(B^*) + \alpha\tau(B^*))$$

THEOREM 3. *The following results hold for any given values of α, k*

1. *We can compute in polynomial time a bundle B of size k whose Copeland$^k(\alpha)$ score is a $\frac{1}{2}$-approximation to that of the optimum Copeland$^k(\alpha)$ bundle.*
2. *There is a greedy algorithm that computes a $\min(2\alpha, \frac{1}{2\alpha})$-approximation to optimum Copeland$^k(\alpha)$ bundle of size k.*

Notice that while the greedy algorithm does not provide good approximations as $\alpha \to 0$, in the range $\alpha \in (0.25, 1)$ the guarantee provided is better than that of the consistent 0.5-approximation. Moreover, at $\alpha = 0.5$, we get the optimum bundle of size k, which matches our previous observation.

PROOF. **(Part 1)**: We prove the first part of our theorem using a direct reduction from our problem to the Max DICUT with given sizes of parts (GSP) problem [1]. Given α, construct a complete weighted directed graph G as follows. For any two alternatives x, y, define the weight of the edge (x, y) in G, $w(x, y) = 1$, if a strict majority of the voters prefer x to y and zero if a strict majority prefer y. Define $w(x, y) = w(y, x) = \alpha$ when an equal number of voters prefer x and y (to the other alternative). Now the problem of finding a Copeland$^k(\alpha)$ winning bundle is equivalent to the problem of finding a cut B of size k in the graph G maximizing the weight of the outgoing cut edges. This is the Max DICUT with GSP problem for which a 2-approximation algorithm based on LP rounding was provided in [1].

(**Part 2**) Given an instance (profile of votes), let B denote the optimum Copeland$^k(0.5)$ bundle, which we know from Theorem 1, can be computed efficiently. We make two simple claims: first, for every $\alpha \in [0.5, 1]$, B is a $\frac{1}{2\alpha}$-approximation to the optimum Copeland$^k(\alpha)$ bundle B^* of the same size k; second, for every $\alpha \in [0, 0.5]$, B is a 2α-approximation to the optimum Copeland$^k(\alpha)$ bundle B^* of the same size k. We begin by showing the first claim for $\alpha \geq 0.5$,

$$\begin{aligned} out(B^*) + \alpha\tau(B^*) &\leq 2\alpha(out(B^*) + 0.5\tau(B^*)) \\ &\leq 2\alpha(out(B) + 0.5\tau(B)) \\ &\leq 2\alpha(out(B) + \alpha\tau(B)). \end{aligned}$$

The first inequality and third inequalities are due to α being larger than or equal to 0.5. The second inequality comes from the fact that B is the optimum bundle for the Copeland$^k(0.5)$ objective. Therefore, for $\alpha \geq 0.5$, the objective value of B is at most a factor 2α smaller than that of B^*. Next consider the case when $0 < \alpha \leq 0.5$.

$$\begin{aligned} out(B^*) + \alpha\tau(B^*) &\leq out(B^*) + 0.5\tau(B^*) \\ &\leq out(B) + 0.5\tau(B) \\ &= \frac{1}{2\alpha}(2\alpha out(B) + \alpha\tau(B)) \\ &\leq \frac{1}{2\alpha}(out(B) + \alpha\tau(B)). \end{aligned}$$

The last inequality follows from the fact that $2\alpha \leq 1$ in the given range. This completes the theorem. □

3.2 Maximin

We now move on to the Maximin voting rule for bundles as defined in Definition 3. Since the rule is consistent with the generalized Condorcet criterion, we can efficiently compute the Maximin^k bundle when a Condorcet winner of size k actually exists. Unfortunately, as we show in the following claim, the decision problem corresponding to the computation of a Maximin^k winner is NP-Complete for the general case. Concretely, we define Dec-Maximin^k to be the decision problem of whether exists a bundle B of size k such that every outgoing edge from B in $W^{(k)}$ has a weight of at least ℓ for some input parameter ℓ.

CLAIM 4. *The Dec-Maximin^k problem is NP-Complete.*

PROOF. We first consider a variant of the oneway bisection problem, that we term 'Restricted Oneway Bisection' and show that this problem is NP-Complete via a reduction to the original oneway bisection problem [13]. Following this, we show a simple reduction from the (decision version of the) problem of computing a Maximin^k bundle to the restricted oneway bisection problem.

Restricted Oneway Bisection Let $G = (N, E)$ be any digraph of n vertices where for every pair of vertices u, v, either (i) $(u, v) \in E, (v, u) \notin E$, (ii) $(v, u) \in E, (u, v) \notin E$, or (iii) $(u, v) \notin E, (v, u) \notin E$. Given an instance of a graph G that satisfies the above constraints, we are asked whether there exists a set $S \subseteq N$, $|S| = \frac{n}{2}$ with no incoming edges, i.e., $\delta(\bar{S}, S) = 0$.

LEMMA 5. *The Restricted Oneway Bisection is NP-Complete.*

Proof sketch. Consider the following reduction from the oneway bisection problem to its restricted variant. We are given an arbitrary instance of the oneway bisection problem, a digraph $H = (M, F)$ with m vertices, and edges F. We construct an instance of Restricted Oneway Bisection given by a graph $G = (N, E)$ with $2m$ vertices as follows: for every vertex $u \in M$, add two vertices u_1, u_2 to N. For every pair of vertices u, v, (i) if $(u, v) \in F$, $(v, u) \notin F$, add edges $(u_1, v_1), (u_1, v_2), (u_2, v_2), (u_2, v_1)$ to E, and (ii) if $(u, v) \in F$ and $(v, u) \in F$, add edges $(u_1, v_1), (v_1, u_2), (u_2, v_2), (v_2, u_1)$ to E. We claim that there is a set T of size $\frac{m}{2}$ with no incoming edges in H if and only if there is a set S of size m with no incoming edges in G.

$\Rightarrow$ It is easy to check that if there exists a set T of size $\frac{m}{2}$ in H with no incoming edges, then there is a set S of size m with no incoming edges in G as follows: for all $u \in T$, add u_1, u_2 to S.

$\Leftarrow$ Now, suppose S is a set of size m with no incoming edges in G, we show that there exists a set T of size $\frac{m}{2}$ with no incoming edges in H. Consider the set T constructed as follows: (i) if $u_1 \in S$ and $u_2 \in S$, add u to T, (ii) from the remaining vertices in S, pick exactly half of them arbitrarily and add the corresponding vertices in M to T. It is easy to check that T is of size $\frac{m}{2}$ and has no incoming edges in H. We leave out the details in the interest of space. □

We now prove the claim by providing a reduction from Restricted Oneway Bisection to Dec-Maximin^k. Let an arbitrary instance of the restricted oneway bisection problem be given by a digraph $G = (N, E)$. We construct an instance of Maximin^k by first defining the weighted tournament graph W over the individual alternatives N and then deriving a profile of R agents' preferences based on this graph. If $(u, v) \in E$ and $(v, u) \notin E$, add edges (u, v) and (v, u) to W with weights $\frac{R}{2} + 1$ and $\frac{R}{2} - 1$ respectively. For every other pair u, v add edges (u, v) and (v, u) with weights $\frac{R}{2}$ each. Using the exact same idea as in the proof of McGarvey's Theorem [6], it is easy to construct a profile of R agents whose preferences give rise to this (weighted) tournament graph. Here R will be polynomial in the number of alternatives $|N| = n$. Given this profile of agents' preferences, the Dec-Maximin^k problem involves answering whether there is a bundle B of size k such that every outgoing edge from B in $W^{(k)}$ is of size at least $\frac{R}{2}$.

We claim that there exists a set S of size $\frac{n}{2}$ with no incoming edges in G if and only there exists a bundle B of size $\frac{n}{2}$ such that for every $x \in B, y \notin B$, $N(x, y) \geq \frac{R}{2}$, and therefore every outgoing edge from this bundle in $W^{(k)}$ must have a weight of at least $\frac{R}{2}$.

$\Rightarrow$ Suppose that there exists such a set S. Consider the bundle $B = S$, and assume for the sake of contradiction that there exists an outgoing edge (B, B') of weight less than $\frac{R}{2}$. Moreover, let x and y be the nodes that are present in B and B' respectively but not in the other bundle. Then, by construction of B, it must be that $(y, x) \in G$, a contradiction to our assumption that S has no incoming edges.

$\Leftarrow$ Suppose that there exists a bundle B of size $\frac{n}{2}$ such that every outgoing edge is at least $\frac{R}{2}$. Then, consider the set $S = B$. Suppose that S has an incoming edge $(x, y), x \notin S, y \in S$. Consider the bundle $B' = B - \{y\} + \{x\}$. Then, by construction of B, it must be the case that the weight on the edge (B, B') in $W^{(k)}$ is $\frac{R}{2} - 1$, a contradiction to our assumption on the value of the maximin objective of B. We conclude by observing that given a bundle B, it is easy to compute its maximin objective in polynomial time. $\square$

Approximation Algorithms for Maximin^k: In this section, we circumvent the hardness result by using a different notion of approximation. Specifically, we relax the constraint that bundle size is exactly k and compute a bundle whose size is at most $2k$ and whose Maximin objective (defined below) is at least as good as that of the optimum Maximin^k bundle. Such size-relaxing approximation algorithms are quite common in many optimization problems including Knapsack [26] and Cut problems [2] and capture situations where the central authority (for example a cable provider) has some flexibility in deciding the size of the bundle.

Formally, given any bundle B of size k, we define its Maximin^k score to be the minimum weight over all outgoing edges from B in the graph $W^{(k)}$. Our result follows.

THEOREM 6. *We can compute in polynomial time a bundle B of size $k \leq s \leq 2k$ such that its Maximin^s score is at least that of the Maximin^k score of the optimum bundle B^* of size k.*

In other words, the computed bundle B is 'as good' as the bundle that we seek (B^*) , except with respect to a different (weighted) majority graph. That is, suppose that w^* is the largest number such that at least w^* users prefer the optimum bundle B^* to every other neighboring bundle in $W^{(k)}$. Then the above theorem guarantees that in the graph $W^{(s)}$, at least w^* users prefer B to every other neighboring bundle of size s.

PROOF. **Notation** Let W denote the weighted tournament graph on $\mathcal{A}$ ($W = W^{(1)}$). For any parameter $1 \leq w \leq N$, let $G(w)$ denote an unweighted, directed subgraph of W consisting only of the edges (i, j) such that $N(i, j) < w$. Finally, define $C(w)$ to be the graph of Strongly Connected Components in $G(w)$. For every $C_i \in C(w)$, the set of all components that can be reached from C_i are its successors, and the set of all components from which C_i can be reached are its predecessors. The algorithm is defined in Figure 1.

1. Iterate over $w = N + 1$ to 1 (this step can be made efficient using a binary search)
2. For each w, compute $G(w)$ and $C(w)$. For every component $C_i \in C(w)$ containing more than k nodes, remove C_i and all of its predecessors from $C(w)$.
3. Let $T = T_1, \ldots, T_r$ denote a topological sorting of $C(w)$ (remember that $C(w)$ is a directed acyclic graph)
4. Let ℓ be the largest index s.t. $|T_r \cup T_{r-1} \cup \ldots \cup T_\ell| < k$
5. The winning bundle of weight w, $WB(w)$ is said to be the union of nodes inside the components $T_r, T_{r-1} \ldots, T_{\ell-1}$. (Define $T_0 = \emptyset$).
6. Let w^{max} be the largest value of w for which $WB(w^{max})$ has a size of k or more. Return $B = WB(w^{max})$.

Figure 1: Algorithm MaximinApprox

Consider the optimum bundle B^* and suppose that its Maximin^k score is w^*. Then, in the graph $G(w^*)$, $\delta(B^*, \bar{B^*}) = 0$. This is the idea that we will use in our approximation algorithm, namely to find an appropriate w and a set B that has no outgoing cut edges in $G(w)$.

LEMMA 7. *For any component $C_i \in C(w^*)$, if some node $i \in C_i$ belongs to the optimum bundle B^*, then every node in C_i is in the optimum bundle B^*.*

PROOF. Suppose that this is not the case and there exists $j \in C_i$ which is not in the optimum bundle. By the definition of a strongly connected component, there must be a path from i to j in C_i and by definition of $G(w^*)$, this path only uses edges whose weight in W is strictly smaller than w^*. Therefore, there must be at least one cut edge from B^* to $\mathcal{A} \backslash B^*$ having a weight smaller than w^*, a contradiction. $\square$

LEMMA 8. *For any component $C_i \in C(w^*)$, if $C_i \subseteq B^*$, then every component C_j that is a successor of C_i must also be in the optimum bundle.*

PROOF. This follows from the fact that in $G(w^*)$, there exists a path from some node $i \in C_i$ to some node $j \in C_j$ using only edges smaller than w^*. Therefore, if C_j is not contained in B^*, then there must once again exist a cut edge smaller than w^* going out of B^*, a contradiction. $\square$

COROLLARY 9. *For any component $C_i \subseteq C(w^*)$ that has strictly more than k nodes, neither C_i nor any of its predecessors can belong to B^*.*

The corollary comes from Lemma 7 and the fact that C_i cannot be in the optimum bundle. The above corollary justifies the pruning step, where we remove all components larger than k and their predecessors.

LEMMA 10. *The bundle $B' = WB(w^*)$ satisfies the conditions of the theorem, i.e., $s' = |B'| < 2k$ and its Maximin$^{s'}$ score is at least w^*.*

PROOF. Suppose that $\bar{C}(w^*)$ denotes all the strongly connected components of $G(w^*)$ after the pruning step (removing the large components). Then, by Corollary 9, clearly $B^* \subseteq \bar{C}(w^*)$, and therefore, the components still remaining in $\bar{C}(w^*)$ contain at least k nodes in total. Moreover, for each component $C_i \in \bar{C}(w^*)$, C_i contains at most k nodes. Therefore, look at the topological sort $T = T_1, \ldots T_r$ of $\bar{C}(w^*)$ where for any T_i and T_j with $i < j$, there is no edge in $G(w^*)$ going from T_j to T_i. Therefore, since $B' = T_r \cup T_{r-1} \cup \ldots \cup T_\ell \cup T_{\ell-1}$, it is clear that there are no edges going across the cut from B' in $G(w^*)$. All the edges going across the cut in W have weight at least w^* and so, the Maximin$^{s'}$-score of B' is at least w^*. Moreover, by definition the union of T_r to T_ℓ contains strictly smaller than k nodes and $T_{\ell-1}$ contains at most k nodes, therefore, B' contains at most $2k-1$ items. □

Now, we are ready to complete the proof of the theorem. Suppose that the algorithm outputs the bundle B' with $|B| \geq k$ as per Lemma 10, then we are done. If this is not the case, then as per the definition of MaximinApprox, it is only possible that $w^{max} > w^*$. Using the same argument as in Lemma 10, we infer that the output bundle B can contain between k and $2k$ nodes. By definition, $\delta(B, \bar{B}) = 0$ in the graph $G(w^{max})$, which means that the Maximim^s score of B is w^{max}, which in turn is not smaller than w^*, the Maximin^k score of B^*. □

3.3 Ranked Pairs

Although we defined the Ranked Pairs^k mechanism in a constructive manner, the construction provided in Definition 4 is not efficient. Our main contribution in this section is a polynomial time algorithm for computing the winner of the Ranked Pairs^k mechanism.

Observe the Ranked Pairs^k mechanism is somewhat ambiguously defined in Definition 4; indeed, the mechanism could result in a different output depending on which edge we pick in the tournament graph $W^{(k)}$ in each round. For convenience, we now define a rule for breaking ties consistently in each iteration of the Ranked Pairs^k mechanism so that it yields a unique winning bundle. Following this, we present a simple algorithm that computes the unique winning bundle with respect to this tie-breaking rule. Given two neighboring bundles B, B' of the same size, define the symmetric difference $B \oplus B'$ to be the ordered pair of nodes (x, y) such that $x \in B$ but not B' and $y \in B'$ but not B.

Tie-Breaking Rule: For every pair of candidates (nodes) $x, y \in \mathcal{A}$, we assign a rank $r(x, y)$ to the edge (x, y), which is distinct from $r(y, x)$. Now, the Ranked Pairs^k mechanism as per Definition 4 can be refined as follows: in each iteration, suppose that there are multiple edges of $W^{(k)}$ that satisfy the criteria for selection, then pick all pairs of neighboring bundles (B, B') with the smallest value of $r(B \oplus B')$; go through these chosen pairs in some arbitrary order and add them to G as long as no cycle is induced.

THEOREM 11. *RankedPairsWinner (Algorithm 2) returns the unique winning bundle corresponding to the Ranked Pairs^k mechanism with our previously defined tie-breaking rule in polynomial time.*

Algorithm 2 RankedPairsWinner

1: **Input:** A profile $\vec{P}$ over $\mathcal{A}$, parameter k.
2: Run the traditional ranked pairs algorithm for the $k = 1$ case and let H be the final DAG obtained.
3: Consider the topological ordering $T = T_1, T_2, \ldots, T_m$ of H
4: **Output:** The optimum bundle $B^* = T_1 \cup T_2 \cup \ldots \cup T_k$.

PROOF. Although our algorithm is rather simple, the proof is somewhat involved so we proceed carefully beginning with some notation. Consider the non polynomial time ranked pairs mechanism in Definition 4 along with our tie-breaking rule, and suppose that G_t is the graph constructed by the mechanism at the end of iteration $t-1$. Similarly, define H_t to be the state of the directed acyclic graph obtained at the end of iteration $t-1$ of the traditional ranked pairs algorithm, that is used in Algorithm 2. Note that while the vertices of G_t are the set of k-sized bundles, H_t's vertex set is the set of alternatives in $\mathcal{A}$. We now show the following equivalence between G_t and H_t. For the rest of this proof, we assume that the traditional ranked pairs mechanism for the $k = 1$ case also breaks ties in favor of edges with a smaller rank $r(x, y)$.

LEMMA 12. *For every iteration t, there exists a path between x and y in H_t if and only if there exists a path between every two neighboring bundles B, B' with $B \oplus B' = (x, y)$ in G_t.*

PROOF. The proof proceeds by induction on the iterations $T = 1, 2, \ldots, t, \ldots$. Remember that during each iteration t, both of the algorithms consider the same (x, y) in the tournament graph. The original Ranked Pairs algorithm adds the edge (x, y) to H_t if there is no path from y to x in H_t. Our bundling algorithm adds edges between every B and B' with $B \oplus B' = (x, y)$ as long as there is no path from B' to B in G_t.

The base step is easy to show. Suppose that (x_1, y_1) is the edge considered in the first iteration, then H_2 simply consists of this one edge. G_2 has an edge between every pair of neighboring bundles with $B \oplus B' = (x, y)$ as this cannot induce any cycles. The lemma statement is true trivially true for $t = 1, 2$.

Now suppose that the induction hypothesis is true for H_t, G_t, and in the t^{th} iteration, the edge (x, y) is under consideration. If (x, y) is not added, then $H_{t+1} = H_t$ which means that according to the induction hypothesis, for every B, B' whose symmetric difference is (x, y), there must be a path from B' to B. Therefore, $G_{t+1} = G_t$ as well.

Next, suppose that there is a path between some i and j in H_{t+1} but not H_t. Since (x, y) was the only edge added in this iteration, this means that there must have been a path between $i \to x$ and $y \to j$ in H_t. Let B_i and B_j be any two bundles whose symmetric difference is (i, j). Then, from the induction hypothesis, in G_t, there must have been a path from B_i to B and B' to B_j where $B = B_i - \{i\} + \{x\}$, and $B' = B_j - \{j\} + \{y\}$. Since B and B' are neighboring bundles, G_t does not contain any path from B' to B because otherwise H_t would have a path from y to x which was know is not the case. Therefore, in the t^{th} iteration an edge from B to B' is added to G_t and therefore, G_{t+1} does contain a path from B_i to B_j.

In the opposite direction, suppose that in G_{t+1}, there is a path from some B_i to B_j with $B_i \oplus B_j = (i,j)$ that was not present in G_t. Then we need to prove that $\exists$ a path from i to j in H_{t+1}. First suppose that the path is simply a direct edge, then it can only be the case that $i = x$ and $j = y$ and therefore there is a path from $i \to j$ in H_{t+1} by definition. We can also infer that for any B, B' with a direct edge in G_{t+1}, the edge $B \oplus B'$ also exists in H_{t+1}.

Next, suppose that the path can be described as $B_i B_1 B_2, \ldots, B_r B_j$, where (B_l, B_{l+1}) in the path is a pair of neighboring bundles with a direct edge between them. Construct a new directed graph Z as follows: the node set of this graph is $\mathcal{A}$ and for each B_l, B_{l+1} in the above path, Z consists of the directed edge $(B_l \oplus B_{l+1})$. Moreover, we are allowed to add multiple copies of any edge, i.e., if we have $(B_{l_1} \oplus B_{l_1+1}) = (B_{l_2} \oplus B_{l_2+1}) = (x,y)$, the edge (x,y) is present twice in Z. From what we inferred in the previous paragraph, it is easy to see that every $(x,y) \in Z$ also belongs to H_{t+1} and we conclude that Z is acyclic as well. Without loss of generality, remove all the nodes in Z with zero indegree and out-degree.

Finally, we claim that except for i and j, for every candidate $p \in Z$, its indegree must equal its out degree. Why is this true? Suppose that $B_{l+1} \setminus B_l = \{p\}$, and $p \notin B_i, B_j$, then in some future edge (say $(B_{l'}, B_{l'+1})$ for $l' > l$), we must substitute p for some other candidate, i.e., $B_{l'} \setminus B_{l'+1} = \{p\}$. If $p \in B_i$, then in some past edge (say $(B_{l''}, B_{l''+1})$), we must have substituted some other candidate for p, i.e., $B_{l'+1} \setminus B_{l''} = \{p\}$. Since every candidate in Z except i, j has the same indegree as out-degree, we can form a path (an Eulerian trail) from i to j in Z. Since every edge in Z is also present in H_{t+1}, we conclude that there exists a path from $i \to j$ in H_{t+1}. This completes the proof. $\square$

Lemma 12 is the main workhorse driving our proof of the theorem. Suppose that G is the final directed acyclic graph on bundles obtained by the Ranked Pairs^k Mechanism as per Definition 4. Then, it is enough for us to show that the bundle $B^* = T_1 \cup T_2 \cup \ldots \cup T_k$ (as defined in Algorithm 2) has no incoming edges in G. Note that H and G can be viewed as the final graphs in the sequences H_t and G_t respectively. By definition of the topological sort, for any $j > k$, we know that there is no path from T_j to T_i in H for every $i \in [1,k]$. Since Lemma 12 is an if and only if statement, this means that in G, there is no path from some bundle B to B^*, where $B \oplus B^* = (T_j, T_i)$ for all $T_i \in B^*$. We conclude that there are no incoming edges into B^* in the graph G, and so it is the Ranked Pairs^k winning bundle. $\square$

3.4 Schulze

We now extend the Schulze mechanism to the problem of computing optimum bundles of size k. Let B, B' be any pair of bundles of size k. Suppose that the *strength* of a path $p = B, B_1, \ldots, B_\ell, B'$ in $W^{(k)}$ is the minimum weight of an edge along the path p. Let $S_B(B, B')$ be the maximum strength of any path from B to B' in $W^{(k)}$. Therefore, for every pair of bundles B, B' in $W^{(k)}$, we can define the quantity $S_B(B, B')$.

DEFINITION 5. *The Schulze^k mechanism selects a bundle B^* (of size k) such that for every bundle B' of the same size, we have that $S_B(B^*, B') \geq S_B(B', B^*)$.*

Existence As with the traditional Schulze method, we argue that the relationships guaranteed by $S_B()$ are transitive. Specifically if $S_B(B_1, B_2) > S_B(B_2, B_1)$ and $S_B(B_2, B_3) > S_B(B_3, B_2)$, then $S_B(B_1, B_3) > S_B(B_3, B_1)$. To see why, note that $S_B(B_1, B_3) \geq \min(S_B(B_1, B_2), S_B(B_2, B_3)) = \min(f, g)$ (say). By contradiction, if $S_B(B_3, B_1) \geq \min(f, g)$, then B_3 has a path to B_2 that is at least $\min(f, g)$ and B_2 also has a path to B_1 that is at least $\min(f, g)$, which contradicts our assumption that $S_B(B_1, B_2) > S_B(B_2, B_1)$ and $S_B(B_2, B_3) > S_B(B_3, B_2)$. By this transitivity, we get that the set of Schulze^k winners is non-empty.

Our main result is a simple algorithm to compute a Schulze^k winning bundle. Before stating the result, we abuse notation and define $S(x, y)$ for any pair of alternatives $x, y \in \mathcal{A}$ to be the maximum strength of any path from x to y in W.

THEOREM 13. *For any instance and parameter k, we can compute a Schulze^k optimal bundle B^* in polynomial time.*

Algorithm: Consider the following unweighted, directed graph G on the set of alternatives $\mathcal{A}$: for any x, y, there exists an edge from x to y if $S(x,y) > S(y,x)$. By the transitivity of the function $S()$, the graph is clearly acyclic. The algorithm proceeds by repeatedly removing a vertex from G that has no incoming edges until we remove k nodes to form a bundle B^*. It is not hard to deduce that $\delta(\bar{B}^*, B^*) = 0$ in G.

To show that B^* is the winning bundle, we only need to prove that for any other bundle B, $S(B^*, B) \geq S(B, B^*)$. Suppose by contradiction that this is not the case, then there must exist some $x \in B^* \setminus B$ and $y \in B \setminus B^*$ such that $S(y,x) > S(x,y)$. However, this contradicts the fact that $\delta(\bar{B}^*, B^*) = 0$. Therefore, B^* is optimal.

4. CONCLUSIONS

Building on previous notions of Condorcet winning committees [8, 16, 25], we provide a generic template for extending Condorcet consistent social choice rules to the bundling setting. Although our definitions are inspired by previous works, the main focus of this paper is on computationally efficient methods to output optimal bundles based on extensions of popular Condorcet consistent methods such as Copeland, Ranked Pairs, and Schulze, and approximately optimal solutions for Copeland$^\alpha$ and Maximin generalizations, where it is NP-Hard to compute the optimum bundles. Our work presents the first known polynomial time algorithm for committee selection based on Condorcet's criterion and indicates that it is possible to select bundles that explicitly appeal to a majority of the population in contrast to previous work where the focus was on utilitarian welfare maximization [10].

Acknowledgments

We would like to thank the anonymous reviewers for their very detailed and helpful comments. Lirong Xia acknowledges National Science Foundation under grant IIS-1453542 for support. Shreyas was partly supported by National Science Foundation under grant CNS-1218374.

REFERENCES

[1] Alexander A. Ageev, Refael Hassin, and Maxim Sviridenko. A 0.5-approximation algorithm for MAX DICUT with given sizes of parts. *SIAM J. Discrete Math.*, 14(2):246–255, 2001.

[2] Konstantin Andreev and Harald Räcke. Balanced graph partitioning. *Theory Comput. Syst.*, 39(6):929–939, 2006.

[3] Haris Aziz, Markus Brill, Vincent Conitzer, Edith Elkind, Rupert Freeman, and Toby Walsh. Justified representation in approval-based committee voting. *Social Choice and Welfare*, pages 1–25.

[4] Haris Aziz, Serge Gaspers, Joachim Gudmundsson, Simon Mackenzie, Nicholas Mattei, and Toby Walsh. Computational aspects of multi-winner approval voting. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, May 4-8, 2015*, pages 107–115, 2015.

[5] Salvador Barberà and Danilo Coelho. How to choose a non-controversial list with k names. *Social Choice and Welfare*, pages 79–96, 2008.

[6] Felix Brandt, Markus Brill, and Paul Harrenstein. Tournament solutions. *Handbook of Computational Social Choice*, pages 57–84, 2016.

[7] John R Chamberlin and Paul N Courant. Representative deliberations and representative decisions: Proportional representation and the borda rule. *American Political Science Review*, 77(03):718–733, 1983.

[8] Danilo Coelho. *Understanding, evaluating and selecting voting rules through games and axioms.* Universitat Autònoma de Barcelona,, 2005.

[9] Vincent Conitzer, Jérôme Lang, and Lirong Xia. Hypercubewise preference aggregation in multi-issue domains. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[10] Edith Elkind, Piotr Faliszewski, Piotr Skowron, and Arkadii Slinko. Properties of multiwinner voting rules. In *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '14, Paris, France, May 5-9, 2014*, pages 53–60, 2014.

[11] Edith Elkind, Jérôme Lang, and Abdallah Saffidine. Condorcet winning sets. *Social Choice and Welfare*, 44(3):493–517, 2015.

[12] Piotr Faliszewski, Edith Hemaspaandra, and Henning Schnoor. Copeland voting: ties matter. In *7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), Estoril, Portugal, May 12-16, 2008, Volume 2*, pages 983–990, 2008.

[13] Uriel Feige and Orly Yahalom. On the complexity of finding balanced oneway cuts. *Inf. Process. Lett.*, 87(1):1–5, 2003.

[14] Peter C Fishburn. Condorcet social choice functions. *SIAM Journal on applied Mathematics*, 33(3):469–489, 1977.

[15] Peter C Fishburn. An analysis of simple voting systems for electing committees. *SIAM Journal on Applied Mathematics*, 41(3):499–502, 1981.

[16] William V Gehrlein. The condorcet criterion and committee selection. *Mathematical Social Sciences*, 10(3):199–209, 1985.

[17] Janusz Kacprzyk, Hannu Nurmi, and Sławomir Zadrożny. The role of the owa operators as a unification tool for the representation of collective choice sets. In *Recent Developments in the Ordered Weighted Averaging Operators: Theory and Practice*, pages 149–166. Springer, 2011.

[18] Eric Kamwa. Stable rules for electing committees and divergence on outcomes. *Group Decision and Negotiation*, pages 1–18, 2016.

[19] D Marc Kilgour. Approval balloting for multi-winner elections. In *Handbook on approval voting*, pages 105–124. Springer, 2010.

[20] Jérôme Lang and Lirong Xia. Voting in combinatorial domains. *Handbook of Computational Social Choice*, pages 197–222, 2016.

[21] Tyler Lu and Craig Boutilier. Budgeted social choice: From consensus to personalized decision making. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, 2011.

[22] Burt L Monroe. Fully proportional representation. *American Political Science Review*, 89(04):925–940, 1995.

[23] Joel Oren and Brendan Lucier. Online (budgeted) social choice. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI)*, 2014.

[24] Ariel D. Procaccia, Jeffrey S. Rosenschein, and Aviv Zohar. On the complexity of achieving proportional representation. *Social Choice and Welfare*, 30(3):353–362, 2008.

[25] Thomas C Ratliff. Some startling inconsistencies when electing committees. *Social Choice and Welfare*, 21(3):433–454, 2003.

[26] David B. Shmoys and Éva Tardos. An approximation algorithm for the generalized assignment problem. *Math. Program.*, 62:461–474, 1993.

[27] Piotr Skowron, Piotr Faliszewski, and Jerome Lang. Finding a collective set of items: from proportional multirepresentation to group recommendation. In *Proceedings of the First Annual National Conference on Artificial Intelligence (AAAI)*, pages 2131–2137. AAAI Press, 2015.

[28] Lirong Xia, Vincent Conitzer, and Jérôme Lang. Voting on multiattribute domains with cyclic preferential dependencies. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 202–207, 2008.

[29] Ronald R Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on systems, Man, and Cybernetics*, 18(1):183–190, 1988.