5

Computing Manipulations is “Usually” Easy

We have seen in the last chapter that computing a manipulation is NP-complete for maximin and ranked pairs. In particular, the coalitional manipulation problem is NP-complete for ranked pairs even for one manipulator. This property was only shown previously for STV and second-order Copeland. In Table 3.1 we observe that computational complexity can serve as a barrier for many common voting rules when there are two or more manipulators. In this chapter, we will prove that computational complexity is not a strong barrier against manipulation for almost all common voting rules. This argument will be supported by two approaches. In Section 5.2 we pursue the “frequency of manipulability” approach, that is, the votes are randomly generated i.i.d. according to some distribution over all linear orders. We will show that with a high probability the UCM problem (Definition 3.1.2) is computationally trivial. In Section 5.4 we pursue an approximation approach. More precisely, we focus on approximating the UCO problem (Definition 3.1.3), and propose an algorithm that approximates UCO with an additive error that only depends on the number of alternatives (but not on the number of voters) for all positional scoring rules.

Instead of proving the results one by one for common voting rules, we take unified approaches. In Section 5.1 we introduce a general framework called *generalized scoring rules*, and then characterize the frequency of manipulability for any generalized scoring rule in Section 5.2. To show how general this class of voting rules is, we give a concise axiomatic axiomatization in Section 5.3. In Section 5.4 we will design an approximation algorithm that works for any positional scoring rule, in light of a novel relationship between UCO and a scheduling problem.

5.1 Generalized Scoring Rules

A generalized scoring rule (GSR) associates a vector of k real numbers with every vote, for some k that depends on (but is not necessarily equal to) m. The decision that the rule makes is based only on the sum of these vectors. Even more specifically, the decision is based only on comparisons among the components in this sum. That is, if we know, for every $i, j \in \{1, \ldots, k\}$, whether the ith component in the sum is larger than the jth component, the jth is larger than the ith, or they are the same, then we know enough to determine the winner. Sometimes, the components can be partitioned so that the decision only depends on comparisons within elements of the partition, which will be helpful.

Let $k \in \mathbb{N}$, and let $\mathscr{K} = \{K_1, \ldots, K_q\}$ be a partition of $K = \{1, \ldots, k\}$. That is, for any $i \leqslant q$, $K_i \subseteq K$, $K = \bigcup_{l=1}^{q} K_l$, and for any $i, j \leqslant q$, $i \neq j$, $K_i \cap K_j = \varnothing$. We say that two vectors of length k are equivalent with respect to a partition if, within each element of the partition, they agree on which components are larger.

Definition 5.1.1. *Let $\mathscr{K}$ be a partition of K. For any $a, b \in \mathbb{R}^k$, we say that a and b are* equivalent with respect to $\mathscr{K}$*, denoted by $a \sim_{\mathscr{K}} b$, if for any $l \leqslant q$, any $i, j \in K_l$, $a_i \geqslant a_j \Leftrightarrow b_i \geqslant b_j$ (where a_i denotes the ith component of the vector a,* etc.*).*

For two partitions $\mathscr{K} = \{K_1, \ldots, K_q\}$ and $\mathscr{K}' = \{K'_1, \ldots, K'_p\}$, $\mathscr{K}'$ is a *refinement* of $\mathscr{K}$ if for any $l \leqslant q$, any $l' \leqslant p$, $K'_{l'} \cap K_l$ is either $K'_{l'}$ or $\varnothing$. That is, $\mathscr{K}'$

is obtained from $\mathscr{K}$ by partitioning the sets in $\mathscr{K}$. In this case, we say that $\mathscr{K}$ is *coarser* than $\mathscr{K}'$, and $\mathscr{K}'$ is *finer* than $\mathscr{K}$.

Proposition 5.1.2. *For any partitions $\mathscr{K}$, $\mathscr{K}'$ such that $\mathscr{K}'$ is a refinement of $\mathscr{K}$, and any $a, b \in \mathbb{R}^k$, if $a \sim_{\mathscr{K}} b$, then $a \sim_{\mathscr{K}'} b$.*

We note that $\{K\}$ (the partition that only contains K itself) is the coarsest partition.

Definition 5.1.3. *Let $\mathscr{K}$ be a partition of K. A function $g : \mathbb{R}^k \to \mathcal{C}$ is* compatible with $\mathscr{K}$ *if for any $a, b \in \mathbb{R}^k$, $a \sim_{\mathscr{K}} b \Rightarrow g(a) = g(b)$.*

That is, for any mapping g that is compatible with $\mathscr{K}$, $g(a)$ is determined (only) by comparisons within each K_l, $l \leqslant q$. Namely, we do not need to compare components across different elements of the partition.

Now we are ready to define generalized scoring rules.

Definition 5.1.4. *Let $k \in \mathbb{N}$, $f : L(\mathcal{C}) \to \mathbb{R}^k$ and $g : \mathbb{R}^k \to \mathcal{C}$, where g is compatible with $\mathscr{K}$. f and g determine the* generalized scoring rule $GS(f, g)$ *as follows. For any profile of votes $V_1, \ldots, V_n \in L(\mathcal{C})$, $GS(f, g)(V_1, \ldots, V_n) = g(\sum_{i=1}^{n} f(V_i))$. We say that $GS(f, g)$ is* of order k, *and* compatible with $\mathscr{K}$.

From Proposition 5.1.2 we know that for any partitions $\mathscr{K}$, $\mathscr{K}'$ such that $\mathscr{K}'$ is a refinement of $\mathscr{K}$, $GS(f, g)$ is compatible with $\mathscr{K}'$, then $GS(f, g)$ is also compatible with $\mathscr{K}$. Given a profile P of votes, we use $f(P)$ as shorthand for $\sum_{V \in P} f(V)$. We will call $f(P)$ the *total generalized score vector*. By definition, any unweighted generalized scoring rule satisfies *anonymity* (that is, every voter is treated equally) and *homogeneity* (that is, if we add any number of copies of the profile to the profile, the winner does not change). Any generalized scoring rule is compatible with the partition $\{K\}$. Nevertheless, being compatible with $\{K\}$ is not vacuous: if we modified the definition so that g is not required to be compatible with any partition, then any anonymous voting rule would belong to the resulting class of rules. If a generalized scoring rule is compatible with a partition, this effectively means that, within each element of the partition, the scores are of the same "type," so that we can compare them.

We now illustrate how general the class of generalized voting rules is by showing how some standard rules belong to the class. Many other rules can also be shown to belong to the class.

Proposition 5.1.5. *All positional scoring rules, Copeland, STV, maximin, ranked pairs, and Bucklin are generalized scoring rules.*

Proof of Proposition 5.1.5: We explicitly give $k, f, g, \mathscr{K}$ for each of these rules. In the remainder of the proof, the number of alternatives is fixed to be m. Let $V \in L(\mathcal{C})$ be a vote, and let P be a profile of votes. To simplify the construction, we will not specify how ties are broken when we describe these rules as generalized scoring rules. It is easy to incorporate the tie-breaking mechanism to define the g function for all these voting rules.

Positional scoring rules: Suppose the scoring vector for the rule is $\vec{s}_m = (s_m(1), \ldots, s_m(m))$. The total generalized score vector will simply consist of the total scores of the individual alternatives. Let

- $k_{\vec{s}_m} = m$.
- $f_{\vec{s}_m}(V) = (\vec{s}_m(V, c_1), \ldots, \vec{s}_m(V, c_m))$.
- $g_{\vec{s}_m}(f_{\vec{v}}(P)) = \arg\max_i (f_{\vec{v}}(P))_i$.
- $\mathscr{K}_{\vec{s}_m} = \{K\}$.

Copeland: For Copeland, the total generalized score vector will consist of the scores in the pairwise elections. Let

- $k_{Copeland} = m(m-1)$; the components are indexed by pairs (i,j) such that $i,j \leqslant m$, $i \neq j$.

- $(f_{Copeland}(V))_{(i,j)} = \begin{cases} 1 & \text{if } c_i \succ_V c_j \\ 0 & \text{otherwise} \end{cases}$

- $g_{Copeland}$ selects the winner based on $f_{Copeland}(P)$ as follows. For each pair $i \neq j$, if $(f_{Copeland}(P))_{(i,j)} > (f_{Copeland}(P))_{(j,i)}$, then add 1 point to i's Copeland score; if $(f_{Copeland}(P))_{(j,i)} > (f_{Copeland}(P))_{(i,j)}$, then add 1 point to j's Copeland score; if tied, then add 0.5 to both i's and j's Copeland scores. The winner is the alternative that gets the highest Copeland score.

- $q_{Copeland} = \frac{m(m-1)}{2}$ (we recall that q is the number of elements in the partition). The elements of the partition are indexed by (i,j), $i < j$. For any $l = (i,j)$, $i < j$, let $K_l = \{(i,j),(j,i)\}$. Let $\mathscr{K}_{Copeland} = \{K_l : l = (i,j), i < j\}$.

STV: For STV, we will use a total generalized score vector with many components. For every proper subset S of alternatives, for every alternative c outside of S, there is a component in the vector that contains the number of times that c is ranked first if all of the alternatives in S are removed. Let

- $k_{STV} = \sum_{i=0}^{m-1} \binom{m}{i}(m-i)$; the components are indexed by (S,j), where S is a proper subset of $\mathcal{C}$ and $j \leqslant m, c_j \notin S$.

- $(f_{STV}(V))_{(S,j)} = 1$, if after removing S from V, c_j is at the top; otherwise, let $(f_{STV}(V))_{(S,j)} = 0$.

- g_{STV} selects the winner based on $f_{STV}(P)$ as follows. In the first round, find $j_1 = \arg\min_j((f_{STV}(P))_{(\varnothing,j)})$. Let $S_1 = \{c_{j_1}\}$. Then, for any $2 \leqslant i \leqslant m-1$, define S_i recursively as follows: $S_i = S_{i-1} \cup \{j_i\}$, where $j_i = \arg\min_j(f_{STV}(P)_{(S_{i-1},j)})$; finally, the winner is the unique alternative in $\mathcal{C} - S_{m-1}$.

- $q_{STV} = 2^m - 1$. The elements of the partition are indexed by the $S \subset \mathcal{C}$. For any $S \subset \mathcal{C}$, let $K_S = \{(S, j) : c_j \notin S\}$. Let $\mathscr{K}_{STV} = \{K_S : S \subset \mathcal{C}\}$.

Maximin: For maximin, we use the same total generalized score vector as for Copeland, that is, the vector of all scores in pairwise elections. Let

- $k_{maximin} = m(m-1)$; the components are indexed by pairs (i, j) such that $i, j \leqslant m$, $i \neq j$.

- $(f_{maximin}(V))_{(i,j)} = \begin{cases} 1 & \text{if } c_i \succ_V c_j \\ 0 & \text{otherwise} \end{cases}$

- $g_{maximin}(f_{maximin}(P))$ is the c_i such that for any $i' \leqslant m$, $i' \neq i$, there exists $j' < m$, $j' \neq i'$ such that for any $j \leqslant m$, $j \neq i$, we have $f_{maximin}(P)_{(i,j)} > (f_{maximin}(P))_{(i',j')}$.

- $\mathscr{K}_{maximin} = \{K\}$.

Ranked pairs: We use the same total generalized score vector as for Copeland and maximin, that is, the vector of all scores in pairwise elections. Let

- $k_{rp} = m(m-1)$; the components are indexed by pairs (i, j) such that $i, j \leqslant m$, $i \neq j$.

- $(f_{rp}(V))_{(i,j)} = \begin{cases} 1 & \text{if } c_i \succ_V c_j \\ 0 & \text{otherwise} \end{cases}$

- g_{rp} selects the winner based on $f_{rp}(P)$ as follows. In each step, we consider a pair of alternatives c_i, c_j that we have not previously considered; specifically, we choose the remaining pair with the highest $(f_{rp}(P))_{(i,j)}$. We then fix the order $c_i > c_j$, unless this contradicts previous orders that we fixed (that is, it violates transitivity). We continue until we have considered all pairs of alternatives. The alternative at the top of the ranking wins.

- $\mathscr{K}_{rp} = \{K\}$.

Bucklin: For Bucklin, the total generalized score vector will have one component for every combination of an alternative and a position; this component contains the number of times that that alternative is ranked either in that position or in a higher position. We only need to consider positions from 1 through $m-1$. Let

- $k_{Bucklin} = 2m(m-1)$; the components are indexed by $(i,j)^1$ and $(i,j)^2$, $i \leqslant m-1$, $j \leqslant m$.

- $f_{Bucklin}(V)_{(i,j)^1} = 1$ and $f_{Bucklin}(V)_{(i,j)^2} = 0$ if c_j is ranked among the top i alternatives in V; otherwise $f_{Bucklin}(V)_{(i,j)^1} = 0$ and $f_{Bucklin}(V)_{(i,j)^2} = 1$.

- $g_{Bucklin}(f_{Bucklin}(P))$ is the c_j such that there exists $i \leqslant m$, $(i,j)^1 > (i,j)^2$, and for any $j' \neq j$, $(i,j')^2 \geqslant (i,j)^1$.

- $q_{Bucklin} = m-1$. For any $l \leqslant m-1$, let $K_l = \{(l,j)^1, (l,j)^2 : j \leqslant m\}$. Let $\mathscr{K}_{Bucklin} = \{K_l : l \leqslant m\}$.

□

We have shown that STV is a generalized scoring rule in the proof. In fact, we can generalize this and show that *any* multiround run-off process where in each round, alternatives are eliminated according to a generalized scoring rule (to be precise, a correspondence) must itself be a generalized scoring rule. For example, for STV, the voting rule that only eliminates one alternative (the alternative that has the lowest plurality score among all remaining alternatives) is used in every round. As another example, for Baldwin's rule, a reverse version of Borda that only eliminates one alternative (the alternative that has the lowest Borda score among all remaining alternatives) is used in every round. The proof can be found in the appendix of Xia and Conitzer (2008b), and is omitted here.

5.2 Frequency of Manipulability for Generalized Scoring Rules

Let π be a probability distribution over $L(\mathcal{C})$ that is positive everywhere. For any $n^* \in \mathbb{N}$, let ϕ_{π,n^*} be the distribution over profiles of n^* voters in which each vote is drawn i.i.d. according to π. Given a manipulation instance (r, P^{NM}, c, n'), if there is only one possible winner, then we say that this manipulation instance is *closed*; otherwise we say this manipulation instance is *open* (Procaccia and Rosenschein, 2007a).

Definition 5.2.1. *A manipulation instance* (r, P^{NM}, c, n') *is* closed *if for any profiles* P_1^M, P_2^M *for the manipulators,* $r(P^{NM} \cup P_1^M) = r(P^{NM} \cup P_2^M)$*. An instance is* open *if it is not closed.*

We note that in the above definition, whether a UCM instance is open or closed does not depend on the choice of c. That is, for any $c, c' \in \mathcal{C}$, (r, P^{NM}, c, n') is open (respectively, closed) if and only if (r, P^{NM}, c', n') is open (respectively, closed). Procaccia and Rosenschein (2007a) have shown that, suppose the following four conditions are satisfied.

1. The rule is a positional scoring rule,
2. the number of manipulators $|M|$ is $o(\sqrt{n})$,
3. the votes are drawn independently, and
4. there exists $d > 0$ such that for each vote's distribution, the variance of the difference in scores for any pair of alternatives is at least d.

Then, when $n \to \infty$, the probability that a weighted manipulation instance is open is 0. In this section, we generalize this result to generalized scoring rules; in addition, we characterize the rate of convergence to 0. However, unlike Procaccia and

Rosenschein, we do assume that votes are drawn i.i.d.; this is needed to obtain the convergence rate. Hence, strictly speaking, our result is not a generalization of their result. We can also obtain a strict generalization of Procaccia and Rosenschein's results to generalized scoring rules, but without proving a convergence rate; we will not do so in this paper.

5.2.1 Conditions under Which Coalitional Manipulability is Rare

In this section, we study the probability that a manipulation instance is open when there are $O(n^{\alpha})$ $(0 \leqslant \alpha < \frac{1}{2})$ manipulators, and the nonmanipulator votes are drawn i.i.d. Let $n^* = |P^{NM}|$ denote the number of nonmanipulators. Then, n is the total number of voters, $n^* + n'$ (nonmanipulators and manipulators). We will prove that for any generalized scoring rule, this probability is $O(\frac{1}{\sqrt{n}})$. Let $T(r, m, n, \pi, n')$ denote this probability. That is, let c be an arbitrary alternative,

$$T(r, m, n, \pi, n') = Pr_{P^{NM} \sim \phi_{\pi, n^*}}\{(r, P^{NM}, c, n') \text{ is open}\}$$

Lemma 5.2.2. *Let $N \in \mathbb{N}$. Let $Y_1, \ldots, Y_N$ be i.i.d. random variables with $E(Y_1) < \infty$, $E((Y_1 - E(Y_1))^2) > 0$, and $E(|Y_1 - E(Y_1)|^3) < \infty$. Let $Y = \sum_{\zeta=1}^{N} Y_\zeta$. For any constant $0 \leqslant p < \frac{1}{2}$ that does not depend on N, and any function $f(N)$ that is $\Omega(1)$, we have that $Pr(|Y| \leqslant f(N))$ is $O(\frac{f(N)}{\sqrt{N}})$.*

Proof of Lemma 5.2.2: Let $\Phi(x)$ be the cumulative distribution function of the standard normal distribution $N(0, 1)$. Let $\sigma^2 = E((Y_1 - E(Y_1))^2)$, $\rho = E(|Y_1 - E(Y_1)|^3)$. Then we have:

$$\begin{aligned} & Pr(|Y| < f(N)) \\ =& Pr\left(-\frac{E(Y_1)N}{\sigma\sqrt{N}} - \frac{f(N)}{\sigma\sqrt{N}} < \frac{Y - E(Y_1)N}{\sigma\sqrt{N}} < -\frac{E(Y_1)N}{\sigma\sqrt{N}} + \frac{f(N)}{\sigma\sqrt{N}}\right) \end{aligned}$$

Then by the Berry-Esséen theorem (Durrett, 1991),

$$
\begin{aligned}
& Pr(|Y| < f(N)) \\
<& \Phi(-\frac{E(Y_1)N}{\sigma\sqrt{N}} + \frac{f(N)}{\sigma\sqrt{N}}) - \Phi(-\frac{E(Y_1)N}{\sigma\sqrt{N}} - \frac{f(N)}{\sigma\sqrt{N}}) + \frac{C\rho}{\sigma^3\sqrt{N}} \\
=& \int_{-\frac{E(Y_1)N}{\sigma\sqrt{N}} - \frac{f(N)}{\sigma\sqrt{N}}}^{-\frac{E(Y_1)N}{\sigma\sqrt{N}} + \frac{f(N)}{\sigma\sqrt{N}}} N(0,1)(x)dx + \frac{C\rho}{\sigma^3\sqrt{N}} \\
<& \frac{2f(N)}{\sigma\sqrt{N}} \times \frac{1}{\sqrt{2\pi}} + \frac{C\rho}{\sigma^3\sqrt{N}}
\end{aligned}
$$

which is $O(\frac{f(N)}{\sqrt{N}})$, because C is a constant that does not depend on N and $f(N) = \Omega(1)$. □

Theorem 5.2.3. *Let $r = GS(f,g)$ be a generalized scoring rule of order k. For any $m \in \mathbb{N}$, any constant $0 \leqslant \alpha < \frac{1}{2}$, and any constant h (where both m and h do not depend on n), there exists a constant $t_{m,\alpha,h} > 0$ (that does not depend on n) such that if $n' \leqslant hn^{\alpha}$, then*

$$T(r, m, n, \pi, n') \leqslant t_{m,\alpha,h} n^{\alpha - \frac{1}{2}}$$

Proof of Theorem 5.2.3: We recall that each vote is drawn i.i.d. according to π. For any pair $i_1, i_2 \leqslant k$, $i_1 \neq j_2$, and any $t > 0$, let

$$R(i_1, i_2, t, \pi, n') = Pr\{|(f(P^{NM}))_{i_1} - (f(P^{NM}))_{i_2}| \leqslant t\}$$

We recall that $(f(P^{NM}))_i$ is the ith component of $f(P^{NM})$. In other words, $R(i_1, i_2, t, \pi, n')$ is the probability of profiles of nonmanipulators' votes P^{NM} such that the difference between the i_1th component and the i_2th component of $f(P^{NM})$ is no more than t, when each vote is drawn i.i.d. according to π. We also recall that $n^* = |P^{NM}|$. Let $Y_1^{i_1,i_2}, \ldots, Y_{n^*}^{i_1,i_2}$ be n^* i.i.d. random variables, where the distribution for each $Y_\zeta^{i_1,i_2}$ is the same as the distribution for $(f(V))_{i_1} - (f(V))_{i_2}$, where V is drawn according to π. That is, for any $V \in L(\mathcal{C})$, with probability $\pi(V)$, $Y_1^{i_1,i_2}$ takes value $(f(V))_{i_1} - (f(V))_{i_2}$. Let $Y^{i_1,i_2} = \sum_{\zeta=1}^{n^*} Y_\zeta$.

Let $v_{max} = \max_{i \leqslant k, V \in L(\mathcal{C})} (f(V))_i$. That is, v_{max} is the maximum component of all score vectors corresponding to a single vote. We note that v_{max} is a constant that does not depend on n. We also note that since n' is $O(n^\alpha)$ and $\alpha < \frac{1}{2}$, it must be that n^* is $\Omega(n)$, so that n is $O(n^*)$, $v_{max}hn^\alpha$ is $O((n^*)^\alpha)$. Therefore, by Lemma 5.2.2 (in which we let $N = n^*$), we know that $Pr(|Y^{i_1,i_2}| \leqslant v_{max}hn^\alpha)$ is $O(\frac{v_{max}hn^\alpha}{\sqrt{n^*}}) = O((n^*)^{\alpha-\frac{1}{2}})$, so it is $O(n^{\alpha-\frac{1}{2}})$. Hence, there exists a constant t_{i_1,i_2} such that

$$Pr(|Y^{i_1,i_2}| \leqslant v_{max}hn^\alpha) < t_{i_1,i_2} n^{\alpha-\frac{1}{2}}$$

We let $t_{max} = \max_{i,j \leqslant k, i \neq j} t_{i,j}$. If a manipulation instance is open, then there exists a profile P^M for the manipulators such that $GS(f,g)(P^M \cup P^{NM}) \neq GS(f,g)(P^{NM})$, which means that $f(P^M \cup P^{NM}) \not\sim f(P^{NM})$. In this case there must exist i, j, $i \neq j$, such that $Pr(|(f(P^{NM}))_i - (f(P^{NM}))_j| \leqslant v_{max}n^*) \leqslant v_{max}hn^\alpha$. Therefore,

$$T(GS(f,g), m, n, \pi, n') \leqslant \sum_{1 \leqslant i < j \leqslant m} R(i, j, v_{max}hn^\alpha, \pi, n')$$

We note that $R(i, j, v_{max}hn^\alpha, \pi, n') = Pr(|Y^{i,j}| \leqslant v_{max}hn^\alpha)$. Therefore, we have

$$T(GS(f,g), m, n, \pi, n') \leqslant \sum_{i \neq j} R(i, j, v_{max}hn^\alpha, \pi, n')$$

$$\leqslant \sum_{i \neq j} t_{i,j} n^{\alpha-1} \leqslant \frac{k(k-1)}{2} t_{max} n^{\alpha-\frac{1}{2}}$$

Let $t_{m,\alpha,h} = \frac{k(k-1)}{2} t_{max}$. We know that $t_{m,\alpha,h}$ is a constant that does not depend on n.

(End of the proof for Theorem 5.2.3.) □

From Proposition 5.1.5 and Theorem 5.2.3, we obtain the following corollary.

Corollary 5.2.4. *Let r be any positional scoring rule, Copeland, STV, maximin, ranked pairs, or Bucklin. For any $m \in \mathbb{N}$, any constant $0 \leqslant \alpha < \frac{1}{2}$, and any constant h (where m, α, and h do not depend on n), there exists a constant $t_{m,\alpha,h} > 0$ (that does not depend on n) such that if $n' \leqslant hn^{\alpha}$, then*

$$T(r, m, n, \pi, n') \leqslant t_{m,\alpha,h} n^{\alpha - \frac{1}{2}}$$

A profile is said to be *tied* if a single additional voter can change the outcome. By letting $\alpha = 0$ and $h = 1$ in Theorem 5.2.3, we have that for any generalized scoring rule and any fixed m, the number of tied profiles is $O(\frac{1}{\sqrt{n}})$.

5.2.2 Conditions under which Coalitions of Manipulators are All-Powerful

Let us consider a positional scoring rule and a distribution over nonmanipulator votes. Furthermore, let us consider each alternative's expected score; let $\mathcal{C}_{max}$ be the set of alternatives with the highest expected score. Procaccia and Rosenschein (2007a) have shown that, suppose the following conditions hold.

1. The number of manipulators is in both $\omega(\sqrt{n})$ and $o(n)$, and

2. votes are drawn i.i.d.

Then, the probability that the manipulators can make any alternative in $\mathcal{C}_{max}$ win converges to 1 as $n \to \infty$. Hence, assuming $|\mathcal{C}_{max}| > 1$, the probability that the instance is open converges to 1 (however, if $|\mathcal{C}_{max}| = 1$, it converges to 0).

In this section, we prove a similar result for generalized scoring rules; in addition, we characterize the rate of convergence to 0. (In fact, in this case, Procaccia and Rosenschein also characterize this rate—for positional scoring rules.)

Specifically, in this section, we study the case where the number of manipulators is $\Omega(n^{\alpha})$ ($\frac{1}{2} < \alpha < 1$) and $o(n)$, the votes are drawn i.i.d. according to π, and a generalized scoring rule is used. We provide a sufficient condition under which the manipulators can make any alternative in a particular set of alternatives win with probability $1 - O(e^{-\Omega(n^{2\alpha-1})})$. (We need the $o(n)$ assumption for a technical reason. If $n' = \Theta(n)$, then obviously the probability that the manipulators are all-powerful is higher.)

Definition 5.2.5. *π is* compatible with *$\mathscr{K}$ w.r.t. f, if, for $V \sim \pi$, for any $l \leqslant q$, any $i, j \in K_l$ ($i \neq j$), $E((f(V))_i) = E((f(V))_j)$.*

That is, π is compatible with $\mathscr{K}$ w.r.t. f if within each element of the partition $\mathscr{K}$, the expectation of the components of $f(V)$ are the same (where V is drawn according to π).

Given $GS(f, g)$, it will be useful to have a profile P such that for some partition $\mathscr{K}$ that $GS(f, g)$ is compatible with, the components of $f(P)$ within each K_l ($l \leqslant q$) are all different. The next definition makes this precise.

Definition 5.2.6. *For any $GS(f, g)$ compatible with $\mathscr{K}$, a profile P is said to be* distinctive *w.r.t. $GS(f, g)$ and $\mathscr{K}$ if for each $l \leqslant q$ and each pair $i, j \in K_l$, $i \neq j$, $(f(P))_i \neq (f(P))_j$.*

The next definition concerns the set of alternatives that can be made to win using a distinctive profile.

Definition 5.2.7. *For any $GS(f, g)$ compatible with $\mathscr{K}$, let $W_{\mathscr{K}}(f, g)$ be a subset of the alternatives defined as follows.*

$$W_{\mathscr{K}}(f, g) = \{GS(f, g)(P) : P \text{ is distinctive w.r.t. } GS(f, g) \text{ and } \mathscr{K}\}$$

For any profile P^M of manipulators and any alternative c, we define $T(m, n, \pi, c, P^M) = Pr(GS(f, g)(P^M \cup P^{NM}) = c)$. That is, given a profile of votes P^M of the manipulators, $T(m, n, \pi, c, P^M)$ is the probability that the winner of the profile $P^M \cup P^{NM}$ is

c, when the number of alternatives is m, the number of voters is n, and the nonmanipulators' votes P^{NM} are drawn i.i.d. according to π. Now we are ready to present the theorem.

Theorem 5.2.8. *Let $GS(f,g)$ be a generalized scoring rule that is compatible with $\mathscr{K}$. Let $\pi_{\mathscr{K}}$ be a distribution over $L(\mathcal{C})$ such that $\pi_{\mathscr{K}}$ is compatible with $\mathscr{K}$ w.r.t. f. For any $m > 0$, there exist constants $t_m > 0$ and $u_m > 0$ (neither of which depend on n) such that for any constant $h > 0$ (that does not depend on n) and any alternative $c \in W_{\mathscr{K}}(f,g)$, if the number of manipulators is at least hn^{α} ($\frac{1}{2} < \alpha < 1$) (as well as $o(n)$), then there exists a coalitional manipulation P^M such that*

$$T(m, n, \pi_{\mathscr{K}}, c, P^M) > 1 - t_m e^{-u_m n^{2\alpha-1}}$$

Theorem 5.2.8 states that when the number of alternatives is held fixed, if the number of manipulators is large ($\Omega(n^{\alpha})$ for $\alpha > \frac{1}{2}$, as well as $o(n)$) then for any alternative $c \in W_{\mathscr{K}}(f,g)$, there exists a manipulation P^M such that when the nonmanipulators' votes are drawn i.i.d. according to $\pi_{\mathscr{K}}$, then c is the winner with a probability of $1 - O(e^{-\Omega(n^{2\alpha-1})})$.

Proof of Theorem 5.2.8: Let $n' \geqslant hn^{\alpha}$. If $W_{\mathscr{K}}(f,g) = \varnothing$, then Theorem 5.2.8 vacuously holds. So we assume that $W_{\mathscr{K}}(f,g) \neq \varnothing$. For each $c \in W_{\mathscr{K}}(f,g)$, we associate c with a distinctive profile (w.r.t. f and $\mathscr{K}$), denoted by P_c^*, such that $c = GS(f,g)(P_c^*)$. We recall that P_c^* is distinctive if and only if for each $l \leqslant q$ and each pair $i, j \in K_l$, $i \neq j$, $(f(P_c^*))_i \neq (f(P_c^*))_j$. Let

$$d_{min} = \min_{l \leqslant q, i,j \in K_l, i \neq j, c \in W_{\mathscr{K}}(f,g)} (|(f(P_c^*))_i - (f(P_c^*))_j|)$$

That is, d_{min} is the minimal difference between any two components within the same element of $\mathscr{K}$ of $f(P_c^*)$, taken over all $c \in W_{\mathscr{K}}(f,g)$. Since $|W_{\mathscr{K}}(f,g)| < m$ (which does not depend on n), and P_c^* is distinctive, we know that $d_{min} > 0$ and does not depend on n. Let $p_{max} = \max_{c \in C} |P_c^*|$. That is, for all $c \in W_{\mathscr{K}}(f, g)$, the number of votes in P_c^* is no more than p_{max}. We note that p_{max} does not depend on n.

For any $c \in C$, define a profile of the manipulator votes P_c^M as follows. P_c^M consists of two parts:

1. $\lfloor \frac{n'}{|P_c^*|} \rfloor P_c^*$, and

2. an arbitrary profile for the remaining $n' - \lfloor \frac{n'}{|P_c^*|} \rfloor |P_c^*|$ votes.

That is, P_c^M consists mostly of $\lfloor \frac{n'}{|P_c^*|} \rfloor$ copies of P_c^*; the remaining votes (at most $|P_c^*|$) are chosen arbitrarily. We note that $|P_c^*|$ is a constant that does not depend on n, so that the second part becomes negligible when $n \to \infty$.

The next claim provides a lower bound on the difference between any two components of $f(P_c^M)$.

Claim 5.2.1. *There exists a constant d_c that does not depend on n such that the minimum difference between components of $f(P_c^M)$ is at least $d_c n^\alpha$.*

Proof of Claim 5.2.1: Since the minimal difference between any two components of P_c^* is at least d_{min}, the minimal difference between any two components of $f(P_c^M)$ is at least $\lfloor \frac{n'}{|P_c^*|} \rfloor d_{min}$. We note that the number of arbitrarily assigned votes in P_c^M is no more than $|P_c^*|$, and the difference between any two components in a vote is no more than v_{max}. Therefore the minimal difference between any two components of $f(P^M)$ is at least

$$\lfloor \frac{n'}{|P_c^*|} \rfloor d_{min} - v_{max} |P_c^*| \geqslant (\frac{n'}{p_{max}} - 1) d_{min} - v_{max} p_{max}$$

Note that this number is $\Omega(n^\alpha)$ because p_{max}, d_{min}, and v_{max} are constants that do not depend on n, and n' is $\Omega(n^\alpha)$. Therefore, there exists a d_c that does not depend on n such that the minimal difference between any two components of $f(P_c^M)$ is at least $d_c n^\alpha$.

(End of the proof for Claim 5.2.1.) $\square$

The next lemma is known as *Chernoff's inequality* (Chernoff, 1952).

Lemma 5.2.9 (Chernoff's inequality)**.** *Let $N \in \mathbb{N}$. Let $Y_1, \ldots, Y_N$ be N i.i.d. random variables with variance σ^2. Let $Y = \sum_{\zeta=1}^{N} Y_\zeta$. For any $0 \leqslant l \leqslant 2\sqrt{N}\sigma$, $Pr(|Y - E(Y)| \geqslant l\sqrt{N}\sigma) \leqslant 2e^{-l^2/4}$.*

For any profile P^{NM} for the nonmanipulators, any $i_1, i_2 \leqslant k$, $i_1 \neq i_2$, let $D(P^{NM}, i_1, i_2) = |(f(P^{NM}))_{i_1} - (f(P^{NM}))_{i_2}|$. The next claim states that if each vote of P^{NM} is drawn i.i.d. according to $\pi_{\mathscr{K}}$, then for any different i_1, i_2 within the same element K_l of the partition $\mathscr{K}$, the probability that the difference between the i_1th and the i_2th component of $f(P^{NM})$ is larger than $d_c n^\alpha$ is $O(e^{-\Omega(n^{2\alpha-1})})$.

Claim 5.2.2. *For any $l \leqslant q$ and any $i_1, i_2 \in K_l$ ($i_1 \neq i_2$), there exists a constant $d_{c,i_1,i_2} > 0$ that does not depend on n such that*

$$Pr(D(P^{NM}, i_1, i_2) > d_c n^\alpha) \leqslant 2e^{-d_{c,i_1,i_2} n^{2\alpha-1}}$$

Proof of Claim 5.2.2: Let $Y_1^{i_1,i_2}, \ldots, Y_{n^*}^{i_1,i_2}$ be n^* i.i.d. random variables such that the distribution for each $Y_\zeta^{i_1,i_2}$ is the same as the distribution for $(f(V))_{i_1} - (f(V))_{i_2}$, where V is drawn according to π. That is, for any $V \in L(\mathcal{C})$, with probability $\pi(V)$, $Y_1^{i_1,i_2}$ takes value $(f(V))_{i_1} - (f(V))_{i_2}$. Let $Y^{i_1,i_2} = \sum_{\zeta=1}^{|NM|} Y_\zeta^{i_1,i_2}$. Then, $Pr(D(P^{NM}, i_1, i_2) > d_c n^\alpha) = Pr(Y^{i_1,i_2} > d_c n^\alpha)$.

Since $\pi_{\mathscr{K}}$ is compatible with $\mathscr{K}$, for any $l \leqslant q$, $i_1, i_2 \in K_l$, we know that $E((f(V))_{i_1}) = E((f(V))_{i_2})$, where V is drawn according to π. Therefore, $E(Y_1^{i_1,i_2}) =$

0. Let $\sigma^2_{i_1,i_2}$ be the variance of $Y_1^{i_1,i_2}$. We note that σ_{i_1,i_2} does not depend on n. If $\sigma^2_{i_1,i_2} = 0$, then for any $V \in L(\mathcal{C})$, $(f(V))_{i_1} = (f(V))_{i_2}$ (because for any $V \in L(\mathcal{C})$, $\pi_{\mathscr{K}}(V) > 0$), which means that $W_{\mathscr{K}}(f,g) = \varnothing$. This contradicts the assumption that $W_{\mathscr{K}}(f,g) \neq \varnothing$. Hence $\sigma^2_{i_1,i_2} > 0$. Since $n' = o(n)$, $n^* = \Omega(n)$, and for sufficiently large n we have $\frac{d_c n^\alpha}{\sigma_{i_1,i_2}\sqrt{n^*}} \leqslant 2\sigma_{i_1,i_2}\sqrt{n^*}$. Therefore, we can use Lemma 5.2.9 (in which we let $N = n^*$) to bound $Pr(D(P^{NM}, i_1, i_2) > d_c n^\alpha)$ above as follows.

$$
\begin{aligned}
& Pr(D(P^{NM}, i_1, i_2) > d_c n^\alpha) \\
=& Pr(|Y^{i_1,i_2}| > d_c n^\alpha) \\
=& Pr(|Y^{i_1,i_2}| > \frac{d_c n^{\alpha-\frac{1}{2}}}{\sigma_{i_1,i_2}\sqrt{n^*}} \times \sigma_{i_1,i_2}\sqrt{n^*}) \\
\leqslant& 2e^{-(\frac{d_c n^{\alpha-\frac{1}{2}}}{\sigma_{i_1,i_2}\sqrt{n^*}})^2/4} && \text{(Lemma 5.2.9)} \\
\leqslant& 2e^{-\frac{d_c^2}{4\sigma^2_{i_1,i_2}} n^{2\alpha-1}} && (n^* \leqslant n)
\end{aligned}
$$

We note that $\frac{d_c^2}{4\sigma^2_{i_1,i_2}}$ is a constant that does not depend on n. Therefore, there exists $u_{c,i_1,i_2} > 0$ such that $Pr(D(P^{NM}, i_1, i_2) > d_c n^\alpha) \leqslant 2e^{-u_{c,i_1,i_2} n^{2\alpha-1}}$.

(End of the proof for Claim 5.2.2.) □

Let $u_c = \min_{l \leqslant q, i,j \in K_l, i \neq j} u_{c,i,j}$. Then $u_c > 0$ and is a constant (that does not depend on n). We note that for any P^{NM}, if $(P^{NM} \cup P_c^M) \not\to_{\mathscr{K}} P_c^M$, then there exists $l \leqslant q$, $i,j \in K_l$, $i \neq j$, such that $|(f(P^{NM}))_i - (f(P^{NM}))_j| > |(f(P_c^M))_i - (f(P_c^M))_j| > d_c n^\alpha$.

Therefore, we can bound the probability of $(P^{NM} \cup P_c^M) \sim_{\mathscr{K}} P_c^M$ below as follows.

$$
\begin{aligned}
& Pr((P^{NM} \cup P_c^M) \sim_{\mathscr{K}} P_c^M) \\
=& 1 - Pr((P^{NM} \cup P_c^M) \nsim_{\mathscr{K}} P_c^M) \\
\geqslant& 1 - Pr((\exists l \leqslant q)(\exists i, j \in K_l) D(P^{NM}, i, j) > d_c n^\alpha) \\
\geqslant& 1 - \sum_{l \leqslant q} \sum_{i,j \in K_l, i \neq j} Pr(D(P^{NM}, i, j) > d_c n^\alpha) \\
\geqslant& 1 - \sum_{l \leqslant q} \sum_{i,j \in K_l, i \neq j} 2e^{-u_{c,i,j} n^{2\alpha - 1}} \\
\geqslant& 1 - \sum_{l \leqslant q} \sum_{i,j \in K_l, i \neq j} 2e^{-u_c n^{2\alpha - 1}} \geqslant 1 - \frac{m(m-1)}{2} \times 2e^{-u_c n^{2\alpha - 1}}
\end{aligned}
$$

When n is sufficiently large, $P_c^M \sim_{\mathscr{K}} P_c^*$. Therefore, we know that there exists a constant $t_c > 0$ (that does not depend on n) such that $Pr((P^{NM} \cup P_c^M) \sim_{\mathscr{K}} P_c^*) \geqslant 1 - t_c e^{-u_c n^{2\alpha - 1}}$. Hence

$$
\begin{aligned}
& T(m, n, \pi_{\mathscr{K}}, c, P^M) \\
\geqslant& Pr((P^{NM} \cup P_c^M) \sim_{\mathscr{K}} P_c^*) \\
\geqslant& 1 - t_c e^{-u_c n^{2\alpha - 1}}
\end{aligned}
$$

(End of the proof for Theorem 5.2.8.) □

5.2.3 All-Powerful Manipulators in Common Rules

We already showed how Theorem 5.2.3, which states a condition under which manipulability is rare, can be applied to common voting rules in Corollary 5.2.4. We have not yet done so for Theorem 5.2.8, and we will do so in this section.[1] Specifically, we prove that if the number of alternatives is fixed, then for any positional scoring rule, Copeland, STV, ranked pairs, and maximin, if the number of manipulators is

[1] Except for Bucklin.

$\Omega(n^\alpha)$ ($\alpha > \frac{1}{2}$) and $o(n)$, and the nonmanipulators' votes are drawn i.i.d. according to the uniform distribution, then for any alternative c, there exists a coalitional manipulation that will make c win with a probability of $1 - O(e^{-\Omega(n^{2\alpha-1})})$.

The next theorem provides a necessary and sufficient condition for $W_{\mathscr{K}}(f,g)$ to be nonempty.

Theorem 5.2.10. *Let $G(f,g)$ be compatible with $\mathscr{K}$. $W_{\mathscr{K}}(f,g) \neq \emptyset$ if and only if for any $l \leqslant q$, any $i,j \in K_l$, $i \neq j$, there exists a vote $V \in L(\mathcal{C})$ such that $(f(V))_i \neq (f(V))_j$*

Proof of Theorem 5.2.10: First we prove the "if" part. Suppose that for any $l \leqslant q$, any $i,j \in K_l$, $i \neq j$, there exists a vote $V \in L(\mathcal{C})$ such that $(f(V))_i \neq (f(V))_j$. For any $l \leqslant q$, let

$$h_{l,max} = \max_{i,j \in K_l, V \in L(\mathcal{C})} \{|(f(V))_i - (f(V))_j|\},$$

$$h_{l,min} = \min_{i,j \in K_l, V \in L(\mathcal{C})} \{|(f(V))_i - (f(V))_j| : |(f(V))_i - (f(V))_j| > 0\}$$

That is, $h_{l,max}$ is the maximum difference between any two components within K_l, for any $f(V)$; $h_{l,min}$ is the minimum *positive* difference between any two components within K_l, for any $f(V)$. Then, for any $l \leqslant q$, $h_{l,max} \geqslant h_{l,min} > 0$. Let h be a natural number such that for any $l \leqslant q$, $h > \frac{h_{l,max}}{h_{l,min}} + 1$. Suppose $L(\mathcal{X}) = \{L_1, \ldots, L_{m!}\}$. Then, let $P = \sum_{s=1}^{m!} h^{m!-s} L_s$. We now show that P is distinctive w.r.t. $GS(f,g)$ and $\mathscr{K}$.

For any $l \leqslant q$, any $i,j \in K_l$, let t be the minimum natural number such that

$(f(L_t))_i \neq (f(L_t))_j$. W.l.o.g. let $(f(L_t))_i > (f(L_t))_j$. Then

$$
\begin{aligned}
&(f(P))_i - (f(P))_j \\
=&\sum_{s=1}^{m!} h^{m!-s}((f(L_s))_i - (f(L_s))_j) \\
=&h^{m!-t}((f(L_t))_i - (f(L_t))_j) + \sum_{s=t+1}^{m!} h^{-s}((f(L_s))_i - (f(L_s))_j) \\
\geqslant&h^{m!-t}h_{l,min} - \sum_{s=t+1}^{m!} h^{m!-s}h_{l,max} \\
=&h^{m!-t}(h_{l,min} - \frac{1}{h}\frac{1-\frac{1}{h^{m!-t}}}{1-\frac{1}{h}}h_{l,max}) \\
>&h^{m!-t}(h_{l,min} - \frac{1}{h-1}h_{l,max}) \\
>&0
\end{aligned}
$$

The last inequality holds because $h > \frac{h_{l,max}}{h_{l,min}} + 1$. Therefore, we know that for any $l \leqslant q$, any $i, j \in K_l$, $i \neq j$, $(f(P))_i \neq (f(P))_j$. Hence, P is distinctive w.r.t. $GS(f, g)$ and $\mathscr{K}$, completing the proof of the "if" part.

Now we prove the "only if" part. Suppose there exist $l \leqslant q$, $i, j \in K_l$ such that for any $V \in L(\mathcal{C})$, $(f(V))_i = (f(V))_j$. Then, for any profile P, $(f(P))_i = (f(P))_j$, which means that P is not distinctive w.r.t. $GS(f, g)$ and $\mathscr{K}$. Therefore $W_{\mathscr{K}}(f, g) = \emptyset$, completing the proof of the "only if" part.

(End of the proof for Theorem 5.2.10.) □

Now we show how the conditions in Theorem 5.2.8 are satisfied for any positional scoring rule, STV, Copeland, maximin, and ranked pairs, when the nonmanipulator votes are drawn from the uniform distribution.

Proposition 5.2.11. *Let π_u be the uniform distribution. For any rule r that is a positional scoring rule, Copeland, STV, maximin, or ranked pairs, let k_r, $GS(f_r, g_r)$*

and $\mathscr{K}_r$ be defined as in Proposition 5.1.5. Then, π_u is compatible with $\mathscr{K}_r$, and for any $l \leqslant q_r$ and any $i, j \leqslant K_l$ ($i \neq j$), there exists a vote $V \in L(\mathcal{C})$ such that $(f_r(V))_i \neq (f_r(V))_j$.

Proof of Proposition 5.2.11: We verify the condition in Theorem 5.2.10 for the common voting rules mentioned in the proposition by simple calculation, w.r.t. the GSR-formulation mentioned in the proof of Proposition 5.1.5.

positional scoring rule with scoring vector $\vec{v}$: for any $i \leqslant m$,

$$E_{V\sim\pi_u}((f_{\vec{v}}(V))_i) = \frac{\sum_{j=1}^m v(j)}{m}$$

Copeland, maximin, or ranked pairs: for any $i \leqslant m$, $j \leqslant m$, $i \neq j$,

$$E_{V\sim\pi_u}((f_r(V))_{(i,j)}) = \frac{1}{2}$$

STV: for any (S, j) such that $S \subset \mathcal{C}$, $|S| = i$, $c_j \notin S$,

$$E_{V\sim\pi_u}((f_{STV}(V))_{(S,j)}) = \frac{1}{m-i}$$

It left us to show, for each of these voting rules, and for any two given components (that lie within the same element of the partition), the vote that makes these two components different.

positional scoring rule with scoring vector $\vec{v}$: for any $i, j \leqslant m$, $i \neq j$, let V be the vote that ranks c_i at the top and c_j at the bottom; then, $(f_{\vec{v}}(V))_i = v(1) \neq v(m) = (f_{\vec{v}}(V))_j$.

Copeland, maximin, or ranked pairs: for any $i_1, i_2 \leqslant m$, $j_1, j_2 \leqslant m$, $i_1 \neq j_1$, $i_2 \neq j_2$, and $(i_1, j_1) \neq (i_2, j_2)$, let V be any vote in which $c_{i_1} \succ_V c_{j_1}$ and $c_{j_2} \succ_V c_{i_2}$. Because $(i_1, j_1) \neq (i_2, j_2)$, such a V exists. Then,

$$(f_r(V))_{(i_1,j_1)} = 1 \neq 0 = (f_r(V))_{(i_2,j_2)}$$

STV: for any $S \subset \mathcal{C}$, $j_1 \neq j_2$ such that $c_{j_1} \notin S$, $c_{j_2} \notin S$, let V be the vote in which c_{j_1} is at the top. Then $(f_{STV}(V))_{(S,j_1)} = 1 \neq 0 = (f_{STV}(V))_{(S,j_2)}$.

(End of the proof for Proposition 5.2.11.) □

By combining Proposition 5.2.11 and Theorem 5.2.10, we know that for any of the rules in Proposition 5.2.11, there exists a distinctive profile; hence, $W_{\mathscr{K}_r}(f, g)$ is nonempty (some alternative will win under the distinctive profile, without any tie). Also, all of these rules are neutral (they treat every alternative in the same way) when restricted to profiles that do not cause a tie, so if $W_{\mathscr{K}_r}(f, g)$ is nonempty, it must be that $W_{\mathscr{K}_r}(f, g) = \mathcal{C}$.

Corollary 5.2.12. *Let π_u be the uniform distribution over $L(\mathcal{C})$. For any rule r that is a positional scoring rule, Copeland, STV, maximin, or ranked pairs, if the number of manipulators is $\Omega(n^\alpha)$ ($\frac{1}{2} < \alpha \leqslant 1$) as well as $o(n)$, then for any $c \in \mathcal{C}$, there exists a coalitional manipulation P^M such that the probability that $r(P^M \cup P^{NM}) = c$ is $1 - O(e^{-\Omega(n^{2\alpha-1})})$.*

5.3 An Axiomatic Characterization for Generalized Scoring Rules

We have explicitly shown in the proof of Proposition 5.1.5 that a variety of common rules fall into the category of GSRs. However, we did not give any formal result about the generality of this class of rules. The apparent wide applicability of GSRs makes this class potentially interesting from the perspective of other problems in computational social choice. Indeed, some such uses are quite obvious. GSRs map every vote to a vector of scores (which are not necessarily associated with alternatives), and the outcome of the rule is based strictly on the sum of these vectors. As a result, the votes of a subset of the electorate can be summarized completely by the sum of their score vectors.[2] In fact, the definition of GSRs is even more restrictive: the final

[2] The problem of summarizing the votes of a subelectorate was introduced and studied (Chevaleyre et al., 2009; Xia and Conitzer, 2010a).

outcome only depends on direct comparisons among the components of the summed score vector. For example, the outcome may depend on a comparison between the first component and the third component of the summed vector; then, it does not matter (for this comparison) whether these components are 42 and 50, respectively, or 101 and 967, because in both cases component 1 is smaller. Because of this, the GSR framework is also useful for preference elicitation, specifically, for determining whether enough information has been elicited from the voters to declare the winner. In particular, if it becomes clear that the remaining (not yet elicited) information about the voters' preferences can no longer change any of the comparisons in scores, then we can terminate elicitation.

In Social Choice, *axiomatic* characterizations of voting rules are important because they give us deeper insight into rules, and can often be used to prove important results about rules. For GSRs, having an axiomatic characterization is especially important in order to know how the frequency-of-manipulability result for large number of manipulators (Theorem 5.2.8), which are negative results for the agenda of making manipulation computationally hard, might be circumvented. Axiomatic characterization of voting rules is a common topic in the social choice literature. For two alternatives, the majority rule has been characterized in May (1952). Young (1975) characterized positional scoring correspondences (that is, the voting correspondences that select all alternatives that have the highest total scores) by consistency, neutrality, and anonymity. Here we say that a voting correspondence r^c satisfies consistency, if for any pair of profiles P_1, P_2, if $r^c(P_1) \cap r^c(P_2) \neq \varnothing$, then $r^c(P_1 \cup P_2) = r^c(P_1) \cap r^c(P_2)$. When r' is a voting rule, that is, it always select a unique winner, this consistency coincides with the consistency defined in Section 2.2. In this section we will only consider voting rules.

In this section, we introduce a new axiomatic property for voting rules, which we call *finite local consistency*. A voting rule satisfies finite local consistency if the set of all profiles can be partitioned into finitely many parts, such that the voting rule is consistent within each part. The minimum number of parts for a rule is the *degree of consistency* for the rule. For example, a consistent rule has degree of consistency 1. We then characterize generalized scoring rules by anonymity, homogeneity, and finite local consistency, and show that the order of a GSR (that is, the dimension of the score vector) is related to the degree of consistency of the rule. It follows that *Dodgson's rule* is not a GSR, because it does not satisfy homogeneity (Brandt, 2009).

5.3.1 Finite Local Consistency

In this subsection, we formally define *finite local consistency.*

Definition 5.3.1. *Let S be a set of profiles. r is* locally consistent on S *if for any $P_1, P_2 \in S$ with $r(P_1) = r(P_2)$, we have $P_1 \cup P_2 \in S$ and $r(P_1 \cup P_2) = r(P_1) = r(P_2)$.*

Definition 5.3.2. *For any natural number t, a voting rule r is* t-consistent *if there exists a partition $\{S_1, \ldots, S_t\}$ of all profiles such that for all $i \leqslant t$, r is locally consistent within S_i. A voting rule r is* finitely locally consistent *if it is t-consistent for some natural number t.*

We emphasize that in this definition, a rule is defined for a fixed number m of alternatives, but for profiles of arbitrarily many voters. Later, we will show that some common rules are finitely locally consistent for *every* $m \in \mathbb{N}$; however, in those cases, t depends on m, which is allowed, as long as t is finite. We note that this finiteness condition is important: for *any* voting rule, there exists a partition that has infinitely many elements such that the voting rule is locally consistent, simply by letting each profile be an element by itself.

The *degree of consistency* of a voting rule r (for a particular m) is the smallest number of elements in a locally consistent partition of profiles. That is, the degree of consistency of r is t if r is t-consistent, and for any $t' < t$, r is not t'-consistent. (We note that the partition corresponding to this lowest t is not necessarily unique.) The degree of consistency can be seen as an approximation to consistency: the lower the degree of consistency of a voting rule, the more "consistent" it is, and 1-consistency is equivalent to the standard definition of consistency. We will be interested in the exact degree of consistency (rather than just whether it is finite or not), because, as we will show, this degree is related to the order of a GSR equivalent to the rule, which in turn is important for the summarization and elicitation problems that we mentioned in the introduction.

5.3.2 Finite local consistency characterizes generalized scoring rules

We now present our main result of this section. Let $\mathcal{P}(k)$ be the number of *total preorders* over k elements, that is, the total number of ways to rank k elements, allowing for ties.

Theorem 5.3.3. *r is a generalized scoring rule if and only if r is anonymous, homogenous, and finitely locally consistent. Moreover, for any t-consistent voting rule r, there exists a GSR of order $(\frac{t(t-1)m(m-1)}{4})m!+1$ that is equivalent to r; conversely, for any GSR $GS(f,g)$ of order k, there exists a $\mathcal{P}(k)$-consistent voting rule r that is equivalent to $GS(f,g)$.*[3]

Proof of Theorem 5.3.3: We prove the "if" part by a geometrical representation of a voting rule that is anonymous and homogenous, similarly to Young (1975). Let $L(\mathcal{C}) = \{l_1, \ldots, l_{m!}\}$ be the set of all linear orders over $\mathcal{C}$. Let r be an anonymous and homogenous voting rule, so that profiles can be represented as multisets of votes. Hence, there is a one-to-one correspondence between the set of all profiles and the set of all points in $\mathbb{N}^{m!}$: any profile $P = \sum_{x=1}^{m!} w_x l_x$, $w_x \in \mathbb{N}$ is associated with the point $\vec{p} = (w_1, \ldots, w_{m!})$, that is, $\vec{p} \in \mathbb{N}^{m!}$, and for any $j \leqslant m!$, the jth component

[3] The $\mathcal{P}(k)$ bound can be improved if more information about the structure of the GSR is taken into account. For the sake of simplicity, we omit further discussions of it.

of $\vec{p}$ is exactly the number of voters whose preferences are l_j in P. Therefore, r can also be seen as a mapping from $\mathbb{N}^{m!}$ to $\mathcal{C}$, defined as follows: for any $\vec{p} \in \mathbb{N}^{m!}$, $r(\vec{p}) = r(P)$, where P is the profile that $\vec{p}$ corresponds to. In the remainder of the proof, we will not distinguish between the point $\vec{p}$ and the profile P. Also, because r is homogenous, the domain of r can be extended to ${\mathbb{Q}_{\geqslant 0}}^{m!}$ (vectors of nonnegative rationales) in the following way. For any $\vec{p} \in {\mathbb{Q}_{\geqslant 0}}^{m!}$, let $h \in \mathbb{N}$ be such that $h\vec{p} \in \mathbb{N}^{m!}$; then, let $r(\vec{p}) = r(h\vec{p})$. (This is well defined because by homogeneity, the choice of h does not matter.)

Because r is t-consistent, there exists a partition $(S_1, \ldots, S_t)$ of $\mathbb{N}^{m!}$ such that r is locally consistent within each S_i. We note that $\vec{p} \in S_i$ implies $h\vec{p} \in S_i$ for each $h \in \mathbb{N}$, because each S_i must be closed under the union of vectors that produce the same result, and we can take the union of h vectors $\vec{p}$. Now, for any $i \leqslant t$, we define $S_i^{\mathbb{Q}} = \{q\vec{p} : q \in \mathbb{Q}_{\geqslant 0}, \vec{p} \in S_i\}$. It follows that ${\mathbb{Q}_{\geqslant 0}}^{m!} = \bigcup_{i=1}^{t} S_i^{\mathbb{Q}}$, and for any $i_1 \neq i_2$, $S_{i_1}^{\mathbb{Q}} \cap S_{i_2}^{\mathbb{Q}} = \{0\}$. For any $i \leqslant t$, any $j \leqslant m$, we define $S_i^j = S_i^{\mathbb{Q}} \cap r^{-1}(c_j)$. That is, S_i^j is the set of points (equivalently, profiles) in $S_i^{\mathbb{Q}}$ whose winner is c_j. It follows that for any $\vec{p}_1, \vec{p}_2 \in S_i^j \cap \mathbb{N}^{m!}$, we have $\vec{p}_1 + \vec{p}_2 \in S_i^j$; for any $\vec{p} \in S_i^j$, any $q \in \mathbb{Q}_{\geqslant 0}$, we must have $q\vec{p} \in S_i^j$. For any $S \subseteq {\mathbb{R}_{\geqslant 0}}^{m!}$, we say that S is $\mathbb{Q}$-*convex* if for any $\lambda \in \mathbb{Q} \cap [0, 1]$, any $\vec{p}_1, \vec{p}_2 \in S$, we have $\lambda\vec{p}_1 + (1 - \lambda)\vec{p}_2 \in S$. We say a $\mathbb{Q}$-convex set S is a $\mathbb{Q}$-*convex cone*, if for any $q \in \mathbb{Q}_{\geqslant 0}$, any $\vec{p} \in S$, we have $q\vec{p} \in S$.

Claim 5.3.1. *For any $i \leqslant t$, any $j \leqslant m$, S_i^j is a $\mathbb{Q}$-convex cone.*

Proof. For any $q_1, q_2 \in \mathbb{Q}_{\geqslant 0}$, any $\vec{p}_1, \vec{p}_2 \in S_i^j$, there exists $T \in \mathbb{N}$ such that $Tq_1\vec{p}_1, Tq_2\vec{p}_2 \in \mathbb{N}^{m!}$. We note that $Tq_1\vec{p}_1, Tq_2\vec{p}_2 \in S_i^j$, which implies that $Tq_1\vec{p}_1 + Tq_2\vec{p}_2$ is also in S_i^j. It follows that $q_1\vec{p}_1 + q_2\vec{p}_2 = \frac{1}{T}(Tq_1\vec{p}_1 + Tq_2\vec{p}_2) \in S_i^j$. □

For any $S \subseteq {\mathbb{R}_{\geqslant 0}}^{m!}$, we let $\text{conv}(S)$ be the *convex hull* of S in ${\mathbb{R}_{\geqslant 0}}^{m!}$. That is, $\text{conv}(S) = \{\sum_{i=1}^{h} \alpha_i\vec{p}_i : h = 1, 2, \ldots, \sum_{i=1}^{h} \alpha_i = 1, (\forall i \leqslant h)\ \alpha_i > 0, \alpha_i \in \mathbb{R}, \vec{p}_i \in S\}$.

Lemma 5.3.4 (proved in Young (1975)). *$S \subseteq \mathbb{Q}^{m!}$ is $\mathbb{Q}$-convex if and only if $S = conv(S) \cap \mathbb{Q}^{m!}$.*

Let $d \in \mathbb{N}$, $S_1, S_2 \subseteq \mathbb{R}^d$, and for any $x \in \mathbb{R}$, let $\delta(x) = 1$ if $x > 0$, $\delta(x) = -1$ if $x < 0$, and $\delta(0) = 0$. We say that S_1 and S_2 are *separated* by a finite set of vectors $I = \{\vec{p}_1, \ldots, \vec{p}_o\}$, in which $\vec{p}_i \in \mathbb{R}^l$ for all $i \leqslant o$, if there exist two sets $O_1, O_2 \subseteq \{-1, 0, 1\}^I$ such that $O_1 \cap O_2 = \{0\}$, and for any $\vec{p} \in S_1$ ($\vec{p} \neq 0$), we have $\delta(\vec{p}, I) = (\delta(\vec{p} \cdot \vec{p}_1), \ldots, \delta(\vec{p} \cdot \vec{p}_o)) \in O_1$; for any $\vec{p} \in S_2$ ($\vec{p} \neq 0$), we have $(\delta(\vec{p} \cdot \vec{p}_1), \ldots, \delta(\vec{p} \cdot \vec{p}_o)) \in O_2$. In this case we also say that I separates S_1 from S_2 *via* O_1, O_2.

$S \subseteq \mathbb{R}^d$ is called an *affine space* if for any $\vec{p}_1, \vec{p}_2 \in S$, any $q_1, q_2 \in \mathbb{R}$, we have $q_1\vec{p}_1 + q_2\vec{p}_2 \in S$. For any $S' \subseteq \mathbb{R}^d$, we let aff$(S')$ denote the *affine extension* of S' as follows: $\text{aff}(S') = \{\sum_{i=1}^{h} \alpha_i \vec{p}_i : h = 1, 2, \ldots, (\forall i \leqslant h)\ \alpha_i \in \mathbb{R}, \vec{p}_i \in S'\}$. That is, aff$(S')$ is the smallest affine space in $\mathbb{R}^d$ that contains S'. We let *relint*(conv(S)) denote the *relative interior* of conv(S), defined as follows. *relint*(conv(S)) is the set of all vectors $\vec{p} \in \mathbb{R}^d$ such that there exists $\epsilon > 0$ such that $B(\vec{p}, \epsilon) \cap \text{aff}(S) \subseteq \text{conv}(S)$, where $B(\vec{p}, \epsilon)$ is the ball centered on $\vec{p}$ with radius ϵ.

Lemma 5.3.5. *Let $S \subseteq \mathbb{R}^{m!}$ be an affine space, and let $S_1, S_2 \subseteq S \cap {\mathbb{Q}_{\geqslant 0}}^{m!}$ be two $\mathbb{Q}$-convex cones such that $S_1 \neq S_2$, $S_1 \cap S_2 = \{0\}$. There exists a finite set of vectors $I \subseteq \mathbb{R}^{m!}$ that separates S_1 from S_2, and $|I| \leqslant dim(S)$.*

Proof. We prove the claim by induction on $\dim(S)$. When $\dim(S) = 1$, it must be the case that one of S_1 and S_2 is $\{0\}$, and the other has an element $\vec{p}' \neq 0$. Without loss of generality, we let $S_1 = \{0\}$, $S_2 \neq \{0\}$. In this case, we let $I = \{\vec{p}'\}$, $O_1 = \{0\}$, and $O_2 = \{0, 1\}$.

Suppose Lemma 5.3.5 holds for $\dim(S) \leqslant d$. Without loss of generality, we assume $\dim(\text{aff}(S_1)) \geqslant \dim(\text{aff}(S_2))$. When $\dim(S) = d + 1$, there are two cases.

Case 1: $\dim(\text{aff}(S_1)) = \dim(\text{aff}(S_2)) = d + 1$. In this case $S = \text{aff}(S_1) = \text{aff}(S_2)$. First we prove that $\text{relint}(\text{conv}(S_1)) \cap \text{relint}(\text{conv}(S_2)) = \varnothing$. If not, suppose $\vec{p} \in \text{relint}(\text{conv}(S_1)) \cap \text{relint}(\text{conv}(S_2))$. Let $\vec{p} = \sum_{j=1}^{h} \alpha_j \vec{p}_j$, where $\sum_{j=1}^{h} \alpha_j = 1$, for all $j \leqslant h$, $\vec{p}_j \in S_1$ and $\alpha_j \geqslant 0$, and $B(\vec{p}, \epsilon) \cap S \subseteq \text{conv}(S_1)$, $B(\vec{p}, \epsilon) \cap S \subseteq \text{conv}(S_2)$. There exist $\beta_j \in \mathbb{Q}_{\geqslant 0}$ $(j \leqslant h)$ such that $\vec{p^*} = \sum_{j=1}^{h} \beta_j \vec{p}_j \neq 0$, and the distance between $\vec{p^*}$ and $\vec{p}$ is less than ϵ (by setting the β_j sufficiently close to the α_j). We note that S_1 is $\mathbb{Q}$-convex, which means that $\vec{p^*} \in S_1$. It follows that $\vec{p^*} \in \text{conv}(S_2)$, because $\vec{p^*} \in B(\vec{p}, \epsilon) \cap S$. From Lemma 5.3.4 we have that $S_2 = \text{conv}(S_2) \cap \mathbb{Q}_{\geqslant 0}{}^{m!}$. Therefore, $\vec{p^*} \in \text{conv}(S_2) \cap \mathbb{Q}_{\geqslant 0}{}^{m!} = S_2$. This contradicts the assumption that $S_1 \cap S_2 = \{0\}$.

Because $\text{relint}(\text{conv}(S_1)) \cap \text{relint}(\text{conv}(S_2)) = \varnothing$, we apply the separating hyperplane theorem: there exists a hyperplane $H_{\vec{p^*}}$ characterized by $\vec{p^*} \in \mathbb{R}^{m!}$, such that for any $\vec{p}_1 \in S_1$, $\vec{p}_1 \cdot \vec{p^*} \leqslant 0$; for any $\vec{p}_2 \in S_2$, $\vec{p}_2 \cdot \vec{p^*} \geqslant 0$; and at least one of S_1 and S_2 is not contained in $H_{\vec{p^*}}$. We let $S' = S \cap H_{\vec{p^*}}$, and $S_1' = S_1 \cap S'$, $S_2' = S_2 \cap S'$. $H_{\vec{p^*}}$ does not contain S, so it follows that $\dim(S') < \dim(S) = d + 1$. Applying Lemma 5.3.5 on S', S_1', S_2' (using the induction assumption), there exists a set of vectors I' that separates S_1' from S_2' via O_1', O_2', $|I'| \leqslant d$. Let $I = \{\vec{p^*}\} \cup I'$ and $O_1 = \{\vec{a} \in \{-1, 0, 1\}^I : \vec{a}|_{\{\vec{p^*}\}} = -1 \vee (\vec{a}|_{\{\vec{p^*}\}} = 0 \wedge \vec{a}|_{I'} \in O_1')\}$ (here, for $J \subseteq I$, let $\vec{a}|_J$ be the components of $\vec{a}$ corresponding to the vectors in J). This works because for any $\vec{p} \in S_1$, either $\vec{p}$ is in the open halfspace $\{\vec{p'} : \vec{p'} \cdot \vec{p^*} < 0\}$, or $\vec{p}$ is in $S_1 \cap H_{\vec{p^*}}$. Similarly, let $O_2 = \{\vec{a} \in \{-1, 0, 1\}^I : \vec{a}|_{\{p^*\}} = 1 \vee (\vec{a}|_{\{p^*\}} = 0 \wedge \vec{a}|_{I'} \in O_2')\}$. It follows that I separates S_1 from S_2 via O_1, O_2, and $|I| = |I'| + 1 \leqslant d + 1$.

Case 2: $\dim(\text{aff}(S_2)) < d + 1$. If $\text{aff}(S_1) = \text{aff}(S_2)$, then let $S' = \text{aff}(S_1)$, $|S'| < d + 1$. Applying Lemma 5.3.5 on S', S_1, S_2 (by the induction assumption), we can conclude that there exists $I' \subseteq \mathbb{Q}_{\geqslant 0}{}^{m!}$ that separates S_1 from S_2, and $|I'| \leqslant d < d+1$. If $\text{aff}(S_1) \neq \text{aff}(S_2)$, then there exists a hyperplane $H_{\vec{p^*}}$ (orthogonal to $\vec{p^*}$) such that $0 \in H_{\vec{p^*}}$, $S_2 \subseteq H_{\vec{p^*}}$, and $S_1 \nsubseteq H_{\vec{p^*}}$ (because the intersection of all hyperplanes that contains S_2 is S_2). Let $S' = \text{aff}(S_2)$, and $S'_1 = S_1 \cap S'$. S' is an affine space whose dimension is $\dim(\text{aff}(S_2)) < d+1$. For any $\vec{p}_1, \vec{p}_2 \in S'_1$, any $\lambda \in \mathbb{Q}_{\geqslant 0}$, we have that $\lambda\vec{p}_1 + (1-\lambda)\vec{p}_2 \in S_1$ (because S_1 is $\mathbb{Q}$-convex), and $\lambda\vec{p}_1 + (1-\lambda)\vec{p}_2 \in S'$ (because S' is an affine space); hence, $\lambda\vec{p}_1 + (1-\lambda)\vec{p}_2 \in S'_1$. Therefore, S'_1 is a $\mathbb{Q}$-convex cone.

By applying Lemma 5.3.5 on S', S'_1, S_2 (using the induction assumption), there exists $I' \subset {\mathbb{Q}_{\geqslant 0}}^{m!}$ ($|I'| \leqslant d$) that separates S'_1 from S_2 via O'_1, O'_2. We let $I = I' \cup \{\vec{p}^*\}$; $O_1 = \{\vec{a} \in \{-1,0,1\}^I : \vec{a}|_{\{\vec{p}^*\}} \neq 0 \vee (\vec{a}|_{\{\vec{p}^*\}} = 0 \wedge \vec{a}|_{I'} \in O'_1)\}$. This works because for any $\vec{p} \in S_1$, either $\vec{p} \cdot \vec{p}^* \neq 0$ (meaning that $\vec{p}$ is not in S'), or $\vec{p} \cdot \vec{p}^* = 0$, and $\delta(\vec{p}, I') \in O'_1$ (meaning that $\vec{p}$ is in $S_1 \cap S'$). Similarly we define $O_2 = \{\vec{a} \in \{-1,0,1\}^I : \vec{a}|_{\{\vec{p}^*\}} = 0 \wedge a|_{I'} \in O'_2\}$. It follows that I separates S_1 from S_2, and $|I| = |I'| + 1 \leqslant d+1$. This completes the proof of Lemma 5.3.5. □

For any $i_1, i_2 \leqslant t$, $j_1, j_2 \leqslant m$, where either $i_1 \neq i_2$ or $j_1 \neq j_2$, $S_{i_1}^{j_1} \cap S_{i_2}^{j_2} = \{0\}$. (We recall that S_i^j is the set of points in $S_i^{\mathbb{Q}}$ whose winner is c_j.) From Lemma 5.3.5, there exists a finite set $I_{i_1j_1,i_2j_2}$ of vectors that separates $S_{i_1}^{j_1}$ from $S_{i_2}^{j_2}$ via $O^1_{i_1j_1,i_2j_2}, O^2_{i_1j_1,i_2j_2}$, where $|I_{i_1j_1,i_2j_2}| \leqslant m!$. Now we can define a corresponding generalized scoring rule, as follows.

• $k = |\bigcup_{(i_1,j_1)\neq(i_2,j_2)} I_{i_1j_1,i_2j_2}| + 1$, and the components are indexed by vectors in some $I_{i_1j_1,i_2j_2}$, and a 0 component (which is always 0). Because $|I_{i_1j_1,i_2j_2}| \leqslant m!$, we have $k \leqslant (\frac{t(t-1)m(m-1)}{4})m! + 1$.

• For any $(i_1, j_1) \neq (i_2, j_2)$, any $\vec{p} = (p_1, \ldots, p_{m!}) \in I_{i_1j_1,i_2j_2}$, any $b \leqslant m!$, the $\vec{p}$ component of the generalized score vector given vote (ranking) l_b is $f(l_b) = p_b$. We note that for any profile $\vec{p} = (w_1, \ldots, w_{m!})$, any $\vec{p}^* = (p^*_1, \ldots, p^*_{m!}) \in I_{i_1j_1,i_2j_2}$, the $\vec{p}^*$ component of $f(\vec{p})$ is $\sum_{x=1}^{m!} w_x p^*_x = \vec{p} \cdot \vec{p}^*$.

• For any $\vec{a} \in {\mathbb{Q}_{\geqslant 0}}^k$ with $\vec{a} \neq 0$, $g(\vec{a}) = c_j$ if and only if there exists $i \leqslant t$ such that for any $i' \leqslant t$, $j' \leqslant m$, there exists $o \in O^1_{ij,i'j'}$ such that for any $\vec{p}^* \in I_{ij,i'j'}$, the following three conditions hold: (1) $\vec{a}|_{\vec{p}^*}$ is strictly larger than 0 (the value of the 0 component), if and only if $o|_{\vec{p}^*} = 1$; (2) $\vec{a}|_{\vec{p}^*}$ is equal to 0, if and only if $o|_{\vec{p}^*} = 0$; and (3) $\vec{a}|_{\vec{p}^*}$ is strictly smaller than 0, if and only if $o|_{\vec{p}^*} = -1$. That is, $g(\vec{a}) = c_j$ if and only if there exists $i \leqslant t$ such that for any i', j', we always have $\vec{a} \notin S_{i'}^{j'}$ by using the set of separation vectors $I_{ij,i'j'}$. (That g is well defined will follow from the following argument.)

Next, we prove that $GS(f,g) = r$. For any profile $\vec{p} \in \mathbb{Q}_{\geqslant 0}{}^{m!}$, suppose $\vec{p} \in S_i^j$. For any $(i,j) \neq (i',j')$, since $\vec{p} \in S_i^j$, by using the separation vectors $I_{ij,i'j'}$ and $O^1_{ij,i'j'}, O^2_{ij,i'j'}$, $\vec{p}$ should be classified as "not in $S_{i'}^{j'}$". That is, there exists $o \in O^1_{ij,i'j'}$ such that for any $\vec{p}^* \in I_{ij,i'j'}$, $o|_{\vec{p}^*} = \delta(\vec{p} \cdot \vec{p}^*)$; and for any $o' \in O^2_{ij,i'j'}$, there exists $\vec{p}^* \in I_{ij,i'j'}$ such that $o'|_{\vec{p}^*} \neq \delta(\vec{p} \cdot \vec{p}^*)$. It follows that $GS(f,g)(\vec{p}) = c_j$.

The "only if" part is straightforward. For any total preorder $\mathcal{O}$ over $\{1, \ldots, k\}$, we let $S_{\mathcal{O}} = \{\vec{p} \in \mathbb{Q}_{\geqslant 0}{}^{m!} : f(\vec{p}) \sim \mathcal{O}\}$. For any $\vec{p}_1, \vec{p}_2 \in S_{\mathcal{O}}$, $f(\vec{p}_1 + \vec{p}_2) = f(\vec{p}_1) + f(\vec{p}_2) \sim \mathcal{O}$, so that $GS(f,g)(\vec{p}_1) = GS(f,g)(\vec{p}_2) = GS(f,g)(\vec{p}_1 + \vec{p}_2)$. Hence, $GS(f,g)$ is locally consistent within $S_{\mathcal{O}}$. It follows that $\{S_{\mathcal{O}}\}$ is a finitely locally consistent partition for the rule, of size $\mathcal{P}(k)$. □

We are not aware of any closed-form formula for $\mathcal{P}(k)$, though there exist recursive formulas. We now give a simple upper bound on $\mathcal{P}(k)$. Any total preorder V can be represented by a strict order $(c_{i_1} \succ c_{i_2} \succ \ldots \succ c_{i_m})$ and a string $\vec{s} = (s_1, \ldots, s_{m-1}) \in \{0,1\}^{m-1}$, as follows: if $s_l = 0$ then $c_{i_l} \succ_V c_{i_{l+1}}$, and if $s_l = 1$ then $c_{i_l} \approx_V c_{i_{l+1}}$. This implies $\mathcal{P}(k) \leqslant k!2^{k-1}$.

5.4 A Scheduling Approach for Positional Scoring Rules

So far in this chapter we have been focusing on characterizing the frequency of manipulability for common voting rules, in order to show that computational complexity is not a strong barrier against manipulation. In this section, we argue that computational complexity is not a strong barrier against manipulation from the viewpoint of approximation. The optimization problem we will look at in this section asks for the smallest total number (weight) of the manipulators that can make a given alternative win. This optimization problem severs as our basis for approximation, and has two dimensions: the first dimension concerns whether the votes are weighted or unweighted, and the second dimension concerns whether the manipulators' votes are divisible (that is, each manipulator can cast a convex combination of linear orders as her vote) or not. For example, when the voters are unweighted and are not allowed cast divisible votes, the problem is the UCO problem (Definition 3.1.3).

Our main contribution is the exploration of a surprising and fruitful connection between coalitional manipulation for positional scoring rules and scheduling. We demonstrate that some of work on the latter problem can be leveraged to obtain nontrivial algorithmic results for the former problem.

The intuition behind the reduction is as follows. The scheduling problem to which we reduce is that of scheduling on parallel machines where the goal is to minimize makespan. In the coalitional manipulation problem for a positional scoring rule with scoring vector $\vec{s}_m$, each manipulator j always ranks the coalition's preferred alternative c first, but must award $\vec{s}_m(i) \cdot w_j$ points to the alternative it ranks ith, where w_j is the manipulator's weight. For any $i \geqslant 2$, we define a machine for $\vec{s}_m(i)$; the larger $\vec{s}_m(i)$ is in relation to $\vec{s}_m(1)$, the slower the machine is. Furthermore, each alternative besides c is a job; the larger the gap between the score of this alternative and the score of c, the larger the job is. When a manipulator with weight w_j ranks an alternative in the ith position, it decreases the gap between c and this alternative by $(\vec{s}_m(1) - \vec{s}_m(i))w_j$ points, which, under the detailed reduction, is equivalent to processing the corresponding job on the $(i-1)$th slowest machine for w_j time units.

In addition to WCM (Definition 3.1.1), UCM (Definition 3.1.2), and UCO (Definition 3.1.3). In this section we also study the following problem for positional scoring rules.

Definition 5.4.1. *The* Coalitional Optimization for divisible votes (COd) *problem is defined as follows. An instance is a tuple $(r, P^{NM}, \vec{w}^{NM}, c)$, where r is a voting rule, P^{NM} is the non-manipulators' profile, $\vec{w}^{NM}$ represents the weights of P^{NM}, and c is the alternative preferred by the manipulators. We are asked to find the minimum W^M such that there exist a divisible vote V^M for one manipulator with weight W^M, such that*

$$r((P^{NM}, \{V^M\}), (\vec{w}^{NM}, W^M)) = c$$

In the remainder of this section, we assume that c is ranked in the top position in the fixed-order tie-breaking mechanism. We let WCMd, UCMd, UCOd denote the variants of WCM, UCM, UCO, respectively, in which votes are divisible.[4] We note that it is irrelevant whether the votes of the non-manipulators are divisible or not; what matters is whether the manipulators' votes are divisible.

In Section 5.4.1, we consider WCMd, which may be interesting in its own right, but mainly serves to prepare the ground for our results regarding WCM. We give a polynomial-time algorithm for WCMd under any positional scoring rule by reducing it to the well-studied scheduling problem known as $Q|pmtn|C_{max}$ (in which preemptions are allowed). This algorithm also solves COd.

In Section 5.4.2 we deal with the indivisible case (WCM), and augment the WCMd algorithm with a rounding technique. Based on existing results from the scheduling literature, we can assume that the scheduling solutions use relatively few preemptive break points. We then show that in the coalitional manipulation problem, we need at most one additional voter per preemptive break point. We obtain the following theorem, which is a somewhat weaker but far more generally applicable version of the main result of Zuckerman et al. regarding Borda (Zuckerman et al.,

[4] We do not need to define similar variant for COd, because it is not hard to see that any solution to a COd instance where the votes are divisible can be converted in polynomial time to a solution to the same instance where the votes are indivisible.

2009, Theorem 3.4).

THEOREM 5.4.10. *Algorithm 2 runs in polynomial time and*

1. *if the algorithm returns* false, *then there is no successful manipulation (even for the WCMd version of the instance);*

2. *otherwise, the algorithm returns a successful manipulation for a modified set of manipulators, consisting of the original manipulators plus at most $m-2$ additional manipulators, each with weight at most $W/2$, where W is the maximum weight of the manipulators.*

Crucially, in most settings of interest (e.g., political elections), the number of alternatives m is small compared to the number of voters, or even the number of manipulators. Moreover, WCM is NP-complete under scoring rules such as Borda and Veto, even when there are only three alternatives (Conitzer et al., 2007). Therefore, in many important scenarios, $m-2$ additional manipulators constitute a very small fraction of the total number of manipulators, that is, the algorithm gives a good "approximation" to WCM.

A direct implication of Theorem 5.4.10 is that in the unweighted case (UCM) our approximation algorithm always finds a manipulation with at most $m-2$ additional manipulators, if there exists one for the given instance. Put another way, the algorithm approximates UCO to an additive term of $m-2$.

In Section 5.5, we establish an "integrality gap," in the following sense: the optimal solution to UCO can require $m-2$ more manipulators than the optimal solution to UCOd (Theorem 5.5.3). Moreover, we show that there is a family of instances of UCO such that any algorithm that is based on rounding an optimal solution for COd requires $m-2$ more votes than the optimal UCO solution (Theorem 5.5.4). These results suggest that the analysis of the guarantees provided by our technique is tight.

5.4.1 Algorithms for WCMd and COd

In this section we present algorithms for WCMd and COd. We devise a polynomial-time algorithm that solves WCMd by reducing it to the scheduling problem known as $Q|pmtn|C_{max}$. This algorithm also solves COd exactly. In the next subsection (Section 5.4.2), we augment the algorithm for WCMd with a rounding technique, and obtain an approximation algorithm for WCM as a result. While our solution for WCMd may be interesting in its own right, its main purpose is to provide intuitions and techniques that are subsequently leveraged for approximating WCM.

We will show how to reduce WCMd/COd to the scheduling problem of *parallel uniform machines with preemption*, categorized as $Q|pmtn|C_{max}$ (see, for example, Brucker (2007) for the meaning of the notation). In an instance of $Q|pmtn|C_{max}$, we are given $\bar{n}$ jobs $\mathscr{J} = \{J_1, \dots J_{\bar{n}}\}$ and $\bar{m}$ machines $\mathcal{M} = \{M_1, \dots, M_{\bar{m}}\}$; each job J_i has a workload $p_i \in \mathbb{R}_+$, and the processing speed of machine M_i is $s^i \in \mathbb{R}_+$, that is, it will finish s^i amount of work in one unit of time. A *preemption* is an interruption of the job that is being processed on one machine (the job may be resumed later, not necessarily on the same machine). Preemptions are allowed in $Q|pmtn|C_{max}$. We are asked for the minimum makespan, i.e., the minimum time to complete all jobs, and an optimal schedule.

We first draw a natural connection between WCMd/COd under positional scoring rules and $Q|pmtn|C_{max}$. After counting the non-manipulators' votes only, each alternative will have a total non-manipulator score. For any $i \leqslant m-1$, we let p_i denote the gap between the non-manipulator score of c_i and the non-manipulator score of c (which is positive if the former is larger; the case where the gap is negative is trivial). In particular, the p_i's can be seen as the workload of $m-1$ jobs. We note that, without loss of generality, the manipulators will always rank c in the top position. Therefore, a manipulator vote (of weight 1) in which c_j is ranked in the

ith position decreases the gap between c_j and c by $\vec{s}_m(1) - \vec{s}_m(i)$ points.

We consider a set of $m-1$ machines $M_1, \ldots, M_{m-1}$ whose speeds are $\vec{s}_m(1) - \vec{s}_m(2), \ldots, \vec{s}_m(1) - \vec{s}_m(m)$, respectively. A ranking (a vote) is equivalent to an allocation of the $m-1$ jobs to machines: an alternative ranked i positions below c corresponds to a job allocated to the ith slowest machine. We can now see that the minimum makespan of the scheduling problem is the minimum total weight of the manipulators required to make c a winner, that is, the optimal solution to COd. For WCMd, the goal is to compute the votes for $\sum_{i=1}^{k} w_i$ "amount" of manipulators (since the votes are divisible, a problem instance with k manipulators with weights $\vec{w}$ is equivalent to a problem instance with a single manipulator whose weight is $\sum_{i=1}^{k} w_i$), such that the final total score of c is at least the final total score of any other alternative. This is equivalent to computing a schedule that completes all jobs within time at most $\sum_{i=1}^{k} w_i$.

Formally, for a WCMd instance $(\vec{s}_m = (\vec{s}_m(1), \ldots, \vec{s}_m(m)), P^{NM}, w^{NM}, c, k, (w_1, \ldots, w_k))$, we construct an instance of $Q|pmtn|C_{max}$ with $m-1$ jobs and $m-1$ machines (that is, $\bar{m} = \bar{n} = m-1$) as follows. For any $i \leqslant m-1$, we let $s^i = \vec{s}_m(1) - \vec{s}_m(i+1)$, $p_i = \max\{\vec{s}_m(P^{NM}, w^{NM}, c_i) - \vec{s}_m(P^{NM}, w^{NM}, c), 0\}$. We do not distinguish between alternative c_i and job J_i. This reduction is illustrated in the following example.

Example 5.4.2. Let $m = 4$, $\mathcal{C} = \{c, c_1, c_2, c_3\}$. The positional scoring rule is Borda (which corresponds to the scoring vector $(3, 2, 1, 0)$). The non-manipulators are unweighted (that is, their weights are 1), and their profile is
$P^{NM} = (V_1^{NM}, V_2^{NM}, V_3^{NM}, V_4^{NM})$, defined as follows.

$$V_1^{NM} = [c_1 \succ c \succ c_2 \succ c_3], \qquad V_2^{NM} = [c_2 \succ c_1 \succ c \succ c_3]$$

$$V_3^{NM} = [c_3 \succ c_2 \succ c_1 \succ c], \qquad V_4^{NM} = [c_1 \succ c_2 \succ c_3 \succ c]$$

We have that $s(P^{NM}, c) = 3$, $s(P^{NM}, c_1) = 9$, $s(P^{NM}, c_2) = 8$, $s(P^{NM}, c_3) = 4$.

Therefore, we construct a $Q|pmtn|C_{max}$ instance in which there are 3 machines M_1, M_2, M_3 whose speeds are $s^1 = 1, s^2 = 2, s^3 = 3$, corresponding to the 2nd, 3rd, and 4th position in the votes, respectively, and 3 jobs J_1, J_2, J_3, whose workloads are $p_1 = 6 = (9-3), p_2 = 5 = (8-3), p_3 = 1 = (4-3)$, respectively. □

Let $W_0 = 0$, $W = \max_{j \leqslant k} w_j$, and for any $1 \leqslant i \leqslant k$, $W_i = \sum_{j=1}^{i} w_j$. A schedule is usually represented by a *Gantt chart*, as illustrated in Figure 5.1. (We note that Figure 5.1 is not the solution to Example 5.4.2.)

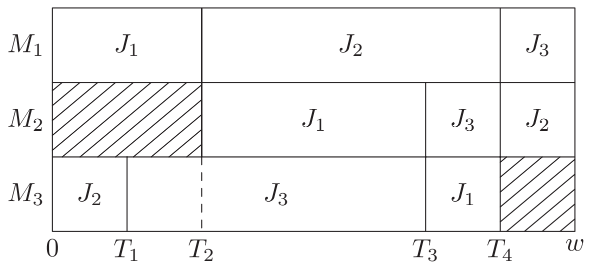

FIGURE 5.1: An example schedule. The machines are idle in shaded areas.

Let w be the minimum makespan for the $Q|pmtn|C_{max}$ instance constructed above, and let $f^* : \mathcal{M} \times [0, w] \to \mathscr{J} \cup \{I\}$ be an optimal solution to $Q|pmtn|C_{max}$, where I means that the machine is idle. If $w > W_k$, then there is no successful manipulation that makes c a winner. If $w \leqslant W_k$, we first extend the optimal solution f^* to make it fully occupy the whole time interval $[0, W_k]$; any way of allocating jobs to machines in the added time would suffice. Let f be the solution obtained in this way.

Given f, for any time $t \in [0, W_k]$, we say that t is a *preemptive break point* if there is a preemption at t—formally, there exists a machine M_i such that for some $\epsilon' > 0$, we have that for all $\epsilon \in [0, \epsilon']$, $f(M_i, t-\epsilon) \neq f(M_i, t+\epsilon)$, that is, the job being processed at time $t - \epsilon$ on M_i is different from the job being processed at time $t + \epsilon$. We let $B_f = \{T_1, \ldots, T_l\}$ denote the preemptive break points of f, where $0 < T_1 < T_2 < \ldots < T_l < W_k$. For example, the set of preemptive break points of the schedule in Figure 5.1 is $B_f = \{T_1, T_2, T_3, T_4\}$.

Example 5.4.3. The minimum makespan of the scheduling problem instance in Example 5.4.2 is $(6 + 5)/5 = 11/5$. An optimal schedule f is as follows.

$\mathbf{M_1}$: For any $0 \leqslant t \leqslant 11/5$, $f(M_1, t) = J_3$.

$\mathbf{M_2}$: For any $0 \leqslant t \leqslant 8/5$, $f(M_2, t) = J_2$; for any $8/5 < t \leqslant 11/5$, $f(M_2, t) = J_1$.

$\mathbf{M_3}$: For any $0 \leqslant t \leqslant 8/5$, $f(M_3, t) = J_1$; for any $8/5 < t \leqslant 11/5$, $f(M_3, t) = J_2$.

$t = 8/5$ is the only preemptive break point in this schedule. □

Any solution to the $Q|pmtn|C_{max}$ instance obtained from the reduction can be converted to a solution to WCMd in the following way. First, we assign jobs to all idle machines arbitrarily to ensure that at any time between 0 and W_k, no machines are idle and all jobs are allocated. Formally, we define $f' : \mathcal{M} \times [0, W_k] \to \mathscr{J}$ such that $\{f'(M_1, t), \ldots, f'(M_{m-1}, t)\} = \{J_1, \ldots, J_{m-1}\}$ for all t, and for any $M \in \mathcal{M}$ and $t \in [0, W_k]$, we have that if $f(M, t) \in \mathscr{J}$, then $f'(M, t) = f(M, t)$. For example, we can assign jobs to the shaded areas (which represent idle time) in the schedule in Figure 5.1 in the way illustrated in Figure 5.2.

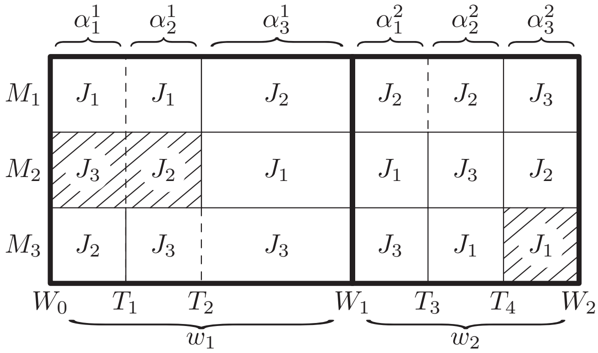

FIGURE 5.2: Conversion of an optimal schedule to a solution for WCMd.

Next, for any $1 \leqslant i \leqslant k$, we convert the schedule to the manipulators' votes in the natural way:

- If there are no preemption break points in (W_{i-1}, W_i), we let manipulator i vote for $c \succ f'(M_1, W_{i-1} + \epsilon) \succ f'(M_2, W_{i-1} + \epsilon) \succ \ldots \succ f'(M_{m-1}, W_{i-1} + \epsilon)$, where $\epsilon > 0$ is sufficiently small.

- If there are preemptive break points in (W_{i-1}, W_i), denoted by $T_a, T_{a+1}, \ldots, T_{a+b-1}$, then we let $V_1^i, \ldots, V_{b+1}^i$ denote the orders that correspond to the schedule at times $W_{i-1} + \epsilon, T_a + \epsilon, \ldots, T_{a+b-1} + \epsilon$, respectively. Let $\alpha_1^i = T_a - W_{i-1}$, $\alpha_2^i = T_{a+1} - T_a, \ldots, \alpha_{b+1}^i = W_i - T_{a+b-1}$. We let manipulator i vote for $\sum_{j=1}^{b+1}[\alpha_j^i/(W_i - W_{i-1})] \cdot V_j^i$.

Example 5.4.4. Suppose there are two manipulators whose weights w_1 and w_2 are illustrated in Figure 5.2. Manipulator 1 votes $[(1/4)(c \succ c_1 \succ c_3 \succ c_2) + (1/4)(c \succ c_1 \succ c_2 \succ c_3) + (1/2)(c \succ c_2 \succ c_1 \succ c_3)]$; manipulator 2 votes $[(1/3)(c \succ c_2 \succ c_1 \succ c_3) + (1/3)(c \succ c_2 \succ c_3 \succ c_1) + (1/3)(c \succ c_3 \succ c_2 \succ c_1)]$. □

On the basis of the exposition above we now refer the reader to Algorithm 1. The algorithm solves WCMd in three steps: 1. convert the WCMd instance to a $Q|pmtn|C_{max}$ instance; 2. apply a polynomial-time algorithm that solves $Q|pmtn|C_{max}$ (for example, the algorithm in Gonzalez and Sahni (1978)); 3. convert the solution to the scheduling instance to a solution to the WCMd instance. Algorithm 1 also solves COd, because the makespan w computed in Line 3 is the optimal solution to COd. It is easy to verify that the algorithm runs in polynomial time. To conclude, we have the following result.

Theorem 5.4.5. *Algorithm 1 solves WCMd and COd (exactly) in polynomial time.*

5.4.2 Algorithm for WCM

We now move on to the more difficult indivisible case. We first note that Algorithm 1 cannot be directly applied to WCM, because the manipulators' votes constructed in

Algorithm 1: compWCMd

1 $\forall i \leqslant m-1$, $s^i \leftarrow \vec{s}_m(1) - s_{i+1}$

2 $\forall i \leqslant m-1$, $p_i \leftarrow \max\{s(P^{NM}, w^{NM}, c_i) - s(P^{NM}, w^{NM}, c), 0\}$

3 Solve the $Q|pmtn|C_{max}$ instance (e.g., by the algorithm in Gonzalez and Sahni (1978)). Let w and f denote the minimum makespan and an extended optimal schedule; let $T_1, \ldots, T_l$ denote the preemptive break points.

4 **if** $w > W_k$ **then**

5   **return** *false.*

6 **end**

7 Let $f' : \mathcal{M} \times [0, W_k] \rightarrow \mathscr{J}$ be such that $\{f'(M_1, t), \ldots, f'(M_{m-1}, t)\} = \{J_1, \ldots, J_{m-1}\}$, and for any $M \in \mathcal{M}$, any $t \in [0, W_k]$, we have that if $f(M, t) \in \mathscr{J}$, then $f'(M, t) = f(M, t)$.

8 **for** $i = 1$ **to** k **do**

9   Let $V_1^i = [c \succ f'(M_1, W_{i-1} + \epsilon) \succ \ldots \succ f'(M_{m-1}, W_{i-1} + \epsilon)]$

10     $j \leftarrow 2$

11   **for** *each preemptive break point* $T \in (W_{i-1}, W_i)$ *(in order)* **do**

12   Let $V_j^i = [c \succ f'(M_1, T + \epsilon) \succ \ldots \succ f'(M_{m-1}, T + \epsilon)]$

13         $j \leftarrow j + 1$

14   **end**

15   For any j, let α_j^i be the length of the jth interval in $[W_{i-1}, W_i]$ induced by the preemptive break points.

16   Let manipulator i vote $\sum_j [\alpha_j^i / (W_i - W_{i-1})] \cdot V_j^i$, and add this vote to P^M

17 **end**

18 **return** P^M

Line 16 can be divisible. For any positional scoring rule, if there is a successful manipulation (in which all manipulators rank c in the top position), and we increase the weights of the manipulators, then c still wins the election. This property is known as *monotonicity in weights* (see Zuckerman et al. (2009) for a formal definition and the proof). Therefore, instead of having manipulator i cast the divisible vote $\sum_j [\alpha_j^i / (W_i - W_{i-1})] \cdot V_j^i$, we let her cast the indivisible vote V_{j*}^i, which is one of the V_j^i with the highest weight among all the V_j^i's constructed for manipulator i. In addition, for any $j \neq j^*$, we add one extra manipulator whose weight is α_j^i, and let the new manipulator vote V_j^i. It turns out that if we use a particular algorithm for the scheduling problem, then the solution will not require too many additional manipulators. This gives us Algorithm 2 for WCM.

Algorithm 2: compWCM

This algorithm is the same as Algorithm 1, except for the following two lines:

3 Use the algorithm in Gonzalez and Sahni (1978) to solve the scheduling problem

16 Let manipulator i vote for V^i_{j*}, where for any $j \neq j^*$, $\alpha^i_{j*} \geqslant \alpha^i_j$; and for any $j \neq j^*$, we add a new manipulator whose weight is α^i_j, and let her vote V^i_j

Example 5.4.6. Let the coalitional manipulation problem instance be the same as in Example 5.4.2. Suppose we have two manipulators whose weights are both 1; then, because the minimum makespan is $11/5 > 2$ (as observed in Example 5.4.3), there is no solution to the WCMd and WCM problem instances. The solution to the COd problem instance is 11/5.

Now suppose we have two manipulators, whose weights are $w_1 = 1$ and $w_2 = 6/5$, respectively. Let f be the optimal schedule defined in Example 5.4.3. A solution to the WCMd problem instance is obtained as follows. Manipulator 1 votes $[c \succ c_3 \succ c_2 \succ c_1]$, and manipulator 2 votes $[(1/2)(c \succ c_3 \succ c_2 \succ c_1) + (1/2)(c \succ c_3 \succ c_1 \succ c_2)]$. For WCM, the vote of manipulator 1 is the same, the vote of manipulator 2 is $[c \succ c_3 \succ c_2 \succ c_1]$, and there is one additional manipulator, whose weight is 3/5 and whose vote is $[c \succ c_3 \succ c_1 \succ c_2]$. □

Example 5.4.7. Suppose there are two manipulators whose weights are illustrated in Figure 5.2. The vote of manipulator 1 is $c \succ c_2 \succ c_1 \succ c_3$, and we introduce two new manipulators with weight $w_1/4$ whose votes are $c \succ c_1 \succ c_3 \succ c_2$ and $c \succ c_1 \succ c_2 \succ c_3$. The vote of manipulator 2 is $c \succ c_2 \succ c_1 \succ c_3$, and we introduce two new manipulators with weight $w_2/3$ whose votes are $c \succ c_2 \succ c_3 \succ c_1$ and $c \succ c_3 \succ c_2 \succ c_1$. Since $|B_f|$ (the number of preemptive break points) is 4, there are in total four additional manipulators. □

For any $j \neq j^*$, we must have $\alpha^i_j \leqslant (W_i - W_{i-1})/2 \leqslant W/2$ (recall that $W =$

$\max_{j \leqslant k} w_j$). Moreover, for any preemptive break point we introduce at most one extra manipulator. Therefore, we immediately have the following lemma that relates the number of the new manipulators to the number of preemptive break points.

Lemma 5.4.8. *If $w \geqslant W_k$, then there is no successful manipulation for WCMd (nor for WCM); otherwise, Algorithm 2 returns a manipulation with at most $|B_f|$ additional manipulators, each with weight at most $W/2$.*

Therefore, the smaller $|B_f|$ is, the fewer new manipulators are introduced by Algorithm 2. $|B_f|$ depends on which algorithm we use to solve $Q|pmtn|C_{max}$ in Line 3. In fact, there are many efficient algorithms that solve $Q|pmtn|C_{max}$. For example, $Q|pmtn|C_{max}$ can be solved in time $O(\bar{n}^2\bar{m})$ by a greedy algorithm (Brucker, 2007). At each time point t, the algorithm (called the *level algorithm*) assigns jobs to the machines in a way such that the greater the remaining workload of a job, the faster the machine it is assigned to.[5] However, this algorithm in some cases generates a schedule that has as many as $\bar{m}(\bar{m}-1)/2$ preemptive break points. Therefore, we turn to the algorithm by Gonzalez and Sahni (1978), which runs in time $O(\bar{n} + \bar{m} \log \bar{n})$ using at most $2(\bar{m} - 1)$ preemptions. Gonzalez and Sahni also showed that this bound is tight. We note that one preemptive break point corresponds to at least two preemptions, and in the instances that were used to show that the $2(\bar{m}-1)$ bound is tight, $\bar{m} - 1$ preemptive break points are required. Therefore, we immediately have the following lemma.

Lemma 5.4.9. *The number of preemptive break points in the solution obtained by the algorithm of Gonzalez and Sahni (1978) is at most $\bar{m} - 1$. Furthermore, this bound is tight.*

We note that $\bar{m} = m - 1$. Hence, combining Lemma 5.4.8 and Lemma 5.4.9, we

[5] The greedy algorithm of Zuckerman et al. (2009) is effectively a discrete-time version of the level algorithm.

have the following theorem, which is our main result.

Theorem 5.4.10. *Algorithm 2 runs in polynomial time and*

1. *if the algorithm returns* false*, then there is no successful manipulation (even for the WCMd version of the instance);*
2. *otherwise, the algorithm returns a successful manipulation for a modified set of manipulators, consisting of the original manipulators plus at most* $m-2$ *additional manipulators, each with weight at most* $W/2$.

5.5 Algorithms for UCM and UCO

We now consider the case where votes are unweighted. UCMd and UCOd can be solved using Algorithm 1. As for UCM/UCO, every manipulator's weight is one (so that $W = 1$), and we are only allowed to add new manipulators whose weight is also 1. We recall that increasing the weights of the manipulators never prevents c from winning. Therefore, in the context of UCM/UCO we use a slight modification of Algorithm 2, by adding one unweighted manipulator whenever Algorithm 2 proposes adding a weighted manipulator (whose weight can be at most 1/2).

Algorithm 3: compWCM

This algorithm is the same as Algorithm 1, except for the following two lines:

3 Use the algorithm in Gonzalez and Sahni (1978) to solve the scheduling problem.

16 Let manipulator i vote for V_1^i; for any $j > 1$, we add a new manipulator who votes for V_j^i.

The following corollary immediately follows from Theorem 5.4.10.

Corollary 5.5.1. *For UCM, if Algorithm 3 returns* false*, then there is no successful manipulation; otherwise, Algorithm 3 returns a successful manipulation with at most* $m-2$ *additional manipulators.*

Recall that Lines 1-3 of Algorithm 3 compute the minimum makespan w (the solution to COd) of the scheduling problem that is obtained from the UCM instance. It is easy to see that if votes are divisible then $\lceil w \rceil$ is the minimum number of unweighted manipulators required to make c win the election, that is, $\lceil w \rceil$ is the optimal solution to UCOd. Therefore, Algorithm 1 can easily be modified to yield an algorithm that solves UCOd. We further note that Algorithm 3 is an approximation algorithm for UCO, as the number of manipulators returned by Algorithm 3 is no more than $\lceil w \rceil + m - 2$. Put another way, Algorithm 3 returns a solution to UCO (with indivisible votes) that approximates the optimal solution to UCOd (with divisible votes) to an additive term of $m - 2$.

Generally, if there exists a successful manipulation, then Algorithm 3 returns a manipulation with additional manipulators. However, there are some special positional scoring voting rules under which UCM can always be solved exactly by Algorithm 1. Given $l \in \{1, \ldots, m-1\}$, the l-approval rule is the scoring rule where $\vec{s}_m(1) = \ldots = \vec{s}_m(l) = 1$ and $\vec{s}_m(l+1) = \ldots = \vec{s}_m(m) = 0$. For example, Plurality (with scoring vector $(1, 0, \ldots, 0)$) and Veto (with scoring vector $(1, \ldots, 1, 0)$) are 1-approval and $(m-1)$-approval, respectively. We note that UCM under any l-approval rule reduces to the scheduling problem in which all machines have the same speed. This corresponds exactly to the scheduling problem $P|pmtn|C_{max}$ in discrete time (that is, the preemptions are allowed only at integer time points), which has a polynomial-time algorithm: *Longest Remaining Processing Time first (LRPT)* Pinedo (2008). Therefore, if we modify Algorithm 3 by solving the reduced scheduling instance with LRPT, then we can solve UCM under any k-approval voting rule in polynomial time.[6] To summarize:

Corollary 5.5.2. *Let $l \in \{1, \ldots, m-1\}$. UCM/UCO for l-approval is in* P.

[6] The simple observation that UCM is in P for approval voting rules was also recently made by Andrew Lin (via personal communication), who employed a completely different (greedy) approach.

5.5.1 On The Tightness of The Results

We presently wish to argue that we have made the most of our technique. The next theorem states that the $m-2$ bound is tight in terms of the difference between the optimal solution to UCO and the optimal solution to UCOd under the same input. It also implies that Algorithm 3 is optimal in the sense that for any $q < m-2$, there is no approximation algorithm for UCO that always outputs a manipulation with at most q manipulators more than the optimal solution to UCOd. This result can be seen as a new type of integrality gap, which applies to our special flavor of rounding.

Theorem 5.5.3. *For any $m \geqslant 3$, there exists a UCO instance such that the (additive) gap between the optimal solution to UCOd and the optimal solution to UCO is $m-2$.*

Proof. For any $m \geqslant 3$, we let the scoring vector be $(m(m-1)(m-2)-1, \ldots, m(m-1)(m-2)-1, m(m-1)(m-2)-2, 0)$. Let $V = [c_1 \succ \ldots \succ c_{m-1} \succ c]$, and let π be the cyclic permutation on $\mathcal{C} \backslash \{c\}$, that is, $\pi : c_1 \to \ldots \to c_{m-1} \to c_1$. For any $i \leqslant m-1$, let V_i be the linear order over $\mathcal{C}$ in which c is ranked in position $(m-1)$, and $\pi^i(c_1) \succ_{V_i} \pi^i(c_2) \succ_{V_i} \ldots \succ_{V_i} \pi^i(c_{m-1})$. Let $P = (V, V_1, \ldots, V_{m-1})$, $P^{NM} = P \cup \pi(P) \cup \ldots \cup \pi^{m-2}(P)$. It follows that for any $i \leqslant m-1$, $s(P^{NM}, c_i) - s(P^{NM}, c) = (m-1)^2 - 1$. Let $V' = [c \succ c_1 \ldots \succ c_{m-1}]$; it can be verified that the divisible vote

$$\frac{1}{m-1}(V', \pi(V'), \pi^2(V'), \ldots, \pi^{m-2}(V'))$$

is sufficient to make c win, hence the optimal solution to UCOd is 1.

We next prove that the solution to UCO is $m-1$. Clearly the following profile is a successful manipulation.

$$(V', \pi(V'), \pi^2(V'), \ldots, \pi^{m-1}(V'))$$

Hence, it remains to show that the solution is at least $m-1$. For the sake of contradiction we assume that the solution is $m-2$, and P^M is the corresponding successful manipulation. Therefore, there must exist $i \leqslant m-1$ such that c_i is not ranked at the bottom of any of the votes of P^M. Therefore,

$$s(P^M, c) - s(P^M, c_i) \leqslant m-2 < (m-1)^2 - 1,$$

which means that $s(P^{NM} \cup P^M, c) - s(P^{NM} \cup P^M, c_i) < 0$. This contradicts the assumption that P^M is a successful manipulation. □

We next ask the following natural question: is it possible to improve the rounding technique so that the algorithm achieves a better bound, relative to the optimal solution for the indivisible case? This is not ruled out by Theorem 5.5.3, since that theorem compares to the optimal UCOd solution rather than the optimal UCO solution. Nevertheless, the answer is negative, as long as all linear orders in an optimal solution to the COd problem appear in the output of the algorithm. We say that an approximation algorithm $\mathcal{I}$ for UCO is *based on COd* if for any UCO instance, there exists an optimal solution to COd such that every linear order that appears in that solution also appears in the output of $\mathcal{I}$ (as a fraction of the vote of a manipulator).

Theorem 5.5.4. *Let $\mathcal{I}$ be an approximation algorithm based on COd. For any $m \geqslant 3$, there exists a UCO instance such that the gap between the optimal solution to UCO and the output of $\mathcal{I}$ is $m-2$.*

Proof. For any $m \geqslant 3$, we construct an instance such that the solution to the UCO problem is 1, but at least $m-1$ linear orders appear in any optimal solution to the COd problem (so the gap is $m-2$).

We let the scoring vector be $(m+2, 1, 0, \ldots, 0)$. Let

$$V = [c \succ c_1 \succ \ldots \succ c_{m-1}],$$

and

$$V' = [c_{m-1} \succ c_1 \succ c \succ c_2 \succ \ldots \succ c_{m-2}].$$

Furthermore, let

$$\pi : c_1 \to c_2 \to \ldots \to c_{m-1} \to c_1,$$

and

$$\pi^* : c \to c_1 \to \ldots \to c_{m-1} \to c.$$

We define preference profiles by letting

$$P = (V', V, \pi^*(V), (\pi^*)^2(V), \ldots, (\pi^*)^{m-2}(V))$$

$$\text{and } P^{NM} = P \cup \pi(P) \cup \ldots \cup \pi^{m-2}(P).$$

We have that $s(P, c) = m + 2$, $s(P, c_1) = m + 4$, and for any $2 \leqslant i \leqslant m - 1$, $s(P, c_i) = m + 3$. Therefore, $s(P^{NM}, c) = (m+2)(m-1)$ and for any $2 \leqslant i \leqslant m-1$, $s(P^{NM}, c_i) = (m + 3)(m - 1) + 1$. Therefore, for any $i \leqslant m - 1$, $s(P^{NM}, c_i) - s(P^{NM}, c) = m$. It follows that one manipulator suffices to make c the winner (by voting $c \succ c_1 \succ \ldots \succ c_{m-1}$).

On the other hand, the minimum weight for COd is $(m - 1)/m$, for example,

$$V^M = \frac{m-1}{m}\left(\frac{1}{m-1}V + \frac{1}{m-1}\pi(V) + \ldots + \frac{1}{m-1}\pi^{m-2}(V)\right).$$

In any manipulator's vote corresponding to the minimum total weight, every alternative except c must appear in the second position for a fraction of the vote. Therefore, any algorithm based on COd must output at least $m - 1$ linear orders. □

5.6 Summary

In this chapter, we extensively examined how strong computational complexity is as a barrier against manipulation. Most results in this chapter are negative. In Section 5.1 we showed that (roughly) for all generalized scoring rules, if the number of manipulators is $o(n^\alpha)$ for some $\alpha < 1/2$, then the probability that these manipulators can succeed goes to 0 as n goes to infinity; however, if the number of manipulators is

$\omega(n^\alpha)$ for some $\alpha > 1/2$, then the probability that these manipulators are all-powerful goes to 1 as n goes to infinity (except they cannot make alternatives against which the nonmanipulators are systematically biased win). We note that as n goes to infinity, $\sqrt{n}/n$ goes to zero.

This "dichotomy" result implies that when the total number of voters is large, even if the number of manipulators is very small compared to the number of nonmanipulators, the manipulators can still manipulate the winner with a high probability. We further gave an axiomatization in Section 5.3, which tells us how general the class of GSRs is—it is the class of all voting rules that satisfies anonymity, homogeneity, and finite local consistency.

Section 5.4 aimed at directly designing (approximation) algorithms for a number of coalitional manipulation problems for positional scoring rules. Built on top of a novel connection between coalitional manipulation problems and scheduling problems, we proposed polynomial-time algorithms that solve WCMd and COd. We also used these algorithms plus a rounding technique to obtain approximation algorithms for WCM, UCM, and UCO, with an additive error bound of $m-2$, which is tight in a sense.

Therefore, it seems that computational complexity is not a very strong barrier against manipulation. An obvious next step is to look for other ways to prevent manipulation. Note that one assumption made in all previous manipulation settings is that the manipulators have full information about the votes of the nonmanipulators. Therefore, a natural question to ask is: What if the manipulators do not have full information about the other voters' votes? The work in the next chapter is motivated by this question. We will study the case of one manipulator with limited information about other voters' votes. We will prove that restricting the information of the manipulator can effectively make a certain type of manipulation, which we call *dominating manipulation*, NP-hard. At one extreme, if the manipulator knows nothing, many voting rules are immune to dominating manipulations. These results seem very natural at a high level, but to obtain them, we need a formal model to analyze voters' strategic behavior.