

Near-Tight Algorithms for the Chamberlin-Courant and Thiele Voting Rules

Krzysztof Sornat, Virginia Vassilevska Williams and Yinzhan Xu

Abstract

We present an almost optimal algorithm for the classic Chamberlin-Courant multiwinner voting rule (CC) on single-peaked preference profiles. Given n voters and m candidates, it runs in almost linear time in the input size, improving the previous best $\mathcal{O}(nm^2)$ time algorithm of Betzler *et al.* (2013). We also study multiwinner voting rules on nearly single-peaked preference profiles in terms of the candidate-deletion operation. We show a polynomial-time algorithm for CC where a given candidate-deletion set D has logarithmic size. Actually, our algorithm runs in $2^{|D|} \cdot \text{poly}(n, m)$ time and the base of the power cannot be improved under the Strong Exponential Time Hypothesis. We also adapt these results to all non-constant Thiele rules which generalize CC with approval ballots.

1 Introduction

We study computational aspects of the Chamberlin-Courant voting rule (CC) [12] with general misrepresentation function and the Thiele rules [49] in which we choose a committee of size k such that the total utility of the voters is maximized. In the case of CC, the utility of a voter from a committee is determined by the most preferred committee member. In the case of a w -Thiele rule, parameterized by a non-increasing sequence $w = (w_1, w_2, \dots)$, the utility of a voter is equal to $w_1 + w_2 + \dots + w_x$, where x is the number of committee members approved by the voter.

CC and Thiele rules are basic multiwinner voting rules studied in social choice theory and, in particular, in the computational social choice community [10, 25]. They were discussed broadly for their applications, not only in parliamentary elections but also in many scenarios when a group of agents has to pick some number of items for joint use [46].

Unfortunately, computing an optimal committee under CC is already NP-hard for the case of *approval ballots*, i.e., when each voter gives a subset of approved candidates [42]. This special case is called *Approval Chamberlin-Courant* (Approval-CC) and it was considered by Procaccia *et al.* [42] as a minimization problem. Its maximization version is equivalent to the well-known MAX k -COVERAGE problem for which many hardness results have been shown [45]. In particular, MAX k -COVERAGE is NP-hard¹ to approximate within $(1 - 1/e + \varepsilon)$ factor for any $\varepsilon > 0$ [26]. Moreover, this hardness of approximation holds even in FPT-time with respect to solution size k assuming a gap version of the Exponential Time Hypothesis [35].

Notice that Approval-CC is equivalent to the $(1, 0, \dots)$ -Thiele rule, so all these hardness results for Approval-CC hold for this basic Thiele rule as well. One may consider other w -Thiele rules with specific w , e.g., w -Thiele rule with $w_i = 1/i$ which is equivalent to the well-known *Proportional Approval Voting* [34] and $w_i = 1/i^2$ which is related to *Penrose apportionment method* [10]. Notice that $(1, 1, \dots)$ -Thiele is equivalent to *Multiwinner Ap-*

¹NP-hardness of MAX k -COVERAGE directly implies NP-hardness of Approval-CC. This consequence was mentioned (without a proof) in the conference paper of Procaccia *et al.* [41].

proval Voting [4], which is solvable in polynomial-time. All the other ones² are NP-hard [46, Theorem 5].

Analogously to MAX k -COVERAGE, hardness of approximation results hold for the natural class of w sequences such that partial sums of w grows in a sublinear way, where a hardness constant depends strictly on w [5, 6, 19]. It follows that for such Thiele rules one should not expect a PTAS nor even an FPT-approximation scheme.

Because of the hardness of CC and Thiele rules, both rules were studied in restricted domains, i.e., on preference profiles with certain structures, e.g., single-peaked (SP) and single-crossing (SC). For a survey on structured preferences, see [22]. Such restricted domains are motivated by real-world elections. For instance, SP preference profiles appear when, intuitively, the candidates can be placed on a one-dimensional axis, and the farther away a candidate is from a voter’s favorite candidate the more the voter dislikes the candidate. Examples of such SP-axis are different policies in an ideological spectrum: the Left vs the Right, more liberal vs more conservative etc.

Fortunately, CC and Thiele rules on SP domain (CC-SP and Thiele-SP respectively) are known to be polynomial-time solvable [7, 38]. However, the dynamic programming algorithm for CC-SP proposed by Betzler *et al.* [7] is not optimal in terms of running time. We fill this gap by showing an almost optimal algorithm and we study computational complexity of CC and Thiele rules with nearly single-peaked preference profiles.

1.1 Our Contribution

The previous best running time for CC-SP was $\mathcal{O}(nm^2)$ [7]. We improve it by presenting an almost linear time algorithm (up to subpolynomial factors) in Theorem 7. To achieve this result we reduce our problem to the Minimum Weight t -Link Path problem and prove that our weight function has the *concave Monge property* (see Lemma 5). First, we construct an edge-weight oracle in $\mathcal{O}(nm \log(n))$ pre-processing time that gives a response in $\mathcal{O}(\log m)$ time. Then, for $k = \Omega(\log(m))$, using the algorithm of Schieber [44] we find a solution in $m \cdot 2^{\mathcal{O}(\sqrt{\log(k) \log \log(m)})} = m^{1+o(1)}$ time and oracle accesses. For the case $k = \mathcal{O}(\log(m))$ we use the $\mathcal{O}(mk)$ time algorithm of Aggarwal and Park [1] to achieve $\mathcal{O}(m \log(m))$ time and oracle accesses. Overall, the running time is $\mathcal{O}(nm \log(n) + m^{1+o(1)} \log(m) + m \log^2(m))$, which is almost linear in the input size.

We believe our algorithm will be efficient in practice as the $nm \log(n)$ part comes from sorting (hence, it is very efficient in practice) and the $m^{1+o(1)}$ part is a lower order term when $n = \Omega(m)$ (which is typical in many voting scenarios). All other components have running time $\mathcal{O}(nm)$ and are efficient. Moreover, one can also replace the $m^{1+o(1)}$ part with any other (asymptotically slower) implementation for Minimum Weight t -Link Path (such as the ones in [2]) to potentially obtain better practical performance for specific applications.

We then investigate the computational complexity of CC when the preference profile is not SP but close to SP in terms of the candidate-deletion operation. We assume a subset of d candidates is given, whose removal makes an instance SP. We call it the *candidate-deletion set*. We use this general problem formulation, as for some popular utility functions a smallest candidate-deletion set is easy to compute, but for some it is NP-hard. One of the standard motivations for considering nearly structured preferences is preference elicitation with small errors possible [22, Chapter 10.5]. Here, we present one more scenario when nearly SP preferences may appear: when a few new politicians come to the political scene and the voters are not sure how to place them on the left-right political spectrum. Clearly, the new politicians form a candidate-deletion set (assuming that the previously known politicians are placed on an SP-axis due to their longer public activity).

²We assume $w_1 = 1$ without loss of generality.

It is known that given the candidate-deletion set, CC can be solved in $\mathcal{O}^*(2^d)$ time [36], and thus is FPT with respect to d . Importantly, we show that the base of the power is optimal assuming the Strong Exponential Time Hypothesis. Notice that for $d = \mathcal{O}(\log(nm))$ our algorithm runs in polynomial-time. On the other hand, we show in Theorem 11 that if d is slightly larger (e.g., $\omega(\log n)$ or $\omega(\log m)$), then polynomial-time solvability of CC contradicts the Exponential Time Hypothesis (ETH). Overall, we derive polynomial-time algorithm for CC for almost all values of d for which such a polynomial-time algorithm may exist under ETH.

We adapt the above algorithm to Thiele rules by extending the Integer Linear Programming approach of Peters [38] to generalized Thiele rules (allowing different weight sequences for each voter) on SP preference profiles. This allows us to pre-elect some winning candidates by guessing the winners from the candidate-deletion set. As a result we obtain an $\mathcal{O}^*(2^d)$ -time algorithm. Unlike the case for CC, we were not able to prove that the base of the power cannot be improved. However, we show that there is no $\mathcal{O}^*(2^{o(d)})$ -time algorithm under ETH for each non-constant w -Thiele rule. Using this, we show that if d is allowed to be slightly larger than $\mathcal{O}(\log(nm))$ then polynomial-time solvability of any non-constant w -Thiele rule contradicts ETH.

1.2 Related Work

First, we present two papers that are the most relevant from a technical point of view.

The paper of Constantinescu and Elkind [14]. Recently, Constantinescu and Elkind [14] also used fast algorithms for Minimum Weight t -Link Path to solve CC, but on SC preference profiles. In their reduction, they construct an instance of Minimum Weight t -Link Path on an $\mathcal{O}(n)$ -node graph with the concave Monge property while the graph we construct has $\mathcal{O}(m)$ nodes. The best algorithm for solving Minimum Weight t -Link Path on an $\mathcal{O}(n)$ -node graph with the concave Monge property runs in $n^{1+o(1)}$ time [44], but Constantinescu and Elkind used an $\mathcal{O}(m)$ time algorithm to query the weight of each edge, so they end up getting an $mn^{1+o(1)}$ running time. In our algorithm, we first pre-process the preference profile in $\mathcal{O}(nm \log(n))$ time, and then an algorithm is able to query the weight of each edge in $\mathcal{O}(\log(m))$ time, so we get an $\mathcal{O}(nm \log(n) + m^{1+o(1)})$ overall running time. Even though both algorithms have near-linear running times, our algorithm is arguably faster since the $n^{o(1)}$ factor of their algorithm (and the $m^{o(1)}$ factor of our algorithm) could be super poly-logarithmic. Therefore, in the natural case where a preference profile is both SC and SP [23], it is potentially more efficient to use our algorithm.

The paper of Misra, Sonar, and Vaidyanathan [36]. Misra *et al.* [36] presented an $\mathcal{O}^*(2^d)$ time algorithm for CC-SP, whose main idea is to try all subsets of a given candidate-deletion set as winners and compute remaining winners. This construction is well-suited for generalization to Thiele rules, so we provide a proof similar to theirs for the CC-SP case for completeness. We note that Misra *et al.* also considered other research directions in their paper, e.g., voter-deletion distance to SP and egalitarian version of CC.

Next, we present related works that are less technically related to our paper but give a broader picture on the topic.

Other nearly SP measures. Different notions of nearly single-peaked preference profiles have been proposed [22, 24]. Let us mention just a few of them. *Voter-deletion distance* is the minimum number of voters one has to delete from the instance to achieve the SP property. *Swap distance* is the minimum number of swaps one has to make within the

votes (in a model with ordinal ballots) to obtain the SP property. Computing some of SP measures appears to be NP-hard, in particular, it is the case for both notions mentioned above [24]. Hence, some FPT and approximation algorithms have been proposed [20].

Generalizations of SP to other graphs. There are generalizations of SP to more complex graphs, e.g., a circle or a tree. For example, all Thiele rules are polynomial-time solvable for SP preference profiles on a line, and also on a circle [39]. Peters *et al.* [40] generalized the algorithm of Betzler *et al.* [7] to SP on a tree, but their running time is $\text{poly}(n, m^\lambda, k^\lambda)$, where λ is the number of leaves.

Clustering problems. CC is equivalent to a maximization version of classic clustering problem called k -MEDIAN (in discrete but not necessarily metric space) [11]. Therefore, CC may be seen as a clustering method in which we partition voters into k clusters, where cluster centers are defined by the winning committee members and the voters from a common cluster have the same most preferred committee member. Procaccia *et al.* [42] showed NP-hardness for the Monroe rule [37] which, in addition to CC, requires to have a balanced representation assignment. The Monroe rule is related to the CAPACITATED k -MEDIAN clustering problem (see e.g. the paper of Cohen-Addad and Li [13]) with all cluster center capacities equal to $\lceil \frac{n}{k} \rceil$ but in the case of the Monroe rule there are also lower-bounds on the capacities used: $\lfloor \frac{n}{k} \rfloor$.

2 Preliminaries

First we introduce some notations and define the computational problems formally. The \tilde{O} notation suppresses factors subpolynomial in the input size, i.e., factors $(nm)^{o(1)}$. The \mathcal{O}^* notation suppresses factors polynomial in the input size, i.e., factors $(nm)^{\mathcal{O}(1)}$.

Elections, misrepresentation, approval ballots. An *election* is a pair (V, C) consisting of a set V of n voters and a set C of m candidates.

A *misrepresentation function* $r : V \times C \rightarrow \mathbb{R}_{\geq 0}$ measures how much a voter is misrepresented by a particular candidate ($\mathbb{R}_{\geq 0}$ denotes a set of non-negative real numbers)³. Its dual measure is the *utility function* $u : V \times C \rightarrow \mathbb{R}_{\geq 0}$ that indicates how well a voter is represented by a particular candidate. A tuple (V, C, r) or (V, C, u) is called a *preference profile* or just a *profile* or *preferences*.

Balloting a misrepresentation function fully is costly for a voter (needs to provide m exact numbers). A simpler and one of the most popular type of a misrepresentation function is based on *approval ballots*. In *approval voting* a voter v gives an approval ballot as a vote, i.e., a subset A_v of candidates that v approves. In the case of approval voting we obtain an *approval utility function*, i.e., $u(v, c) = 1$ iff $c \in A_v$ and $u(v, c) = 0$ iff $c \notin A_v$. Furthermore, we define an *approval misrepresentation function* as $r(v, c) = 1 - u(v, c)$.

Another popular misrepresentation function is based on *Borda scores*. We call r a *Borda misrepresentation function* if for each voter v we have that $r(v, \cdot)$ is a bijection from C to $\{0, 1, \dots, m - 1\}$. It represents a linear ordering of candidates and counts how many candidates beat a particular candidate within the vote.

For a voter v , if $r(v, c)$ has different values for distinct candidates, then a vote of v can be seen as a *linear order* over candidates. If there exists two distinct candidates c, c' such that $r(v, c) = r(v, c')$ then we call a vote of v a *weak order*.

³We consider the real-RAM model of computation in this paper. Our algorithms can run perfectly on a typical word-RAM machine if the misrepresentations are given as integers.

Chamberlin-Courant voting rule. Originally, Chamberlin and Courant [12] defined a voting rule on the Borda misrepresentation function (currently, it is often called Borda-CC). In this paper, analogously to Betzler *et al.* [7], we study a more general computational problem. In the CHAMBERLIN-COURANT problem (CC) we are given an election (V, C) , a misrepresentation function r , a misrepresentation bound $R \in \mathbb{R}_{\geq 0}$ and a positive integer k . Our task is to find a size- k subset $W \subseteq C$ (called a *winning committee*) such that $\sum_{v \in V} \min_{c \in W} r(v, c) \leq R$ or return NO if such subset does not exist. Additionally, for $C' \subseteq C$, we overload the notation by writing $r(v, C') = \min_{c \in C'} r(v, c)$ and we define *total misrepresentation of C'* by $r(C') = \sum_{v \in V} r(v, C')$. In a minimization version of the CC problem we want to find a size- k committee W with the minimum total misrepresentation value $r(W)$.

w -Thiele voting rules. We define a computational problem w -THIELE parameterized by an infinite non-increasing non-negative sequence⁴ $w = (w_1, w_2, \dots)$, with $w_1 = 1$, as follows. We are given an election (V, C) , approval ballots A_v for each voter $v \in V$, an utility bound $U \in \mathbb{R}_{\geq 0}$, and a positive integer k . Our task is to find a size- k subset $W \subseteq C$ such that $\sum_{v \in V} \sum_{i=1}^{|A_v \cap W|} w_i \geq U$ or return NO if no such subset exists. Additionally, for $C' \subseteq C$, we overload the notation by writing $u(v, C') = \sum_{i=1}^{|A_v \cap C'|} w_i$ and we define *total utility of C'* by $u(C') = \sum_{v \in V} u(v, C')$. In a maximization version of w -THIELE we want to find a size- k committee W with the maximum total utility value. We notice that, a dual minimization version of w -THIELE is equivalent to the *OWA k -Median* problem with 0/1 connection costs [11].

Single-peaked preference profiles. Following Proposition 3 in [7] we say that a preference profile (V, C, r) is *single-peaked* (SP)⁵ if there exists a linear order \prec over C (called the *single-peaked-axis* or *SP-axis*) such that for every triple of distinct candidates $c_i, c_j, c_k \in C$ with $c_i \prec c_j \prec c_k$ or $c_k \prec c_j \prec c_i$ we have the following implication for each voter $v \in V$: $r(v, c_i) < r(v, c_j) \implies r(v, c_j) \leq r(v, c_k)$. Note that an SP profile with approval ballots can be seen as intervals of candidates (when ordering the candidates w.r.t. an SP-axis).

Candidate-deletion set. For a given preference profile (V, C, r) , a subset of candidates $D \subseteq C$ is called a *candidate-deletion set* if its removal from the instance makes the profile SP, i.e., $(V, C \setminus D, r_{-D})$ is an SP profile, where its misrepresentation function $r_{-D} : V \times (C \setminus D) \rightarrow \mathbb{R}_{\geq 0}$ is defined as r restricted to $V \times (C \setminus D)$. Typically we use $d = |D|$.

Approval-CC-SP. As we mentioned earlier, Approval-CC is equivalent to $(1, 0, \dots)$ -THIELE. If an SP-axis is given then the input of Approval-CC-SP can be given as two numbers per voter: both endpoints of the approval interval. Then the input size is $\Theta(n)$. It means that the running time of our algorithm from Theorem 7, which is $\tilde{O}(nm) = \tilde{O}(n^2)$, can be much larger than the input size. We would like to observe that Approval-CC-SP, i.e. CC with a preference restriction called *Candidate Interval* [22], can also be solved in almost optimal time because it is actually equivalent to finding the maximum k -cliques of an *interval graph*.

Observation 1. (*Application III in [44]*) *Approval-CC-SP can be solved in time $n^{1+o(1)}$ assuming that an SP-axis is given.*

⁴If not stated otherwise, we consider only such sequences throughout this paper and call them *Thiele sequences*.

⁵Sometimes called *possibly single-peaked* [22] as in this paper we allow ties (weak orders).

Parameterized complexity, ETH and SETH. We assume basic knowledge of parameterized complexity as this is common in computational social choice research [8, 18]. For a textbook on the topic see, e.g., [15]. To provide lower bounds we will use two popular conjectures on solving the satisfiability of propositional formulas in conjunctive normal form (CNF-SAT): Exponential Time Hypothesis (ETH) and its stronger version—Strong Exponential Time Hypothesis (SETH). For formal statements see, e.g., Conjectures 14.1 and 14.2 in [15], but here we state simple consequences of both conjectures as follows. 1) ETH implies that there is no $2^{o(N)}$ -time algorithm for 3-SAT with N variables and $\mathcal{O}(N)$ clauses, 2) SETH implies that there is no $(2 - \varepsilon)^N$ -time algorithm for CNF-SAT, with N variables and $\mathcal{O}(N)$ clauses.

3 Single-Peaked Preferences and Chamberlin-Courant Voting Rule

In this section we present our algorithm for CC-SP that is almost optimal under the assumption that an SP-axis is given in the input or at least one vote is a linear order. In the latter case we can find an SP-axis in $\mathcal{O}(nm)$ time, otherwise, if all votes are weak orders we do this in $\mathcal{O}(nm^2)$ time [27]. This is a bottle-neck for our algorithm. In such a case, we obtain the same $\mathcal{O}(nm^2)$ running time as the original dynamic programming approach of Betzler *et al.* [7].

We first label the candidates so that $c_1 \prec c_2 \prec \dots \prec c_m$ is the SP-axis. For simplicity of our analysis, we add an artificial candidate $c_0 \prec c_1$ so that $r(v, c_0) = U$ for any $v \in V$, where U is the maximum value of misrepresentation. Similarly, we add another artificial candidate $c_{m+1} \succ c_m$ so that $r(v, c_{m+1}) = U$ for any $v \in V$. After the additions, the profile is still single-peaked and the solution to the SP-CC instance does not change.

We first reduce CC-SP to the well-studied Minimum Weight t -Link Path problem which is defined as follows.

Definition 2 (Minimum Weight t -Link Path). *Given an edge-weighted directed acyclic graph (DAG), a source node and a target node, compute a min-weight path from the source to the target that uses exactly t edges.*

We create a graph on vertex set $\{0, \dots, m+1\}$. For every i, j where $0 \leq i < j \leq m+1$, we add an edge from i to j with weight $w(i, j) = r(\{c_i, c_j\}) - r(\{c_i\})$. We let the source vertex be vertex 0 and let the target vertex be vertex $m+1$. We set $t = k+1$. To show the correctness of this reduction, we use the following claim which is a key consequence of SP.

Claim 3. *For any $v \in V$, $0 \leq i < j \leq m+1$, and for any $C' \subseteq \{c_0, \dots, c_{i-1}\}$, it holds $r(v, \{c_i, c_j\}) - r(v, \{c_i\}) = r(v, C' \cup \{c_i, c_j\}) - r(v, C' \cup \{c_i\})$.*

Proof. Suppose c_i has the minimum misrepresentation of v among $C' \cup \{c_i\}$, then clearly $r(v, C' \cup \{c_i\}) = r(v, \{c_i\})$ and $r(v, C' \cup \{c_i, c_j\}) = r(v, \{c_i, c_j\})$ so the equality follows.

Otherwise, there exists $c' \in C'$ such that $r(v, c') < r(v, c_i)$. Since the profile is single-peaked and $c' \prec c_i \prec c_j$, we must have $r(v, c_i) \leq r(v, c_j)$. Therefore, $r(v, \{c_i\}) = r(v, \{c_i, c_j\})$ and $r(v, C' \cup \{c_i\}) = r(v, C' \cup \{c_i, c_j\})$, so the equality follows. \square

The following lemma shows the correctness of our reduction from CC-SP to Minimum Weight t -Linked Path.

Lemma 4. *Given the min-weight path from vertex 0 to vertex $m+1$ that uses $k+1$ edges, we can compute a winning committee $W \subseteq C$ of size k in linear time.*

Proof. Let P be any path from vertex 0 to vertex $m + 1$ that uses $k + 1$ edges, and let the vertices on P be $a_0, a_1, \dots, a_k, a_{k+1}$ for some $0 = a_0 < a_1 < \dots < a_k < a_{k+1} = m + 1$. The weight of path P is thus

$$\begin{aligned}
\sum_{i=0}^k w(a_i, a_{i+1}) &\stackrel{\text{def.}}{=} \sum_{v \in V} \sum_{i=0}^k r(v, \{c_{a_i}, c_{a_{i+1}}\}) - r(v, \{c_{a_i}\}) \\
&\stackrel{\text{Claim 3}}{=} \sum_{v \in V} \sum_{i=0}^k r(v, \{c_{a_j}\}_{0 \leq j \leq i+1}) - r(v, \{c_{a_j}\}_{0 \leq j \leq i}) \\
&= \sum_{v \in V} r(v, \{c_{a_j}\}_{0 \leq j \leq m+1}) - r(v, \{c_{a_j}\}_{0 \leq j \leq 0}) \\
&= -Un + \sum_{v \in V} r(v, \{c_{a_1}, \dots, c_{a_k}\}),
\end{aligned}$$

which is $-Un$ plus the total misrepresentation of the committee $\{c_{a_1}, \dots, c_{a_k}\} \subseteq C$.

Conversely, for any subset of C of size k with total misrepresentation R , we can find a path from vertex 0 to vertex $m + 1$ using $k + 1$ edges with weight $-Un + R$ in the graph by following the equations reversely. \square

By Lemma 4, it suffices to compute the minimum weight t -link path of the auxiliary graph. To do so efficiently, we use the fact that the edge weights have a certain structure.

Lemma 5. *The edge weights of the Minimum Weight t -Link Path instance satisfy the concave Monge property, i.e., for all $1 \leq i+1 < j \leq m$, it holds that $w(i, j) + w(i+1, j+1) \leq w(i, j+1) + w(i+1, j)$.*

We defer the proof of Lemma 5 to Appendix A.

By Lemma 5, we can use the $m^{1+o(1)}$ time algorithm for the Minimum Weight t -Link Path instance (we use the algorithm by Schieber [44] for the case where $k = \Omega(\log m)$ and use the algorithm by Aggarwal and Park [1] for the case where $k = \mathcal{O}(\log(m))$). Note that by a brute-force algorithm, we can compute each edge weight in $\mathcal{O}(n)$ time, so the overall running time becomes $nm^{1+o(1)}$. The subpolynomial factor in this running time is $2^{\mathcal{O}(\sqrt{\log(k) \log \log(m)})}$ when $k = \Omega(\log(m))$, which can be quite large. The following lemma gives a way to compute the edge weights faster, and consequently improves the subpolynomial factor.

Lemma 6. *After an $\mathcal{O}(m \text{IntSort}(n, nm))$ time pre-processing where $\text{IntSort}(n, nm)$ is the time for sorting n non-negative integers bounded by $\mathcal{O}(nm)$, we can create a data structure that supports edge weight queries in $\mathcal{O}(\log(m))$ time per query.*

We defer the proof of Lemma 6 to Appendix B.

In the real-RAM model of computation we consider in this paper, we can use any standard sorting algorithm to sort n integers in $\mathcal{O}(n \log(n))$ time, so the pre-processing time of Lemma 6 becomes $\mathcal{O}(nm \log(n))$. In the word-RAM model of computation, faster algorithms for sorting integers are known. For instance, it is known that $\text{IntSort}(n, nm) = \mathcal{O}(n \sqrt{\log \log(n)})$ [29]. Also, in a common case where $m = \text{poly}(n)$, $\text{IntSort}(n, nm) = \mathcal{O}(n)$ by radix sort. One may use such faster integer sorting algorithms to obtain faster running time for the overall algorithm in the word-RAM model of computation, when all misrepresentations are given as integers (e.g. when we use Borda misrepresentation function).

Combining Lemma 6 with previous ideas we obtain the following theorem.

Theorem 7. *We can solve CHAMBERLIN-COURANT on single-peaked preference profiles in time $\mathcal{O}(nm \log(n) + m^{1+o(1)})$ assuming that an SP-axis is given or at least one vote is a linear order.*

4 Nearly Single-Peaked Preferences and Chamberlin-Courant Voting Rule

Betzler *et al.* [7] showed that Approval-CC is $W[2]$ -hard w.r.t. k . Their reduction is from the HITTING SET problem (HS), in which we are given a family $\mathcal{F} = \{F_1, \dots, F_n\}$ of sets over a universe $U = \{u_1, \dots, u_m\}$ and a positive integer k , and the task is to decide whether there exists a size- k subset $U' \subseteq U$ such that $U' \cap F_i \neq \emptyset$ for every $F_i \in \mathcal{F}$.

Actually, the reduction presented by Betzler *et al.* [7] also works for the Monroe rule. This makes the reduction more complex than it is needed for Approval-CC. A simple reduction from HS to Approval-CC has a strict correspondence between a universe and candidates, and between a family of sets and voters. A voter approves a candidate if the corresponding set to the voter contains the corresponding element to the candidate. The misrepresentation bound is equal to 0 as this will require to cover (hit) all the voters (sets). Because of this correspondence, Approval-CC with $R = 0$ is equivalent to HS.

It was observed recently [28] that, assuming *Set Cover Conjecture* (SCC) there is no $\mathcal{O}^*((2-\varepsilon)^n)$ -time algorithm for Approval-CC or Borda-CC, for every $\varepsilon > 0$. This observation follows directly from the reductions presented by Betzler *et al.* [7] and SCC-hardness of HS [15, Conjecture 14.36]. On the other hand, for general misrepresentation function, Betzler *et al.* [7] showed an $\mathcal{O}^*(n^n)$ -time algorithm. An interesting open question is to close this gap for general misrepresentation functions. Note that for the case of Borda-CC, Gupta *et al.* [28] showed an $\mathcal{O}^*(2^n)$ -time algorithm.

Below, we observe the SETH-hardness of CC, due to the SETH-hardness of HS [16] and the reduction of Betzler *et al.* [7].

Observation 8. *Assuming the Strong Exponential Time Hypothesis, for every $\varepsilon > 0$, there is no $\mathcal{O}^*((2-\varepsilon)^m)$ -time algorithm for CC, even for approval ballots.*

It means that the $\mathcal{O}^*(2^m)$ -time brute-force algorithm, which checks all size- k subsets of candidates, is essentially optimal for CC in terms of parameter m . On the other hand, for CC-SP we showed an almost optimal algorithm (Theorem 7). Hence, it is natural to study computational complexity of CC on preferences that are close to SP in terms of the candidate-deletion operation.

We assume that a candidate-deletion set is given as this makes the problem formulation as general as possible. Otherwise, finding the minimum size candidate-deletion set would be a bottleneck for our algorithm as, in general, the problem is NP-hard. The hardness follows from the observation that the problem with approval ballots is equivalent to the problem of deleting a minimum number of columns to transform a given binary matrix into a matrix with the *consecutive ones property* (Min-COS-C). For an overview on the computational complexity of Min-COS-C see, e.g., [17]. An interesting hardness result for Min-COS-C described therein is that an α -approximation algorithm for Min-COS-C derives an $\alpha/2$ -approximate solution to the *Vertex Cover* problem. As a consequence, it is NP-hard to approximate the smallest candidate-deletion set within a factor of 2.72 [33] and it is Unique Games Conjecture-hard to approximate within a factor of $4 - \epsilon$, for every $\epsilon > 0$ [32]. The hardness holds already for approval preferences, where each voter approves at most 2 candidates. On the other hand, we can compute the minimum size candidate-deletion set in polynomial-time if all votes are linear orders [24, 43]. This holds, e.g., when the misrepresentation function is derived from a *committee scoring rule* [21] where a classic example is Borda scoring rule used in Borda-CC.

In the following theorem we show that CC is FPT w.r.t. the size of a given candidate-deletion set⁶ (denoted by D) and the obtained algorithm is essentially optimal assuming

⁶This result was presented before by Misra *et al.* [36], however our construction is well-suited for generalization to Thiele rules presented in Section 5, so we provide a proof for completeness.

SETH. The main idea of the algorithm is to: 1) guess pre-elected winners among D ; 2) delete D from the instance together with appropriate modification of the instance with respect to the pre-elected winners; 3) as the modified instance appears to be a CC-SP instance, we use our algorithm from Theorem 7.

Theorem 9. *We can solve CHAMBERLIN-COURANT with a given candidate-deletion set of size d in time $\mathcal{O}^*(2^d)$. Furthermore, assuming the Strong Exponential Time Hypothesis, for any $\varepsilon > 0$ there is no $\mathcal{O}^*((2-\varepsilon)^d)$ -time algorithm that solves the problem, even for approval ballots.*

We defer the proof of Theorem 9 to Appendix C. A simple corollary from Theorem 9 is that CC is polynomial-time solvable if d is logarithmic in the input size.

Corollary 10. *CHAMBERLIN-COURANT with a given candidate-deletion set of size d , where $d \leq \mathcal{O}(\log(nm))$, is polynomial-time solvable.*

One would ask whether we can have polynomial-time algorithm for larger values of d . Unfortunately, in the following theorem we show that if d is slightly larger than logarithm in the input size, say $d = \mathcal{O}(\log n \log \log m)$, then it is not possible under ETH.

Theorem 11. *Under the Exponential Time Hypothesis, there is no polynomial-time algorithm for CHAMBERLIN-COURANT with a given candidate-deletion set of size at most $f(n, m)$ for any function $f(n, m) = \omega(\log(n))$ or $f(n, m) = \omega(\log(m))$.*

Proof. By the known ETH hardness of HS [30] and the above-mentioned reduction from HS to Approval-CC, Approval-CC does not have $2^{o(N)}$ time algorithm when there are $\mathcal{O}(N)$ candidates and $\mathcal{O}(N)$ voters under ETH. From such hard instances, we construct hard instances for every function $f(n, m) = \omega(\log(n))$ or $f(n, m) = \omega(\log(m))$.

For the first case, given an Approval-CC instance with $\mathcal{O}(N)$ candidates and $\mathcal{O}(N)$ voters, we aim to create an instance where the size of the candidate-deletion set d is $\mathcal{O}(h(n, m) \log(n))$ for $h(n, m) = f(n, m) / \log(n) = \omega(1)$. We then add some dummy voters, one by one, until $n \geq 2^{N/h(n, m)}$ so that for each dummy voter v , $r(v, c) = 0$ for every candidate c (r is still an approval misrepresentation function). The minimum total misrepresentation will not change with the addition of these voters. The candidate-deletion set could be the set of all candidates. In the new instance $n \geq 2^{N/h(n, m)}$ and $d = m = \mathcal{O}(N)$. Thus, $d = \mathcal{O}(h(n, m) \log(n))$ as desired. Since the reduction clearly has running time $2^{o(N)}$, the new instance does not have $\text{poly}(n) = \text{poly}(2^{N/h(n, m)}) = 2^{o(N)}$ time algorithm under ETH.

The other case is analogous. Given an Approval-CC instance with $\mathcal{O}(N)$ candidates and $\mathcal{O}(N)$ voters, we aim to create an instance where $d = \mathcal{O}(h(n, m) \log(m))$ for $h(n, m) = f(n, m) / \log(m) = \omega(1)$. We then add some dummy candidates until $m \geq 2^{N/h(n, m)}$ so that for each dummy candidate c , $r(v, c) = 1$ for every voter v (r is still an approval misrepresentation function). The minimum total misrepresentation will not change with the addition of these candidates. Also, note that if we delete the original $\mathcal{O}(N)$ candidates, the profile will be single-peaked. Thus, $d \leq \mathcal{O}(N) = \mathcal{O}(h(n, m) \log m)$ as desired. Under ETH, the new instance does not have $\text{poly}(m) = \text{poly}(2^{N/h(n, m)}) = 2^{o(N)}$ time algorithm. \square

5 Nearly Single-Peaked Preferences and Thiele Voting Rules

In this section, we first provide an ETH-based lower bound on the running times of algorithms that solve any non-constant w -THIELE.

The first proof of NP-hardness of all non-constant w -THIELE follows from NP-hardness proof for the *OWA-Winner* problem for a specific family of *OWA vectors* [46, Theorem

5]. To show ETH-hardness straightforwardly, we will instead use a reduction from the INDEPENDENT SET problem (IS). In IS we are given a graph, a positive integer k and we are asked whether there exists a size- k subset of vertices such that no two of them are adjacent. The idea to reduce from IS was used in [4] for the above-mentioned *Proportional Approval Voting*. Actually, their reduction works for any w -THIELE with $w_2 < 1$. The idea was extended by Jain *et al.* [31] to other non-constant w -Thiele rules when $w_2 = 1$ by introducing a proper number of dummy candidates. They presented their result for a more general problem and, actually, they did not include a formal construction in their paper, so in the proof of Theorem 12 we include a simplified proof that works specifically for all non-constant w -THIELE.

In our reduction, we have $m = |V(G)| + \mathcal{O}(1)$ and $n = \frac{3}{2}|V(G)|$, where $V(G)$ is the set of vertices in an IS instance on a 3-regular graph. Therefore, since IS on 3-regular graphs does not have $2^{\mathcal{O}(|V(G)|)}$ time algorithms under ETH [3, Theorem 5], we obtain the following theorem.

Theorem 12. *Assuming the Exponential Time Hypothesis, there is neither $\mathcal{O}^*(2^{\mathcal{O}(m)})$ nor $\mathcal{O}^*(2^{\mathcal{O}(n)})$ -time algorithm for any non-constant w -THIELE.*

We defer the proof of Theorem 12 to Appendix D.

Theorem 12 means that the $\mathcal{O}^*(2^m)$ -time brute-force algorithm is essentially optimal for non-constant w -Thiele rules in terms of parameter m . In contrast to CC, the gap for parameter n is much larger in the case of Thiele rules. Known algorithm that is FPT w.r.t. n for Thiele rules has double exponential dependence, namely $\mathcal{O}^*(2^{2^{\mathcal{O}(n)}})$ [9]. Narrowing the gap for the parameter n is an interesting research direction.

It is known that w -THIELE is polynomial-time solvable on SP profiles by, e.g., using an *Integer Linear Programming* formulation (ILP) which has *totally unimodular matrix* (TUM) [38]. Such ILP can be solved by a polynomial-time algorithm for *Linear Programming*.

Analogously to CC, we study the computational complexity of w -THIELE where a candidate-deletion set D of size d is given.

In Theorem 14 we show that w -THIELE is also FPT w.r.t. d , with an $\mathcal{O}^*(2^d)$ running time by checking all subsets of a candidate-deletion set as pre-elected winners. However, the way we solve an instance with pre-elected winners is completely different from what we do for CC (Theorem 9). In order to derive the above result we first define generalized Thiele rules in which each voter has a private Thiele sequence. Formally, the input of GENERALIZED THIELE is a superset of the input of regular w -THIELE and additionally there is also a *collection of n Thiele sequences*, one for each voter. (In contrast, the Thiele sequence is not an input to w -THIELE. We emphasize that w -THIELE is a collection of computational problems parameterized by a Thiele sequence w , hence we can provide results for specific Thiele sequences, e.g., in Theorem 12). Precisely, a collection of n Thiele sequences w is represented by a function of two arguments: $w : V \times \mathbb{N}^+ \rightarrow [0, 1]$, where $(w(v, i))_{i \in \mathbb{N}^+}$ is a Thiele sequence for each voter $v \in V$. For brevity we define $w_i^v = w(v, i)$. Now, utility of a voter v from a committee C' is defined as $u(v, C') = \sum_{i=1}^{|A_v \cap C'|} w_i^v$. As in w -THIELE, in GENERALIZED THIELE our task is to find a size- k subset $W \subseteq C$ such that the total utility $u(W)$ is at least U .

It is easy to see that any w -THIELE is a special case of GENERALIZED THIELE. Indeed, a given instance of w -THIELE is an instance of GENERALIZED THIELE where w is the Thiele sequence for each voter.

One immediate yet interesting result, is that GENERALIZED THIELE on SP profiles is polynomial-time solvable. The proof follows from a proper modification of the objective function of ILP presented by Peters [38].

Theorem 13. GENERALIZED THIELE on single-peaked preferences profiles can be solved in polynomial-time.

Proof. We use the approach presented by Peters [38] for w -THIELE, i.e. the ILP formulation labeled PAV-IP therein, which has totally unimodular matrix. We modify the objective function as follows. Instead of

$$\max \sum_{v \in V} \sum_{\ell=1}^k w_{\ell} \cdot x_{v,\ell},$$

we use

$$\max \sum_{v \in V} \sum_{\ell=1}^k w_{\ell}^v \cdot x_{v,\ell}.$$

A binary variable $x_{v,\ell}$ encodes information whether v has at least ℓ approved candidates in a solution. Clearly, such ILP solves GENERALIZED THIELE. Notice that the constraints are exactly the same as in PAV-IP, therefore the constraint matrix is totally unimodular for SP profiles, so we can solve GENERALIZED THIELE on SP profiles using any polynomial-time algorithm for Linear Programming. \square

Now we are ready to show our main result in this section, i.e., an algorithm which is FPT w.r.t. the size of a given candidate-deletion set that works not only for w -THIELE but also for GENERALIZED THIELE.

Theorem 14. We can solve GENERALIZED THIELE with a given candidate-deletion set of size d in time $\mathcal{O}^*(2^d)$. Furthermore, assuming the Exponential Time Hypothesis, there is no $\mathcal{O}^*(2^{o(d)})$ -time algorithm that solves the problem, even for any non-constant w -THIELE.

Proof. Let D be a given candidate-deletion set of size d . First, we guess pre-elected winners among D , call them W_D . Next we delete D from the instance, but then the proper modification of the instance with respect to the pre-elected winners is different with that in Theorem 9. Instead of modifying utility function we modify Thiele sequences for each voter depending on the number of approved candidates among pre-elected winners W_D .

For each voter $v \in V$ we define a new Thiele sequence such that $\hat{w}_i^v = w_{i+|A_v \cap W_D|}^v$ for every i . Note that a given instance of GENERALIZED THIELE with removed candidates from D and modified Thiele sequences as defined above is a GENERALIZED THIELE instance with an SP profile. Therefore, we can solve it in polynomial-time using Theorem 13.

In at least one case, when trying all subsets of D , we consider the maximum subset of an optimal solution that is included in D . In this case we find the remaining winners making a committee at least as good as an optimal one, so also optimal. Because of the guessing phase, the running time is clearly $\mathcal{O}^*(2^d)$.

Lower bound. Analogously to the lower bound proof of Theorem 9, for a given instance of non-constant w -THIELE we define D as a set of all candidates except arbitrary 2 candidates. After removing D from the instance, we get a profile with 2 candidates, hence an SP profile, so D is a candidate-deletion set. Then, an $\mathcal{O}^*(2^{o(d)})$ -time algorithm for GENERALIZED THIELE on our instance with a given candidate-deletion set D would solve I in time at most $\mathcal{O}^*(2^{o(d)}) \leq \mathcal{O}^*(2^{o(m)})$. This is a contradiction with ETH due to Theorem 12. \square

Analogously to the results for CC, GENERALIZED THIELE is polynomial-time solvable if d is logarithmic in the input size and is not polynomial-time solvable under ETH if d is slightly larger.

Corollary 15. GENERALIZED THIELE with a given candidate-deletion set of size d , where $d \leq \mathcal{O}(\log(nm))$, is polynomial-time solvable.

Theorem 16. *Under Exponential Time Hypothesis, there is no polynomial-time algorithm for GENERALIZED THIELE with a given candidate-deletion set of size at most $f(n, m)$ for any function $f(n, m) = \omega(\log(n))$ or $f(n, m) = \omega(\log(m))$, even for any non-constant w -THIELE.*

Proof. After we apply Theorem 12, the remaining proof is essentially the same as the proof of Theorem 11. The only difference is that in the case of adding dummy voters, which approve all the candidates, we have to increase the required utility bound properly. Indeed, each dummy voter will get exactly $\sum_{i=1}^k w_i$ score from any solution of size k . \square

6 Conclusions

We showed an almost optimal algorithm for CC-SP (Theorem 7). The bottleneck of the algorithm is finding (if not given) an SP-axis in the case where all the votes are given as weak orders. Finding an SP-axis on weak orders takes $\mathcal{O}(nm^2)$ time [27]—they reduce this problem to the Consecutive Ones problem and apply known algorithms for the latter problem. Can it be improved to $\mathcal{O}(nm)$ -time?

We showed that CC on general instances is FPT w.r.t. the size of a given candidate-deletion set D (Theorem 9). Moreover, the achieved $\mathcal{O}^*(2^{|D|})$ -time algorithm is essentially optimal under SETH. We adapted this result to Thiele rules (Theorem 13), but we do not know how much the base of the power in our $\mathcal{O}^*(2^{|D|})$ -time algorithm for Thiele rules can be improved. We know that, under ETH, we cannot hope for an $\mathcal{O}^*(2^{o(|D|)})$ -time algorithm. Can one provide a stronger lower bound (e.g., under SETH or SCC)?

OWA-based rules are generalization of both CC and Thiele rules. w -OWA-Winner rule generalizes w -THIELE by allowing any utility values (as in CC) instead of approval ballots [46]. It is tricky to define total utility that a voter gets from a committee as we have a sequence w and different utilities for all the members of a committee. It is defined by sorting these utilities and taking a dot product with w (the highest utility is multiplied by the highest weight etc.) A minimization version of the problem was studied under the name OWA k -Median [11]. An open question is whether such w -OWA-Winner is FPT w.r.t. the size of a given candidate-deletion set. Our approach from Theorem 14 does not seem to work because we need to know how to modify Thiele sequences for each voter when we pre-define winners.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments.

Virginia Vassilevska Williams was supported by NSF Grants CCF-2129139 and CCF-1909429, a BSF Grant BSF:2020356, a Google Research Fellowship and a Sloan Research Fellowship. Yinzhan Xu was partially supported by NSF Grant CCF-2129139. Krzysztof Sornat was partially supported by the SNSF Grant 200021_200731/1, the National Science Centre, Poland (NCN; grant number 2018/28/T/ST6/00366) and the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No 101002854). Part of this work was done while Krzysztof was a postdoc at MIT CSAIL, USA.

A conference version of this work appeared in IJCAI 2022 [47]. A full version of the paper is available at arXiv [48].

References

- [1] Alok Aggarwal and James K. Park. Notes on Searching in Multidimensional Monotone Arrays (Preliminary Version). In *Proceedings of the 29th Annual Symposium on Foundations of Computer Science (FOCS 1988)*, pages 497–512, 1988.
- [2] Alok Aggarwal, Baruch Schieber, and Takeshi Tokuyama. Finding a Minimum-Weight k -Link Path in Graphs with the Concave Monge Property and Applications. *Discret. Comput. Geom.*, 12:263–280, 1994.
- [3] Saeed Akhoondian Amiri. A Note on the Fine-Grained Complexity of MIS on Regular Graphs. *Inf. Process. Lett.*, 170:106123, 2021.
- [4] Haris Aziz, Serge Gaspers, Joachim Gudmundsson, Simon Mackenzie, Nicholas Mattei, and Toby Walsh. Computational Aspects of Multi-Winner Approval Voting. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015)*, pages 107–115, 2015.
- [5] Siddharth Barman, Omar Fawzi, and Paul Fermé. Tight Approximation Guarantees for Concave Coverage Problems. In *Proceedings of the 38th International Symposium on Theoretical Aspects of Computer Science (STACS 2021)*, volume 187 of *LIPICs*, pages 9:1–9:17, 2021.
- [6] Siddharth Barman, Omar Fawzi, Suprovat Ghoshal, and Emirhan Gürpınar. Tight Approximation Bounds for Maximum Multi-Coverage. *Math. Program.*, 192(1):443–476, 2022.
- [7] Nadja Betzler, Arkadii Slinko, and Johannes Uhlmann. On the Computation of Fully Proportional Representation. *J. Artif. Intell. Res.*, 47:475–519, 2013.
- [8] Robert Brederick, Jiehua Chen, Piotr Faliszewski, Jiong Guo, Rolf Niedermeier, and Gerhard J. Woeginger. Parameterized Algorithmics for Computational Social Choice: Nine Research Challenges. *Tsinghua Science and Technology*, 19(4):358–373, Aug 2014.
- [9] Robert Brederick, Piotr Faliszewski, Andrzej Kaczmarczyk, Dusan Knop, and Rolf Niedermeier. Parameterized Algorithms for Finding a Collective Set of Items. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI 2020)*, pages 1838–1845, 2020.
- [10] Markus Brill, Jean-François Laslier, and Piotr Skowron. Multiwinner Approval Rules as Apportionment Methods. *J. Theor. Politics*, 30(3):358–382, 2018. doi: 10.1177/0951629818775518.
- [11] Jarosław Byrka, Piotr Skowron, and Krzysztof Sornat. Proportional Approval Voting, Harmonic k -Median, and Negative Association. In *Proceedings of the 45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, volume 107 of *LIPICs*, pages 26:1–26:14, 2018.
- [12] John R. Chamberlin and Paul N. Courant. Representative Deliberations and Representative Decisions: Proportional Representation and the Borda Rule. *Am. Political Sci. Rev.*, 77:718–733, 9 1983.
- [13] Vincent Cohen-Addad and Jason Li. On the Fixed-Parameter Tractability of Capacitated Clustering. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *LIPICs*, pages 41:1–41:14, 2019.

- [14] Andrei Costin Constantinescu and Edith Elkind. Proportional Representation under Single-Crossing Preferences Revisited. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI 2021)*, pages 5286–5293, 2021.
- [15] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.
- [16] Marek Cygan, Holger Dell, Daniel Lokshtanov, Dániel Marx, Jesper Nederlof, Yoshio Okamoto, Ramamohan Paturi, Saket Saurabh, and Magnus Wahlström. On Problems as Hard as CNF-SAT. *ACM Trans. Algorithms*, 12(3):41:1–41:24, 2016.
- [17] Michael Dom, Jiong Guo, and Rolf Niedermeier. Approximation and Fixed-Parameter Algorithms for Consecutive Ones Submatrix Problems. *J. Comput. Syst. Sci.*, 76(3-4): 204–221, 2010.
- [18] Britta Dorn and Ildikó Schlotter. Having a Hard Time? Explore Parameterized Complexity! In Ulle Endriss, editor, *Trends in Computational Social Choice*, chapter 11, pages 209–230. AI Access, 2017.
- [19] Szymon Dudycz, Pasin Manurangsi, Jan Marcinkowski, and Krzysztof Sornat. Tight Approximation for Proportional Approval Voting. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI 2020)*, pages 276–282, 2020.
- [20] Edith Elkind and Martin Lackner. On Detecting Nearly Structured Preference Profiles. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI 2014)*, pages 661–667, 2014.
- [21] Edith Elkind, Piotr Faliszewski, Piotr Skowron, and Arkadii Slinko. Properties of Multiwinner Voting Rules. *Soc. Choice Welf.*, 48(3):599–632, 2017.
- [22] Edith Elkind, Martin Lackner, and Dominik Peters. Structured Preferences. In Ulle Endriss, editor, *Trends in Computational Social Choice*, chapter 10, pages 187–207. AI Access, 2017.
- [23] Edith Elkind, Piotr Faliszewski, and Piotr Skowron. A Characterization of the Single-Peaked Single-Crossing Domain. *Soc. Choice Welf.*, 54(1):167–181, 2020.
- [24] Gábor Erdélyi, Martin Lackner, and Andreas Pfandler. Computational Aspects of Nearly Single-Peaked Electorates. *J. Artif. Intell. Res.*, 58:297–337, 2017.
- [25] Piotr Faliszewski, Piotr Skowron, Arkadii Slinko, and Nimrod Talmon. Multiwinner Voting: A New Challenge for Social Choice Theory. In Ulle Endriss, editor, *Trends in Computational Social Choice*, chapter 2, pages 27–47. AI Access, 2017.
- [26] Uriel Feige. A Threshold of $\ln(n)$ for Approximating Set Cover. *J. ACM*, 45(4):634–652, 1998.
- [27] Zack Fitzsimmons and Martin Lackner. Incomplete Preferences in Single-Peaked Electorates. *J. Artif. Intell. Res.*, 67:797–833, 2020.
- [28] Sushmita Gupta, Pallavi Jain, Saket Saurabh, and Nimrod Talmon. Even More Effort Towards Improved Bounds and Fixed-Parameter Tractability for Multiwinner Rules. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI 2021)*, pages 217–223, 2021.

- [29] Yijie Han and Mikkel Thorup. Integer Sorting in $O(n \sqrt{\log \log n})$ Expected Time and Linear Space. In *Proceedings of the 43rd Symposium on Foundations of Computer Science (FOCS 2002)*, pages 135–144, 2002.
- [30] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which Problems Have Strongly Exponential Complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.
- [31] Pallavi Jain, Krzysztof Sornat, and Nimrod Talmon. Participatory Budgeting with Project Interactions. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI 2020)*, pages 386–392, 2020.
- [32] Subhash Khot and Oded Regev. Vertex Cover Might be Hard to Approximate to within $2-\epsilon$. *J. Comput. Syst. Sci.*, 74(3):335–349, 2008.
- [33] Subhash Khot, Dor Minzer, and Muli Safra. On Independent Sets, 2-to-2 Games, and Grassmann Graphs. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2017)*, pages 576–589, 2017.
- [34] D. Marc Kilgour. Approval Balloting for Multi-Winner Elections. In Jean-François Laslier and Remzi M. Sanver, editors, *Handbook on Approval Voting*, pages 105–124. Springer, 2010.
- [35] Pasin Manurangsi. Tight Running Time Lower Bounds for Strong Inapproximability of Maximum k -Coverage, Unique Set Cover and Related Problems (via t -Wise Agreement Testing Theorem). In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA 2020)*, pages 62–81, 2020.
- [36] Neeldhara Misra, Chinmay Sonar, and P. R. Vaidyanathan. On the Complexity of Chamberlin-Courant on Almost Structured Profiles. In *Proceedings of the 5th International Conference on Algorithmic Decision Theory (ADT 2017)*, volume 10576 of *LNCS*, pages 124–138, 2017.
- [37] Burt L. Monroe. Fully Proportional Representation. *Am. Political Sci. Rev.*, 89(4): 925–940, 1995.
- [38] Dominik Peters. Single-Peakedness and Total Unimodularity: New Polynomial-Time Algorithms for Multi-Winner Elections. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI 2018)*, pages 1169–1176, 2018.
- [39] Dominik Peters and Martin Lackner. Preferences Single-Peaked on a Circle. *J. Artif. Intell. Res.*, 68:463–502, 2020.
- [40] Dominik Peters, Lan Yu, Hau Chan, and Edith Elkind. Preferences Single-Peaked on a Tree: Multiwinner Elections and Structural Results. *J. Artif. Intell. Res.*, 73:231–276, 2022.
- [41] Ariel D. Procaccia, Jeffrey S. Rosenschein, and Aviv Zohar. Multi-Winner Elections: Complexity of Manipulation, Control and Winner-Determination. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pages 1476–1481, 2007.
- [42] Ariel D. Procaccia, Jeffrey S. Rosenschein, and Aviv Zohar. On the Complexity of Achieving Proportional Representation. *Soc. Choice Welf.*, 30(3):353–362, 2008.
- [43] Tomasz Przedmojski. Algorithms and Experiments for (Nearly) Restricted Domains in Elections. Master’s thesis, TU Berlin, 2016.

- [44] Baruch Schieber. Computing a Minimum Weight k -Link Path in Graphs with the Concave Monge Property. *J. Algorithms*, 29(2):204–222, 1998.
- [45] Piotr Skowron and Piotr Faliszewski. Chamberlin-Courant Rule with Approval Ballots: Approximating the MaxCover Problem with Bounded Frequencies in FPT Time. *J. Artif. Intell. Res.*, 60:687–716, 2017.
- [46] Piotr Skowron, Piotr Faliszewski, and Jérôme Lang. Finding a Collective Set of Items: From Proportional Multirepresentation to Group Recommendation. *Artif. Intell.*, 241: 191–216, 2016.
- [47] Krzysztof Sornat, Virginia Vassilevska Williams, and Yinzhan Xu. Near-Tight Algorithms for the Chamberlin-Courant and Thiele Voting Rules. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI 2022)*, pages 482–488, 2022.
- [48] Krzysztof Sornat, Virginia Vassilevska Williams, and Yinzhan Xu. Near-Tight Algorithms for the Chamberlin-Courant and Thiele Voting Rules. *CoRR*, abs/2212.14173, 2022.
- [49] T. N. Thiele. Om Flerfoldsvalg. In *Oversigt over det Kongelige Danske Videnskabernes Selskabs Forhandlinger (in Danish)*, pages 415–441. København: A.F. Høst, 1895.

A Proof of Lemma 5

Proof. By expanding the definition of $w(\cdot, \cdot)$ and focusing on the terms involving some $v \in V$, it suffices to show

$$\begin{aligned} & r(v, \{c_i, c_j\}) - r(v, \{c_i\}) + r(v, \{c_{i+1}, c_{j+1}\}) - r(v, \{c_{i+1}\}) \\ & \leq r(v, \{c_i, c_{j+1}\}) - r(v, \{c_i\}) + r(v, \{c_{i+1}, c_j\}) - r(v, \{c_{i+1}\}), \end{aligned}$$

which simplifies to

$$\begin{aligned} & r(v, \{c_i, c_j\}) - r(v, \{c_{i+1}, c_j\}) \\ & \leq r(v, \{c_i, c_{j+1}\}) - r(v, \{c_{i+1}, c_{j+1}\}). \end{aligned} \tag{1}$$

Since the profile is single-peaked, one of the following must happen.

1. $r(v, c_i) \geq r(v, c_{i+1}) \geq r(v, c_j)$. In this case, Equation (1) further simplifies to

$$r(v, c_j) - r(v, c_j) \leq r(v, \{c_i, c_{j+1}\}) - r(v, \{c_{i+1}, c_{j+1}\}).$$

The right hand side is non-negative because $r(v, c_i) \geq r(v, c_{i+1})$, so the inequality holds.

2. $r(v, c_i) \geq r(v, c_{i+1})$ and $r(v, c_j) \leq r(v, c_{j+1})$. In this case, we consider a function $f(x) = \min(r(v, c_i), x) - \min(r(v, c_{i+1}), x)$. Since $r(v, c_i) \geq r(v, c_{i+1})$, it is not hard to verify that $f(x)$ is non-decreasing. Since the left hand side of Inequality (1) is $f(r(v, c_j))$ and the right hand side is $f(r(v, c_{j+1}))$, the inequality holds because $r(v, c_j) \leq r(v, c_{j+1})$ and $f(x)$ is non-decreasing.
3. $r(v, c_{i+1}) \leq r(v, c_j) \leq r(v, c_{j+1})$. In this case, Equation (1) further simplifies to

$$r(v, \{c_i, c_j\}) - r(v, c_{i+1}) \leq r(v, \{c_i, c_{j+1}\}) - r(v, c_{i+1}),$$

which is true because $r(v, c_j) \leq r(v, c_{j+1})$ implies $r(v, \{c_i, c_j\}) \leq r(v, \{c_i, c_{j+1}\})$. □

B Proof of Lemma 6

Proof. Recall that the edge weight of an edge from i to j is $r(\{c_i, c_j\}) - r(\{c_i\})$. We can easily compute $r(\{c_i\})$ for every i in $\mathcal{O}(nm)$ time and store the result, so it remains to consider $r(\{c_i, c_j\})$, which was defined as

$$\sum_{v \in V} \min(r(v, c_i), r(v, c_j)).$$

Fix v , we let c_{l_v} be any of the candidates where $r(v, c_{l_v})$ is minimized. Then we define $f(i, j)$ as

$$\sum_{\substack{v \in V \\ r(v, c_i) > r(v, c_j) \text{ OR} \\ (r(v, c_i) = r(v, c_j) \text{ AND } i \leq l_v)}} r(v, c_j).$$

We define $g(i, j)$ as

$$\sum_{\substack{v \in V \\ r(v, c_i) < r(v, c_j) \text{ OR} \\ (r(v, c_i) = r(v, c_j) \text{ AND } i > l_v)}} r(v, c_i)$$

to cover the remaining cases. Clearly, $w(i, j) = f(i, j) + g(i, j)$, so we can focus on $f(i, j)$ from now on as we can handle $g(i, j)$ similarly.

We need to have the following claim.

Claim 17. *For any fixed j and v , the set of $i < j$ that satisfies $r(v, c_i) > r(v, c_j)$ OR $(r(v, c_i) = r(v, c_j)$ AND $i \leq l_v)$ always has the form $\{0, 1, \dots, p_{v,j}\}$ for some $p_{v,j}$. Furthermore, we can compute the values of $p_{v,j}$ for all v and j in $\mathcal{O}(nm)$ time in total.*

Proof. Let S be the set of $i < j$ such that $r(v, c_i) > r(v, c_j)$ OR $(r(v, c_i) = r(v, c_j)$ AND $i \leq l_v)$. We aim to show that if $x \in S$, then any integer y , where $0 \leq y < x$, is also in S .

Then there are two cases:

- In the first case, $r(v, c_x) > r(v, c_j)$. In this case, we must have $r(v, c_y) \geq r(v, c_x)$ because the profile is single-peaked. Therefore, $r(v, c_y) > r(v, c_j)$, and thus $p \in S$.
- In the second case, $r(v, c_x) = r(v, c_j)$. In this case, clearly $y \leq l_v$, so it suffices to show $r(v, c_y) \geq r(v, c_j) = r(v, c_x)$. For the sake of contradiction, suppose $r(v, c_y) < r(v, c_x)$. Since $y < x \leq l_v$, $r(v, c_y) < r(v, c_x)$ implies $r(v, c_x) \leq r(v, c_{l_v})$. Also, since $r(v, c_{l_v})$ is minimum, we must have $r(v, c_x) = r(v, c_{l_v})$, i.e., $r(v, c_x)$ is also a minimum, and it contradicts to $r(v, c_y) < r(v, c_x)$.

For each v , if we iterate j from 0 to $m + 1$, the value of $p_{v,j}$ will first (weakly) increase and then (weakly) decrease, due to single-peakedness. More formally, we aim to show that for any $j_1 < j_2 < j_3$, $p_{v,j_1} > p_{v,j_2} \implies p_{v,j_2} \geq p_{v,j_3}$. Assume for the sake of contradiction that $p_{v,j_1} > p_{v,j_2}$ and $p_{v,j_2} < p_{v,j_3}$ are both true. In other words, there exists i such that $i \leq p_{v,j_1}$ and $i \leq p_{v,j_3}$ while $i > p_{v,j_2}$. Since $i > p_{v,j_2}$, one of the following must happen:

- $i \geq j_2$. This is not possible since $i \leq p_{v,j_1}$ implies $i < j_1 < j_2$.
- $i < j_2$ and $r(v, c_i) < r(v, c_{j_2})$. Since $i < j_2 < j_3$, the single-peakedness of the profile implies that $r(v, c_{j_2}) \leq r(v, c_{j_3})$. Therefore, $r(v, c_i) < r(v, c_{j_3})$, which contradicts to $i \leq p_{v,j_3}$.
- $i < j_2$, $r(v, c_i) = r(v, c_{j_2})$ and $i > c_{l_v}$. In this case, if $r(v, c_i)$ equals $r(v, c_{j_1})$, i must be less than or equal to c_{l_v} , thus leading to a contradiction. Therefore, $r(v, c_i) > r(v, c_{j_1})$. Thus, $r(v, c_{j_2}) > r(v, c_{j_1})$. By single-peakedness, $r(v, c_{j_3}) \geq r(v, c_{j_2})$. Thus, $r(v, c_{j_3}) \geq r(v, c_i)$. Since $i \leq p_{v,j_3}$, it is not possible that $r(v, c_{j_3}) > r(v, c_i)$. Therefore, $r(v, c_{j_3}) = r(v, c_i)$ and $i \leq c_{l_v}$, leading to a contradiction.

Therefore, by increasing or decreasing the value of $p_{v,j}$ accordingly after we increment j , we can efficiently compute $p_{v,j}$ for all j . The total amount of changes to the values of $p_{v,j}$ over all values of j for a fixed v is $\mathcal{O}(m)$. Thus, this takes $\mathcal{O}(nm)$ time overall to compute all $p_{v,j}$. \square

Now we can use $p_{v,j}$ to reformulate $f(i, j)$: for any $i < j$, $f(i, j) = \sum_{v \in V, i \leq p_{v,j}} r(v, c_j)$. Conceptually, for every j , we consider n numbers $p_{v,j}$, where $p_{v,j}$ has a weight $r(v, c_j)$. Then for each i , we just need to report the total weights of numbers that are at least i . To achieve this task, we first sort n numbers $n \cdot p_{v,j} + v$ for $v \in V$. We will be able to retrieve v from each $n \cdot p_{v,j} + v$ in the sorted array, and thus retrieve $r(v, c_j)$. Then we compute the suffix sums of $r(v, c_j)$ in the sorted order. For every query i , we can easily find the smallest $p_{v,j}$ that is greater than or equal to i in $\mathcal{O}(\log(m))$ time (for instance, we can first remove duplicated $p_{v,j}$ during pre-processing to guarantee that there are at most m of them, then use binary search to find such a $p_{v,j}$), then use the suffix sum associated with it as the answer.

Therefore, the time complexity for the pre-processing is upper bounded by the cost of sorting n non-negative integers bounded by $\mathcal{O}(nm)$, multiplied by a factor of m , and the query time is $\mathcal{O}(\log(m))$. \square

C Proof of Theorem 9

Proof. Let $I = (V, C, r, k)$ be an instance of the optimization version of CC and D be a candidate-deletion set of size d . For the sake of analysis let us fix an optimal solution W^* to the instance I .

Algorithm. First, we would like to find $W_D^* = W^* \cap D$ which we call *pre-elected winners*. It is enough to consider all subsets $W_D \subseteq D$ of size at most k . We have at most $\sum_{i=0}^{\min(k,d)} \binom{d}{i} \leq 2^d$ many such subsets and W_D^* is one of them. For each subset W_D , we will remove D from the instance and find remaining $k' = k - |W_D|$ winners that minimizes total misrepresentation of a committee consisted of the members of W_D and the remaining k' candidates. We will store all committees constructed for each considered W_D and we will return the committee with the smallest total misrepresentation. Notice that in the case where $W_D = W_D^*$ we will construct W^* (or another optimal solution to the instance I) as one of the potential solutions. Below, we describe how to fill optimally a pre-elected committee W_D .

If $|W_D| = k$ then we simply save W_D as one of potential solutions of our algorithm. Otherwise, if $|W_D| < k$, we need to choose additional $k' = k - |W_D|$ winners to the committee among a subset $C' = C \setminus D$. For this purpose, we define a new instance $I' = (V, C', r', k')$ of CC, where $r' : V \times C' \rightarrow \mathbb{R}_{\geq 0}$ is defined as follows:

$$r'(v, c) = \min(r(v, W_D), r(v, c)) = r(v, W_D \cup \{c\}).$$

The misrepresentation function r' is constructed in such a way that misrepresentation of v from a candidate $c \in C'$ is not greater than misrepresentation of v by any pre-elected candidate, i.e., a candidate from W_D . Moreover, we have $r'(v, W') = r(v, W_D \cup W')$. It means that an optimal solution to I' has the same misrepresentation as an optimal extension of the pre-elected committee W_D , as desired. Note that, when $W_D = W_D^*$, an optimal solution to I' has the same total misrepresentation as an optimal solution W^* . Below we show formally that the obtained preference profile (V, C', r') is single-peaked.

Lemma 18. *The preference profile (V, C', r') is single-peaked.*

Proof. By the definition of D , we know that $(V, C \setminus D, r_{-D})$ is SP. It means that there exists a linear order \prec over $C \setminus D = C'$ such that r_{-D} has the single-peakedness conditions. We will show that the linear order \prec is also an SP-axis for r' . For this purpose, let us consider a triple of distinct candidates $a, b, c \in C'$ such that $a \prec b \prec c$ and let $v \in V$ be any voter. If $r'(v, a) < r'(v, b)$ then by the definition of r' we have $r(v, W_D \cup \{a\}) < r(v, W_D \cup \{b\})$. Due to the strictness of the inequality and the definition of $r(v, \cdot)$ we obtain $r(v, a) < r(v, b)$. Single-peakedness of (V, C', r) on the SP-axis \prec implies $r(v, b) \leq r(v, c)$. From the definition of $r(v, \cdot)$ we obtain $r(v, W_D \cup \{b\}) \leq r(v, W_D \cup \{c\})$ and, by the definition of r' , this is equivalent to $r'(v, b) \leq r'(v, c)$ as desired in the condition for single-peakedness for (V, C', r') . The condition when $c \prec b \prec a$ can be proven analogously. \square

Therefore, the new instance is an CC-SP instance which can be solved in polynomial-time, e.g., using our algorithm from Theorem 7. In such a way, we find the remaining k' winners optimally.

Running time. The running time comes from a consideration of all subsets of D of size at most k (there are at most 2^d such subsets) and, for each such subset, run an algorithm that solves an CC-SP instance in polynomial-time.

Lower bound. Let $I = (V, C, r, k)$ be an instance of Approval-CC. Let D be any size- $(m-2)$ subset of C . Note that D is a candidate-deletion set, as a profile with two candidates is always SP. Then, an $\mathcal{O}^*((2-\varepsilon)^d)$ -time algorithm for Approval-CC on an instance I with a given candidate-deletion set D , for some $\varepsilon > 0$, would solve I in time at most $\mathcal{O}^*((2-\varepsilon)^d) \leq \mathcal{O}^*((2-\varepsilon)^m)$. This is a contradiction with SETH due to Observation 8. \square

D Proof of Theorem 12

Proof. First, we provide a formal construction of the reduction from IS to w -THIELE for any non-constant Thiele sequence w . Its idea was described in [31] for an even more general problem (related to a participatory budgeting problem which allows interactions between items we choose). For completeness we include a simplified construction of the reduction tailored for our needs.

Reduction. Let (G, K) be an instance of IS, where G is a 3-regular graph with a vertex set $V(G)$ and an edge set $E(G)$, and K is a positive integer. For a fixed Thiele sequence w we construct an instance of w -THIELE as follows. For each vertex $v \in V(G)$ we define a vertex-candidate c_v . As w is a fixed non-constant Thiele sequence, there exists a minimum constant $\ell \in \{2, 3, \dots\}$ such that $w_\ell < 1$. We add $\ell - 2$ dummy candidates $C_D = \{d_1, d_2, \dots, d_{\ell-2}\}$ to the instance. Hence, the total number of candidates equals $m = |V(G)| + \ell - 2 = |V(G)| + \mathcal{O}(1)$. For each edge $e \in E(G)$ we define a voter v_e which approves exactly two candidates that are not dummy corresponding to the endpoints x and y of e and all dummy candidates, i.e., $A_{v_e} = \{x, y\} \cup C_D$. The number of voters equals $n = E(G) = \frac{3}{2}V(G)$. We define the desired size of the committee as $k = K + \ell - 2$ and the utility bound as $U = (\ell - 2)E(G) + 3K$. Clearly, it is a polynomial-time reduction.

Correctness. Let V' be a size- K independent set of G . We use $C(V')$ to denote the set of candidates corresponding to vertices from V' . We define a solution W to the constructed instance of w -THIELE as candidates from $C(V')$ and all dummy candidates, i.e., $W = C(V') \cup C_D$. We will show that this is a feasible solution. Clearly $|W| = K + \ell - 2 = k$. The total utility achieved by the solution W consists of: $|E(G)| \sum_{i=1}^{\ell-2} w_i = |E(G)|(\ell - 2)$ from dummy candidates and $3K$ from C_D as each vertex from the corresponding size- K independent set covers exactly 3 edges and none of the edges is covered twice (hence each voter is covered at most $\ell - 1$ times). Formally, the total utility achieved by W can be computed as follows:

$$\begin{aligned}
u(W) &= \sum_{e \in E(G)} u(v_e, W) = \sum_{e \in E(G)} \sum_{i=1}^{|A_{v_e} \cap W|} w_i \\
&= \sum_{\{x, y\} \in E(G)} \sum_{i=1}^{|\{c_x, c_y\} \cup C_D \cap (C(V') \cup C_D)|} w_i \\
&= \sum_{\{x, y\} \in E(G)} \sum_{i=1}^{|\{c_x, c_y\} \cap C(V')| + \ell - 2} w_i \\
&= (\ell - 2)|E(G)| + \sum_{\{x, y\} \in E(G)} \sum_{i=1}^{|\{c_x, c_y\} \cap C(V')|} w_{i+\ell-2} \\
&= (\ell - 2)|E(G)| + \sum_{\{x, y\} \in E(G)} \sum_{i=1}^{|\{x, y\} \cap V'|} w_{i+\ell-2} \\
&\stackrel{(*)}{=} (\ell - 2)|E(G)| + \sum_{\{x, y\} \in E(G)} |\{x, y\} \cap V'| \cdot w_{\ell-1} \\
&\stackrel{(**)}{=} (\ell - 2)|E(G)| + 3K \cdot 1 = U,
\end{aligned}$$

where (*) follows because V' is an independent set, so it is never true that $|\{x, y\} \cap V'| = 2$, and (**) follows from the fact that $w_{\ell-1} = 1$ and the number of incident edges to V' is $3K$ as G is a 3-regular graph and $|V'| = K$. Hence, W is a solution to the constructed instance.

In the other direction, let W be a feasible solution to the constructed instance. First, we notice that $C_D \subseteq W$. Otherwise, the total utility achieved by W would be at most: $(\ell-3)|E(G)| + 3(K+1) = (\ell-2)|E(G)| + 3K + (3 - |E(G)|) < U$. We define a size- K subset of $V(G)$ as vertices that correspond to $W \setminus C_D$ and next, we argue that the constructed subset is an independent set. Indeed, if it is not the case, then there exists an edge $e = (x, y) \in E(G)$ such that $c_x, c_y \in W \setminus C_D$, hence the voter v_e is covered exactly ℓ times. Then, the total utility achieved by W is at most: $(\ell-2)|E(G)| + 3K - 1 + w_\ell = U + (w_\ell - 1) < U$. It is a contradiction with the fact that W is a feasible solution.

Lower bound. To achieve a lower bound, let us assume there is an $\mathcal{O}^*(2^{o(m)})$ -time algorithm for a fixed non-constant w -THIELE. Using the reduction presented above we can solve IS on a 3-regular graph G in time: $\mathcal{O}^*(2^{o(m)}) = \mathcal{O}^*(2^{o(|V(G)| + \mathcal{O}(1)}) = \mathcal{O}^*(2^{o(|V(G)|)})$, which contradicts ETH [3, Theorem 5].

Analogously we prove the other lower bound using the fact that $n = \frac{3}{2}|V(G)|$. \square

Krzysztof Sornat
 IDSIA, USI-SUPSI
 Lugano, Switzerland
 Email: krzysztof.sornat@idsia.ch

Virginia Vassilevska Williams
 MIT CSAIL
 Cambridge, MA, USA
 Email: virgi@mit.edu

Yinzhan Xu
 MIT CSAIL
 Cambridge, MA, USA
 Email: xyzhan@mit.edu