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

**Krzysztof Sornat**[1*], **Virginia Vassilevska Williams**[2], **Yinzhan Xu**[2]
[1]AGH University, Poland
[2]MIT CSAIL, USA
sornat@agh.edu.pl, virgi@mit.edu, xyzhan@mit.edu

## 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. 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 $2^{|D|} \cdot \mathrm{poly}(n,m)$ 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) [Chamberlin and Courant, 1983] with general misrepresentation function and the Thiele rules [Thiele, 1895] 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 $w_1 + w_2 + \cdots + w_x$, 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 [Faliszewski *et al.*, 2017; Brill *et al.*, 2018]. 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 [Skowron *et al.*, 2016].

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 [Procaccia *et al.*, 2008]. This special case is called *Approval Chamberlin-Courant* (Approval-CC) and it was considered by Procaccia et al. [2008] 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 [Skowron and Faliszewski, 2017]. In particular, MAX $k$-COVERAGE is NP-hard to approximate within $(1 - 1/e + \varepsilon)$ factor for any $\varepsilon > 0$ [Feige, 1998].

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* [Kilgour, 2010]. Notice that $(1, 1, \dots)$-Thiele is equivalent to *Multiwinner Approval Voting* [Aziz *et al.*, 2015], which is solvable in polynomial-time. All the other ones[1] are NP-hard [Skowron *et al.*, 2016, 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 $w$ [Dudycz *et al.*, 2020; Barman *et al.*, 2021].

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 [Elkind *et al.*, 2017b]. Such restricted domains are motivated by real-world elections. For instance, SP preference profiles appear when, intuitively, the candidates can be placed on an 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 [Betzler *et al.*, 2013; Peters, 2018]. However, the dynamic programming algorithm for CC-SP proposed by Betzler et al. [2013] is not optimal in terms of running time. We fill this gap by showing an almost optimal algorithm and we study polynomial-time solvability for nearly single-peaked preference profiles for CC and Thiele rules.

[*]Part of this work was done while Krzysztof was a postdoc at MIT CSAIL, USA.

[1]We assume $w_1 = 1$ without loss of generality.

## 1.1 Our Contribution

The previous best running time for CC-SP was $\mathcal{O}(nm^2)$ [Betzler *et al.*, 2013]. We improve it by presenting an almost linear time algorithm (up to subpolynomial factors) in Theorem 6. 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 4). 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 [1998] 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 [1988] to achieve $\mathcal{O}(m\log(m))$ time and oracle accesses. Overall, the running time is $\mathcal{O}(nm\log(n) + m^{1+o(1)}\log(m) + m\log^2(m)) = \tilde{\mathcal{O}}(nm)$.

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 [Aggarwal *et al.*, 1994]) 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 [Elkind *et al.*, 2017b, 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).

We prove that CC is FPT with respect to $d$ (Theorem 8). Our algorithm runs in $\mathcal{O}^*(2^d)$-time. 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 9 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 [2018] 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.

Due to space limitation, we will defer some proofs to the full version of the paper.

## 1.2 Related Work

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

**The paper of Constantinescu and Elkind [2021].** Recently, Constantinescu and Elkind 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 [Schieber, 1998], 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 [Elkind *et al.*, 2020], it is potentially more efficient to use our algorithm. The difference enlarges if an SP-axis and an SC-axis are not given. Indeed, for ballots given as linear orders[2] the problem of finding an SC-axis is known to be computable in $\mathcal{O}(nm^2)$ time [Bredereck *et al.*, 2013] while finding an SP-axis can be done in $\mathcal{O}(nm)$ (even if only one vote is a linear order) [Fitzsimmons and Lackner, 2020].

**The paper of Misra, Sonar, and Vaidyanathan [2017].** Our $\mathcal{O}^*(2^d)$-time algorithm for CC-SP is essentially the same as an algorithm presented by Misra et al. [2017]. The main idea for both algorithms is to try all subsets of a given candidate-deletion set as winners and compute remaining winners. However, this construction is well-suited for generalization to Thiele rules, so we keep the proof 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.

[2] It is the case for, e.g., utility functions defined as *committee scoring rules* [Elkind *et al.*, 2017a] such as *Borda scores*.

**Other nearly SP measures.** Different notions of nearly single-peaked preference profiles have been proposed [Elkind *et al.*, 2017b; Erdélyi *et al.*, 2017]. 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 [Erdélyi *et al.*, 2017]. Hence, some FPT and approximation algorithms have been proposed [Elkind and Lackner, 2014].

**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 [Peters and Lackner, 2020].

## 2 Preliminaries

First we introduce some notations and define the computational problems formally. The $\tilde{\mathcal{O}}$ 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* $r : V \times C \to \mathbb{R}_{\geq 0}$ 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* $u : V \times C \to \mathbb{R}_{\geq 0}$ that indicates how well a voter is represented by a particular candidate. We can define it as: $u(v, c) = \max_{c' \in C}(r(v, c')) - r(v, c)$. 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 $C_v$ of candidates that $v$ approves. In the case of approval voting we obtain an *approval misrepresentation function*, i.e., $r(v, c) = 0$ iff $c \in C_v$ and $r(v, c) = 1$ iff $c \notin C_v$.

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*.

[3]We 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 [1983] defined a voting rule on the Borda misrepresentation function (currently, it is often called Borda-CC). In this paper, analogously to Betzler et al. [2013], 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 sequence[4] $w = (w_1, w_2, \dots)$, with $w_1 = 1$, as follows. We are given an election $(V, C)$, approval ballots $C_v$ 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 [Byrka *et al.*, 2018].

**Single-peaked preference profiles.** Following Proposition 3 in [Betzler *et al.*, 2013] 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 $r_{-D} : V \times (C \setminus D) \to \mathbb{R}_{\geq 0}$ is defined as $r$ restricted to $V \times (C \setminus D)$. Typically we use $d = |D|$.

**Parameterized complexity, ETH and SETH.** We assume basic knowledge of parameterized complexity as this is common in computational social choice [Bredereck *et al.*, 2014; Dorn and Schlotter, 2017]. For a textbook on the topic, see e.g. [Cygan *et al.*, 2015]. To provide lower bounds we will use two popular conjectures: Exponential Time Hypothesis

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

[5]Sometimes called *possibly single-peaked* [Elkind *et al.*, 2017b] as in this paper we allow ties (weak orders).

(ETH) and its stronger version—Strong Exponential Time Hypothesis (SETH). For formal statements see, e.g., Conjectures 14.1 and 14.2 in [Cygan *et al.*, 2015].

## 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 [Fitzsimmons and Lackner, 2020]. 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. [2013].

We first label the candidates so that $c_1 \prec c_2 \prec \cdots \prec c_m$ with respect to 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 addition, 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 1** (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 $\{0, \ldots, m+1\}$. For every $i, j$ where $0 \le i < j \le m{+}1$, 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 2.** *For any $v \in V$, $0 \le i < j \le m + 1$, and for any $C' \subseteq \{c_0, \ldots, c_{i-1}\}$, it holds $r(v, \{c_i, c_j\}) - r(v, \{c_i\}) = r(v, C' \cup \{c_i, c_j\}) - r(v, C' \cup \{c_i\})$.*

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

**Lemma 3.** *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 $a_0, a_1, \ldots, a_k, a_{k+1}$ for some $0 = a_0 < a_1 < \ldots < a_k < a_{k+1} = m + 1$. The weight of path $P$ is thus

$$\sum_{i=0}^{k} w(a_i, a_{i+1}) \overset{\text{def.}}{=} \sum_{v \in V} \sum_{i=0}^{k} r(v, \{c_{a_i}, c_{a_{i+1}}\}) - r(v, \{c_{a_i}\})$$

$$\overset{\text{Claim 2}}{=} \sum_{v \in V} \sum_{i=0}^{k} r(v, \{c_{a_j}\}_{0 \le j \le i+1}) - r(v, \{c_{a_j}\}_{0 \le j \le i})$$

$$= \sum_{v \in V} r(v, \{c_{a_j}\}_{0 \le j \le m+1}) - r(v, \{c_{a_j}\}_{0 \le j \le 0})$$

$$= -Un + \sum_{v \in V} r(v, \{c_{a_1}, \ldots, c_{a_k}\}),$$

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

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. □

By Lemma 3, 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 4.** *The edge weights of the Minimum Weight $t$-Link Path instance satisfy the concave Monge property, i.e., for all $1 \le i + 1 < j \le m$, it holds $w(i, j) + w(i + 1, j + 1) \le w(i, j + 1) + w(i + 1, j)$.*

By Lemma 4, we can use the $m^{1+o(1)}$ time algorithm for the Minimum Weight $t$-Link Path instance (we use the algorithm by Schieber [1998] for the case where $k = \Omega(m)$ and use the algorithm by Aggarwal and Park [1988] 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 this subpolynomial factor.

**Lemma 5.** *After an $\mathcal{O}(m\,\mathrm{IntSort}(n, nm))$ time pre-processing where $\mathrm{IntSort}(n, nm)$ 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.*

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 5 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 $\mathrm{IntSort}(n, nm) = \mathcal{O}(n\sqrt{\log\log(n)})$ [Han and Thorup, 2002]. Also, in a common case where $m = \mathrm{poly}(n)$, $\mathrm{IntSort}(n, nm) = \mathcal{O}(n)$ 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 5 with previous ideas we obtain the following theorem.

**Theorem 6.** *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. [2013] 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 $U = \{u_1, \dots, u_m\}$, 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. [2013] works also 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.

Below, we observe SETH-hardness of CC, due to SETH-hardness of HS [Cygan *et al.*, 2016] and the reduction of Betzler et al. [2013].

**Observation 7.** *Assuming 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 6). 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 computational complexity of Min-COS-C see, e.g., [Dom *et al.*, 2010]. An interesting hardness result for Min-COS-C described therein is that an $\alpha$-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 [Khot *et al.*, 2017]. 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 [Erdélyi *et al.*, 2017]. This holds, e.g., when the misrepresentation function is derived from *a committee scoring rule* [Elkind *et al.*, 2017a] which 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 set[6] (denoted by $D$) and the obtained algorithm is essentially optimal assuming 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 6.

**Theorem 8.** *We can solve* CHAMBERLIN-COURANT *with a given candidate-deletion set of size $d$ in time $\mathcal{O}^*(2^d)$. Furthermore, assuming 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.*

A simple corollary from Theorem 8 is that CC is polynomial-time solvable if $d$ is logarithmic in the input size.

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 9.** *Under 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 $f(n, m) = \omega(\log(m))$.*

## 5 Nearly Single-Peakedness and Thiele 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* [Skowron *et al.*, 2016, 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 [Aziz *et al.*, 2015] for the above-mentioned *Proportional Approval Voting*. Actually, their reduction works for any $w$-THIELE with $w_2 < 1$. The idea was extended to other non-constant $w$-Thiele rules when $w_2 = 1$ by introducing a proper number of dummy candidates [Jain *et al.*, 2020]. Jain et al. 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 10 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 $2^{o(|V(G)|)}$ time algorithms under ETH [Amiri, 2021, Theorem 5], we obtain the following theorem.

**Theorem 10.** *Assuming Exponential Time Hypothesis, there is neither $\mathcal{O}^*(2^{o(m)})$ nor $\mathcal{O}^*(2^{o(n)})$-time algorithm for any non-constant $w$-*THIELE.

[6]This result was presented before by Misra et al. [2017], however our construction is well-suited for generalization to Thiele rules presented in Section 5, so we keep it for completeness.

It means that the $\mathcal{O}^*(2^m)$-time brute-force algorithm is essentially optimal for non-constant $w$-Thiele rules in terms of parameter $m$. Known algorithm that is FPT w.r.t. $n$ for Thiele rules has double exponential dependence, namely $\mathcal{O}^*(2^{2^{\mathcal{O}(n)}})$ [Bredereck *et al.*, 2020]. 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) [Peters, 2018]. 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 12 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 8). 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 10). Precisely, a collection of $n$ Thiele sequences $w$ is represented by a function of two arguments: $w : V \times \mathbb{N}^+ \to [0, 1]$, 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 while $w$ 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 [2018].

**Theorem 11.** GENERALIZED THIELE *on single-peaked preferences profiles can be solved in polynomial-time.*

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

The idea for the algorithm is as follows. 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 required. We modify Thiele sequences for each voter depending on the number of approved candidates among pre-elected winners $W_D$. Precisely, for each voter $v \in V$ we define a new Thiele sequence such that $\hat{w}_i^v = w_{i+|C_v \cap W_D|}^v$ for every $i$. Such a modified instance is SP and is an instance of GENERALIZED THIELE. Hence, we can use a polynomial-time algorithm from Theorem 11.

The lower bound proof is analogous to the lower bound proof of Theorem 8 but derived from the ETH-based lower bound from Theorem 10.

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.

**Theorem 13.** *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.

# 6 Conclusions

We showed an almost optimal algorithm for CC-SP (Theorem 6). 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 [Fitzsimmons and Lackner, 2020]. 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 8). Moreover, the achieved $\mathcal{O}^*(2^{|D|})$-time algorithm is essentially optimal under SETH. We adapted this result to Thiele rules (Theorem 11), 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 $\mathcal{O}^*(2^{o(|D|)})$-time algorithm. Can one provide a stronger lower bound (e.g., under SETH or SCC)?

# Acknowledgements

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).

# References

[Aggarwal and Park, 1988] A. Aggarwal and J. K. Park. Notes on searching in multidimensional monotone arrays. In *FOCS*, pages 497–512, 1988.

[Aggarwal *et al.*, 1994] A. Aggarwal, B. Schieber, and T. Tokuyama. Finding a minimum-weight k-link path in graphs with the concave Monge property and applications. *Discret. Comput. Geom.*, 12:263–280, 1994.

[Amiri, 2021] S. A. Amiri. A note on the fine-grained complexity of MIS on regular graphs. *Inf. Process. Lett.*, 170:106123, 2021.

[Aziz *et al.*, 2015] H. Aziz, S. Gaspers, J. Gudmundsson, S. Mackenzie, N. Mattei, and T. Walsh. Computational aspects of multi-winner approval voting. In *AAMAS*, pages 107–115, 2015.

[Barman *et al.*, 2021] S. Barman, O. Fawzi, and P. Fermé. Tight approximation guarantees for concave coverage problems. In *STACS*, pages 9:1–9:17, 2021.

[Betzler *et al.*, 2013] N. Betzler, A. Slinko, and J. Uhlmann. On the computation of fully proportional representation. *J. Artif. Intell. Res.*, 47:475–519, 2013.

[Bredereck *et al.*, 2013] R. Bredereck, J. Chen, and G. J. Woeginger. A characterization of the single-crossing domain. *Soc. Choice Welf.*, 41(4):989–998, 2013.

[Bredereck *et al.*, 2014] R. Bredereck, J. Chen, P. Faliszewski, J. Guo, R. Niedermeier, and G. J. Woeginger. Parameterized algorithmics for computational social choice: Nine research challenges. *Tsinghua Science and Technology*, 19(4):358–373, Aug 2014.

[Bredereck *et al.*, 2020] R. Bredereck, P. Faliszewski, A. Kaczmarczyk, D. Knop, and R. Niedermeier. Parameterized algorithms for finding a collective set of items. In *AAAI*, pages 1838–1845, 2020.

[Brill *et al.*, 2018] M. Brill, J.-F. Laslier, and P. Skowron. Multiwinner approval rules as apportionment methods. *J. Theor. Politics*, 30(3):358–382, 2018.

[Byrka *et al.*, 2018] J. Byrka, P. Skowron, and K. Sornat. Proportional approval voting, harmonic k-median, and negative association. In *ICALP*, pages 26:1–26:14, 2018.

[Chamberlin and Courant, 1983] J. R. Chamberlin and P. N. Courant. Representative deliberations and representative decisions: Proportional representation and the Borda rule. *Am. Political Sci. Rev.*, 77:718–733, 9 1983.

[Constantinescu and Elkind, 2021] A. C. Constantinescu and E. Elkind. Proportional representation under single-crossing preferences revisited. In *AAAI*, pages 5286–5293, 2021.

[Cygan *et al.*, 2015] M. Cygan, F. V. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.

[Cygan *et al.*, 2016] M. Cygan, H. Dell, D. Lokshtanov, D. Marx, J. Nederlof, Y. Okamoto, R. Paturi, S. Saurabh, and M. Wahlström. On problems as hard as CNF-SAT. *ACM Trans. Algorithms*, 12(3):41:1–41:24, 2016.

[Dom *et al.*, 2010] M. Dom, J. Guo, and R. Niedermeier. Approximation and fixed-parameter algorithms for consecutive ones submatrix problems. *J. Comput. Syst. Sci.*, 76(3-4):204–221, 2010.

[Dorn and Schlotter, 2017] B. Dorn and I. Schlotter. Having a hard time? Explore parameterized complexity! In U. Endriss, editor, *Trends in Computational Social Choice*, chapter 11, pages 209–230. AI Access, 2017.

[Dudycz *et al.*, 2020] S. Dudycz, P. Manurangsi, J. Marcinkowski, and K. Sornat. Tight approximation for proportional approval voting. In *IJCAI*, pages 276–282, 2020.

[Elkind and Lackner, 2014] E. Elkind and M. Lackner. On detecting nearly structured preference profiles. In *AAAI*, pages 661–667, 2014.

[Elkind *et al.*, 2017a] E. Elkind, P. Faliszewski, P. Skowron, and A. Slinko. Properties of multiwinner voting rules. *Soc. Choice Welf.*, 48(3):599–632, 2017.

[Elkind *et al.*, 2017b] E. Elkind, M. Lackner, and D. Peters. Structured preferences. In U. Endriss, editor, *Trends in Computational Social Choice*, chapter 10, pages 187–207. AI Access, 2017.

[Elkind *et al.*, 2020] E. Elkind, P. Faliszewski, and P. Skowron. A characterization of the single-peaked single-crossing domain. *Soc. Choice Welf.*, 54(1):167–181, 2020.

[Erdélyi *et al.*, 2017] G. Erdélyi, M. Lackner, and A. Pfandler. Computational aspects of nearly single-peaked electorates. *J. Artif. Intell. Res.*, 58:297–337, 2017.

[Faliszewski *et al.*, 2017] P. Faliszewski, P. Skowron, A. Slinko, and N. Talmon. Multiwinner voting: A new challenge for social choice theory. In U. Endriss, editor, *Trends in Computational Social Choice*, chapter 2, pages 27–47. AI Access, 2017.

[Feige, 1998] U. Feige. A threshold of ln(n) for approximating set cover. *J. ACM*, 45(4):634–652, 1998.

[Fitzsimmons and Lackner, 2020] Z. Fitzsimmons and M. Lackner. Incomplete preferences in single-peaked electorates. *J. Artif. Intell. Res.*, 67:797–833, 2020.

[Han and Thorup, 2002] Y. Han and M. Thorup. Integer sorting in O(n sqrt(log log n)) expected time and linear space. In *FOCS*, pages 135–144, 2002.

[Jain *et al.*, 2020] P. Jain, K. Sornat, and N. Talmon. Participatory budgeting with project interactions. In *IJCAI*, pages 386–392, 2020.

[Khot *et al.*, 2017] S. Khot, D. Minzer, and M. Safra. On independent sets, 2-to-2 games, and Grassmann graphs. In *STOC*, pages 576–589, 2017.

[Kilgour, 2010] D. M. Kilgour. Approval balloting for multi-winner elections. In J.-F. Laslier and R. M. Sanver, editors, *Handbook on Approval Voting*, pages 105–124. Springer, 2010.

[Misra *et al.*, 2017] N. Misra, C. Sonar, and P. R. Vaidyanathan. On the complexity of Chamberlin-Courant on almost structured profiles. In *ADT*, pages 124–138, 2017.

[Peters and Lackner, 2020] D. Peters and M. Lackner. Preferences single-peaked on a circle. *J. Artif. Intell. Res.*, 68:463–502, 2020.

[Peters, 2018] D. Peters. Single-peakedness and total unimodularity: New polynomial-time algorithms for multi-winner elections. In *AAAI*, pages 1169–1176, 2018.

[Procaccia *et al.*, 2008] A. D. Procaccia, J. S. Rosenschein, and A. Zohar. On the complexity of achieving proportional representation. *Soc. Choice Welf.*, 30(3):353–362, 2008.

[Schieber, 1998] B. Schieber. Computing a minimum weight k-link path in graphs with the concave Monge property. *J. Algorithms*, 29(2):204–222, 1998.

[Skowron and Faliszewski, 2017] P. Skowron and P. 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.

[Skowron *et al.*, 2016] P. Skowron, P. Faliszewski, and J. Lang. Finding a collective set of items: From proportional multirepresentation to group recommendation. *Artif. Intell.*, 241:191–216, 2016.

[Thiele, 1895] T. N. Thiele. Om Flerfoldsvalg. In *Oversigt over det Kongelige Danske Videnskabernes Selskabs Forhandlinger (in Danish)*, pages 415–441. København: A.F. Høst, 1895.