Polynomial-Time Algorithm for Thiele Voting Rules with Voter Interval Preferences

Pasin Manurangsi
Google Research
pasin@google.com

Krzysztof Sornat
AGH University, Poland
sornat@agh.edu.pl

Abstract

We present a polynomial-time algorithm for computing an optimal committee of size k under any given Thiele voting rule for elections on the Voter Interval domain (i.e., when voters can be ordered so that each candidate is approved by a consecutive block of voters). Our result extends to the Generalized Thiele rule, in which each voter has an individual weight (scoring) sequence. This resolves a 10-year-old open problem that was originally posed for Proportional Approval Voting and later extended to every Thiele rule (Elkind and Lackner, IJCAI 2015; Peters, AAAI 2018).

Our main technical ingredient is a new structural result—a concavity theorem for Voter Interval profiles. It shows that, given two solutions of different sizes, one can construct a solution of any intermediate size whose score is at least the linear interpolation of the two scores. As a consequence, on Voter Interval profiles, the optimal total Thiele score is a concave function of the committee size. We exploit this concavity within an optimization framework based on a Lagrangian relaxation of a natural integer linear program formulation, obtained by moving the cardinality constraint into the objective. On Voter Interval profiles, the resulting constraint matrix is totally unimodular, so it can be solved in polynomial time.

Our main algorithm and its proof were obtained via human–AI collaboration. In particular, a slightly simplified version of the main structural theorem used by the algorithm was obtained in a single call to Gemini Deep Think.

1 Introduction

Approval-based committee (ABC) elections are an important topic in computational social choice literature [LS23] having applications beyond committee selection, e.g., in recommendation systems [SFL16] and blockchain [BBC+24]. In ABC elections, voters cast votes in a form of approval ballots, i.e., every voter indicates a subset of candidates. An ABC voting rule takes a collection of approval ballots (an approval profile) and outputs a committee of a given size k.

One of a general class of ABC voting rules are Thiele rules [Thi95] which contains, as special cases, such basic voting rules as: Multiwinner Approval Voting (AV), Proportional Approval Voting (PAV) [AGG+15] and Approval Voting Chamberlin-Courant (CC) [CC83], which covers different properties from efficiency via proportionality to excellence [LS21]. A Thiele voting rule is parameterized by a non-increasing infinite sequence of weights $(w_1, w_2, \dots)$. The partial sum $w_1 + w_2 + \cdots + w_R$ models satisfaction of a voter (score of a voter) from receiving R representatives in a committee. A w-Thiele rule selects a committee of size k that maximizes total score, i.e., sum of voters scores.

From the computational complexity point of view, selecting an optimal committee under Thiele rules is NP-hard (except for AV) [AGG+15, SFL16]. Therefore, computability of Thiele rules has been investigated under different paradigms: approximation algorithms [SFL16, BSS18, DMMS20,

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

BFF21, BFGG22], parameterized algorithms [BSU13, EL15, YW18, BFK+20, SVX22, YW23, GJS+25, LS26] and structured domains [BSU13, EL15, ELP17, CE21, SVX22, ELP25, LS26]. In particular, Thiele rules on Candidate Interval (CI) domain (i.e., when candidates can be ordered so that each voter approves consecutive candidates) are polynomial-time solvable by a proper Integer Linear Programming (ILP) formulation which has been proven to be totally unimodular (TU) [Pet18]. This result extends to dichotomous Single-Peaked on a Circle profiles (this domain generalizes approving intervals of candidates on a path into approving intervals of candidates on a circle graph) [PL20]. An analogous to CI is the Voter Interval (VI) domain which contains all approval profiles with the property that voters can be ordered so that each candidate is approved by a consecutive voters.

Despite the effort made over the years and tackling Thiele rules under different paradigms and using different techniques, the computational complexity of solving any given Thiele rule on a VI profile is already a 10-year-old open problem [EL15, ELP17, Pet18, YW18, PL20, GBSF21, LS23, YW23, ELP25, LS26]. It did not help that VI profiles are analogous to CI profiles by transposing voters with candidates. In particular, as observed by Peters [Pet18], a corresponding ILP formulation for VI profiles is not TU, in contrast to ILP formulation for CI profiles. This comes from the fact that the constraint matrix for VI has the consecutive-ones property column-wise, but the cardinality constraint corresponds to a row of ones. A concrete family of non-TU instances (with unbounded determinant prohibiting TU property) has been given in [LS26]. However, the constraint matrix not being TU does not imply hardness of the problem.

Over the time, partial progress has been established with two approaches. First, for some particular Thiele rules on VI there have been designed polynomial time algorithms. Namely, for CC [BSU13] and for all Thiele rules that are ℓ-truncated, i.e., when $w_i = 0$ for $i \geq \ell + 1$ [EL15]. Without knowing whether the problem is NP-hard, a second approach was designing FPT algorithms, e.g., algorithms parameterized by the size of an approval ballot and by the support size [EL15, LS26].

1.1 Our Contribution

We resolve this open problem by giving a polynomial-time algorithm that, for any fixed Thiele voting rule, computes an optimal size-k committee for elections on the Voter Interval domain. Our result also extends to the Generalized Thiele rule, in which each voter has an individual weight (scoring) sequence.

Our main technical ingredient is a novel structural result—a concavity theorem for VI profiles (see Section 3): given any two committees S, S' (not necessarily optimal) of sizes $k_1 < k_2$, we can construct, for any $k \in [k_1, k_2]$, a size-k committee contained in $S \cup S'$ whose total score is at least the corresponding linear interpolation of the scores of S and S'. By recursively applying this construction, we can obtain committees of all intermediate sizes in polynomial time. In particular, it implies that the optimal total score is a concave function of the committee size on VI profiles.

To compute an optimal size-k committee (see Section 4), we use a Lagrangian relaxation of a natural ILP formulation, moving the cardinality constraint into the objective. On VI profiles, the remaining constraint matrix becomes TU, so for any Lagrange multiplier we can compute an integral optimum in polynomial time. We then perform a binary search over the multiplier values to obtain solutions bracketing size k—either it finds a solution of size exactly k or it finds two committees: one of size at most k and one of size at least k. In the latter case, using the concavity theorem, we convert such bracketing solutions into an optimal committee of size exactly k.

Finally, we remark that key ideas for our main proofs are obtained via human–AI collaboration through iterative interactions with Gemini. In particular, both the main idea of using Lagrangian relaxation and the main structural result (concavity theorem) are generated by Gemini—which later were adapted, fully verified and slightly generalized by the authors. We view this as a concrete case study of AI-assisted research in theoretical computer science open problems. A detailed account of our interaction with the model is given in Section 5.

The proofs of our main technical results are presented in the main text. Statements whose proofs are deferred to the appendix are marked by a linked asterisk $(*)$ in their headings.

1.2 Independent Work

The first version of our paper appeared on arXiv on April 8, 2026 [MS26]. On May 6, 2026, a paper independently developed by Avramidis, Lassota, Schmidt-Kraepelin, and Vetta [ALSKV26a, ALSKV26b] appeared on arXiv. They also obtained a polynomial-time algorithm for Thiele rules on Voter Interval profiles, but using a different technique. Their approach exploits the candidate domination relation, previously studied in [DBW+25, LS26]. In particular, it was shown in [LS26] that an optimal committee can be chosen to be non-dominated using a domination-based exchange argument. Avramidis et al. [ALSKV26a] extend this argument to fractional LP solutions, establishing the existence of an integral optimum of the standard LP relaxation. Their algorithm further extends to the more general VCI and LC domains. A detailed comparison of the two approaches is provided in Appendix A.

2 Preliminaries

We use the notation $[n] = \{1, 2, \dots, n\}$. Let $C = \{c_1, \dots, c_m\}$ be a set of candidates and $V = \{v_1, \dots, v_n\}$ be a set of voters. $A_v \subseteq C$ denotes an *approval ballot* of voter $v \in V$. A collection of approval ballots $A = (A_v)_{v \in V}$ is called an *approval profile*. Any subset $W \subseteq C$ is called a *committee*. The *set of supporters* of $c \in C$ is denoted by $V_c \coloneqq \{v \in V : c \in A_v\}$. An *approval-based committee* (ABC) election is usually defined as a tuple $E = (C, V, A, k)$, where $k \in \mathbb{N}$ is the required committee size. In this paper, however, we represent an election simply by a pair (A, k), since from (A, k) we can infer the set of voters and the set of candidates $C = \bigcup_{v \in V} A_v$, i.e., the candidates that receive at least one approval. For the computational problems studied in this paper, candidates with zero approvals are relevant only in the trivial case $\left|\bigcup_{v \in V} A_v\right| < k$. An *ABC voting rule* is a function that maps an election to a set of *winning committees* of size k.

In this paper, we focus on w-Thiele voting rules, which is a class of ABC voting rules that are parameterized by an infinite non-increasing sequence $w = (w_1, w_2, \dots)$, called a *Thiele sequence*. Winning committees under w-Thiele rules are those that achieve the maximum *total score*:

$$\max_{W \subseteq C, |W| = k} \mathrm{score}_w(W), \quad \text{where } \mathrm{score}_w(W) \coloneqq \sum_{v \in V} \sum_{i=1}^{|A_v \cap W|} w_i.$$

In the corresponding computational optimization problem w-Thiele one has to find a winning committee (of a given size). By w-Thiele(A, k) we denote the maximum total score achieved by a winning committee of size k on a given instance (A, k) and we use a shortcut Thiele(k) if A and w are clear from the context.

As is standard in the literature, we assume w.l.o.g. that $w_1 = 1$ as scaling the values of w does not change the set of optimal solutions. Multiwinner Approval Voting (AV) is a Thiele rule with $w = (1, 1, \dots)$. Proportional Approval Voting (PAV) is a Thiele rule with $w_i = 1/i$. Approval Voting Chamberlin-Courant (CC) is a Thiele rule with $w = (1, 0, 0, \dots)$.

Our main results hold for a more general problem than w-Thiele and, for clarity, we use a separate name: the Generalized Thiele problem generalizes w-Thiele by allowing each voter to use a different Thiele sequence [SVX22]: Generalized Thiele takes as input an approval election (with n voters) and additionally a collection of n Thiele sequences $w \colon V \times \mathbb{N} \to [0, 1]$, where $(w_i^v)_{i \in \mathbb{N}}$ is a Thiele sequence of voter $v \in V$ (values of weights being in $[0, 1]$ are due to normalization; note that we do not require w_1^v to be 1 for every $v \in V$). For generality, we assume that all weights $(w_i^v)_{i \in [m], v \in V}$ are given as rational numbers, with numerators and denominators encoded in binary. Thiele(A, k, w) (and Thiele(k) if A and w are clear from the context) is the maximum total score achieved by a winning committee of size k on a given instance (A, k, w), i.e.,

$$\max_{W \subseteq C, |W| = k} \mathrm{score}_w(W), \quad \text{where } \mathrm{score}_w(W) \coloneqq \sum_{v \in V} \sum_{i=1}^{|A_v \cap W|} w_i^v.$$

An approval profile is *Voter Interval* (VI) if there exists an ordering of voters such that every candidate is approved by an interval of voters in this ordering [EL15, ELP17, ELP25]. Analogously, an approval profile is *Candidate Interval* (CI) if there exists an ordering of candidates such that every voter approves an interval of candidates in this ordering. If a profile is VI (or CI), the respective ordering can be found in linear time [EL15].

Both VI and CI profiles can be interpreted as a family[1] of intervals over V and C respectively. We define a few types of families of intervals which are used in this paper; a family U of intervals over $(1, 2, \ldots, n)$ is:

- *inclusion-free* if for every $I, I' \in U$ we have $I \not\subseteq I'$ and $I' \not\subseteq I$.
- *dominancy-free* if for every $I, I' \in U$ such that $I \subseteq I'$ we have $\min(I) = \min(I')$ and $\max(I) = \max(I')$, i.e., $I = I'$ [DBW+25, LS26].
- *endpoint-nested* if for every $I, I' \in U$ such that $I \subseteq I'$ we have $\min(I) = \min(I')$ or $\max(I) = \max(I')$.

Inclusion-freeness and dominancy-freeness can be defined on any family of sets (not necessarily intervals). Every family of sets that is inclusion-free is also dominancy-free. When considering a family of intervals, by the definitions of these properties, additionally we have that dominancy-freeness implies endpoint-nestedness. Moreover, all these implications are strict.

A matrix $M \in \{-1, 0, 1\}^{p \times q}$ is *totally unimodular* (TU) if every square submatrix of M has determinant -1, 0 or 1. A matrix $M \in \{0, 1\}^{p \times q}$ has the *consecutive-ones property* (for columns) if its rows can be permuted so that, in every column, the entries equal to 1 appear in a single consecutive block (possibly empty). Note that a VI approval profile interpreted as a binary matrix from $\{0, 1\}^{V \times C}$ has the consecutive-ones property. It is a well known fact, that a binary matrix with the consecutive-ones property is TU [Sch99, p. 279]. Moreover, every integer linear program with a TU constraint matrix has integral basic solutions which implies that a linear programming solver can find an integral optimal solution, i.e., we can solve the ILP with TU constraint matrix in polynomial time. For more on theory of linear programming we defer to [Sch99].

In the remainder of this section we make a few observations. First we give an equivalent definition of endpoint-nestedness.

Lemma 1 (∗)**.** *A family U of intervals over $(1, 2, \ldots, n)$ is endpoint-nested, if and only if, there exists an ordering $(I_1, I_2, \ldots, I_{|U|})$ of intervals from U such that: $\min(I_j) \leq \min(I_{j+1})$ and $\max(I_j) \leq \max(I_{j+1})$, for every $j \in [|U| - 1]$.*

We introduce a new structured domain of approval profiles, based on the endpoint-nestedness property.

Definition 2. *An approval profile A is* Endpoint-Nested *(EN) if there exists an ordering of voters such that the supporters of every candidate form a consecutive block and the resulting family of intervals $(V_c)_{c \in C}$ is endpoint-nested.*

It is known that a VI profile that is dominancy-free is also CI [DBW+25]. This leads to a polynomial time algorithm for GENERALIZED THIELE on VI dominancy-free profiles using the algorithm for CI profiles [Pet18, PL20]. Below we show that EN profiles, that form a strict superset of VI dominancy-free profiles, are also CI, so GENERALIZED THIELE can be solved in polynomial time on EN profiles (Corollary 4).

Theorem 3 (∗)**.** *An Endpoint-Nested approval profile is Candidate Interval.*

Corollary 4. GENERALIZED THIELE *on Endpoint-Nested profiles can be solved in polynomial time.*

3 Concavity of the objective function

In this section we prove that, for a given VI approval profile A, the function $k \mapsto \text{THIELE}(k)$ is concave, i.e., the maximum achievable total score is a concave function of the committee size (Corollary 9). This follows from Theorem 5 which states a more general result and we believe it is of independent interest: it shows how to combine two committees of different sizes (not necessarily optimal) into a committee of an intermediate size whose total score is at least the corresponding linear interpolation of the scores of the original committees.

[1]Throughout the paper, a family is understood as an indexed collection. In particular, different voters or candidates may correspond to identical approval ballots or supporters sets, but they are still treated as distinct objects.

Theorem 5. *Let (A, k, w) be an instance of* GENERALIZED THIELE*, where A is a Voter Interval approval profile. Given two committees $S, S' \subseteq C$ such that $|S| < |S'|$, there is a polynomial-time algorithm that computes $P = R \cup R' \cup (S \cap S')$ of size $|P| \in \left\{ \left\lfloor \frac{|S|+|S'|}{2} \right\rfloor, \left\lceil \frac{|S|+|S'|}{2} \right\rceil \right\}$ such that $R \subseteq S \setminus S', R' \subseteq S' \setminus S$ and*

$$\mathrm{score}_w(P) \geq (1-\theta) \cdot \mathrm{score}_w(S) + \theta \cdot \mathrm{score}_w(S'), \qquad \textit{where } \theta \coloneqq \tfrac{|P|-|S|}{|S'|-|S|}.$$

Proof. Fix a VI ordering $(v_1, \ldots, v_n)$ of the voters. We identify each candidate $c \in C$ with the interval $I_c \coloneqq \{i \in [n] : c \in A_{v_i}\}$. We regard these intervals as indexed objects, so distinct candidates remain distinct even if they have identical sets of supporters. Thus, we identify committees with their corresponding families of intervals. We underline that we treat families of intervals as multisets, and so, we interpret set operators such as $\cup$, $\cap$, and $\setminus$ as the corresponding multiset union/intersection/difference (i.e., with multiplicities).

For any family X of intervals over $(1, 2, \ldots, n)$, let $c_X(i) \coloneqq |\{I \in X : i \in I\}|$ denote the coverage at point $i \in [n]$. For every $i \in [n]$, we define $f_i(t) \coloneqq \sum_{j=1}^{t} w_j^{v_i}$ and $F(X) \coloneqq \sum_{i=1}^{n} f_i(c_X(i))$. Since the Thiele sequences are non-negative and non-increasing, the functions f_i are non-decreasing and concave. Moreover, for every committee X, we have $F(X) = \mathrm{score}_w(X)$.

First we prove the theorem for the case that $S \cap S' = \emptyset$. In the last part of the proof, we show how to handle the case $S \cap S' \neq \emptyset$.

The case $S \cap S' = \emptyset$. Let $U \coloneqq S \cup S'$ be the union of the given families, which has a total size of $u \coloneqq |S| + |S'|$.

Before proceeding, we outline the four steps of the proof. Our goal is to find a subset of U of intermediate size satisfying the desired score bound, but it is not clear how to construct such a subset directly. Instead, in Step 1, we transform U into a canonical family U^* in which the intervals covering each point form a consecutive block. In Step 2, this structure allows us to partition U^* into two parts with voterwise balanced coverage. In Step 3, concavity implies that at least one of these parts satisfies the desired interpolation bound. Finally, in Step 4, we transform the selected part back into a subset of the original family U without decreasing its score.

Step 1: The canonical Endpoint-Nested family U^*. Extract all u left endpoints and u right endpoints from the intervals in U, and independently sort them into non-decreasing sequences:

$$L = (l_1, l_2, \ldots, l_u) \qquad \text{such that} \quad l_1 \leq l_2 \leq \cdots \leq l_u,$$
$$R = (r_1, r_2, \ldots, r_u) \qquad \text{such that} \quad r_1 \leq r_2 \leq \cdots \leq r_u.$$

We define a *canonical family* of intervals $U^* \coloneqq \{J_j = [l_j, r_j]\}_{j=1}^{u}$ by pairing the j-th smallest left and right endpoints. Claim 6 verifies validity of definition of U^* by showing that the intervals $\{J_j = [l_j, r_j]\}_{j=1}^{u}$ are well defined.

Claim 6. *For all $j \in [u]$ it holds that $l_j \leq r_j$.*

Proof of Claim 6. For $i \in [n]$ we define $U_L(i)$ as the set of intervals in U that have left endpoint being strictly less than i, i.e., $U_L(i) = \{I \in U : \min(I) < i\}$. Analogously for right endpoints we define $U_R(i) = \{I \in U : \max(I) < i\}$. For every $i \in [n]$ we have $U_R(i) \subseteq U_L(i)$ since for every $I \in U_R(i)$ we have $i > \max(I) \geq \min(I)$, hence also $I \in U_L(i)$. Now, for any $j \in [u]$, l_j is the j-th smallest left endpoint in U. It implies that at most $j-1$ intervals in U have left endpoint being strictly less than l_j, hence $j - 1 \geq |U_L(l_j)| \geq |U_R(l_j)|$. It means that also at most $j-1$ intervals in U have right endpoint being strictly less than l_j, therefore the j-th smallest right endpoint is at least l_j, i.e., $r_j \geq l_j$. □

We note that the canonical family of intervals U^* is endpoint-nested since the definition of U^* matches the condition of Lemma 1.

Step 2: Alternating partition of U^*. Since U^* is endpoint-nested, using the same arguments as in the proof of Theorem 3, we obtain that the intervals in U^* covering i correspond exactly to a contiguous block of indices $\{q, q+1, \dots, p\}$, for some $q, p \in [u]$.

We partition U^* into two families A^* and B^* with sizes $\lfloor u/2 \rfloor$ and $\lceil u/2 \rceil$ respectively, according to index parity:

$$A^* \coloneqq \{J_2, J_4, \dots\} = \{J_j \in U^* : j \text{ is even}\},$$
$$B^* \coloneqq \{J_1, J_3, \dots\} = \{J_j \in U^* : j \text{ is odd}\}.$$

We call such (A^*, B^*) an *alternating partition of* U^*. Because any contiguous sequence of integers alternates between even and odd, the coverage made by intervals from A^* and B^* at any point $i \in [n]$ must differ by at most 1:

$$|c_{A^*}(i) - c_{B^*}(i)| \le 1. \tag{1}$$

Step 3: The alternating partition yields the interpolation bound. Since (S, S') is a partition of U and (A^*, B^*) is a partition of U^*, we have $c_S(i) + c_{S'}(i) = c_U(i)$ and $c_{A^*}(i) + c_{B^*}(i) = c_{U^*}(i)$. Furthermore, because U and U^* share the same endpoint sets L and R, their total point-wise coverage is identical: $c_{U^*}(i) = c_U(i)$. Putting these equalities together we obtain $c_S(i) + c_{S'}(i) = c_U(i) = c_{A^*}(i) + c_{B^*}(i)$, for every $i \in [n]$.

For every $i \in [n]$, the function $x \mapsto f_i(x) + f_i(c_U(i) - x)$ is concave (since f_i is concave) and symmetric around $c_U(i)/2$. Therefore, it is maximized when the split of $c_U(i)$ into $(x, c_U(i) - x)$ is as balanced as possible[2], i.e., when the two arguments in $(x, c_U(i) - x)$ differ by at most 1. Since this property holds for (A^*, B^*) (due to Equation (1)), for all $i \in [n]$ we get

$$f_i(c_{A^*}(i)) + f_i(c_{B^*}(i)) = \max_{x \in \{0,1,\dots,c_U(i)\}} \big(f_i(x) + f_i(c_U(i) - x)\big) \ge f_i(c_S(i)) + f_i(c_{S'}(i)).$$

Summing the above inequality over all $i \in [n]$, we obtain

$$F(A^*) + F(B^*) \ge F(S) + F(S'). \tag{2}$$

Let $\theta_{A^*} \coloneqq \frac{|A^*|-|S|}{|S'|-|S|}$ and $\theta_{B^*} \coloneqq \frac{|B^*|-|S|}{|S'|-|S|}$. Since $|A^*| + |B^*| = u = |S| + |S'|$, we have

$$\theta_{A^*} + \theta_{B^*} = 1. \tag{3}$$

If $F(A^*) \ge (1 - \theta_{A^*}) \cdot F(S) + \theta_{A^*} \cdot F(S')$, then we set $P^* \coloneqq A^*$. Otherwise, $F(A^*) < (1 - \theta_{A^*}) \cdot F(S) + \theta_{A^*} \cdot F(S')$ and we obtain

$$\begin{aligned} F(B^*) &\overset{(2)}{\ge} F(S) + F(S') - F(A^*) \\ &> F(S) + F(S') - \big((1 - \theta_{A^*}) \cdot F(S) + \theta_{A^*} \cdot F(S')\big) \\ &= \theta_{A^*} \cdot F(S) + (1 - \theta_{A^*}) \cdot F(S') \\ &\overset{(3)}{=} (1 - \theta_{B^*}) \cdot F(S) + \theta_{B^*} \cdot F(S'). \end{aligned}$$

Hence, in this case we set $P^* \coloneqq B^*$. Therefore, there exists $P^* \in \{A^*, B^*\}$ satisfying the following interpolation bound:

$$F(P^*) \ge (1 - \theta_{P^*}) \cdot F(S) + \theta_{P^*} \cdot F(S'), \qquad \text{where } \theta_{P^*} \coloneqq \frac{|P^*| - |S|}{|S'| - |S|}. \tag{4}$$

Moreover, $|P^*| \in \{|A^*|, |B^*|\} = \{\lfloor u/2 \rfloor, \lceil u/2 \rceil\}$ as desired. This does not finish the proof of the theorem because $P^* \subseteq U^*$, i.e., P^* is a subset of the canonical family, but we need to construct a subset of original input intervals $U = S \cup S'$ (recall that we consider the case $S \cap S' = \emptyset$). In the next step, we will show how to transform any subset of U^* to a subset of U without decreasing its coverage at any point $i \in [n]$.

[2]This inequality is in fact a special case of Karamata's inequality [Kar32, MOA11].

Step 4: Transformation back to U. In order to construct $P \subseteq U$ based on $P^* \subseteq U^*$ we rely on the following structural lemma, which shows how to transform U^*, the canonical form of U, back into U via swaps of right endpoints. This lemma applies to any family of intervals but we state it in terms of U and its canonical form U^* for concreteness.

Lemma 7. *Any family of intervals U can be obtained from its canonical endpoint-nested form U^* by a sequence of at most $O(|U|^2)$ right-endpoint swap operations, where such an operation replaces two intervals $I_1 = [l_a, r_a]$ and $I_2 = [l_b, r_b]$ satisfying $l_a \leq l_b \leq r_a \leq r_b$ with the two intervals $I_\cup = [l_a, r_b]$ and $I_\cap = [l_b, r_a]$, that is, with their union and intersection.*

Proof of Lemma 7. We show the reverse: U can be transformed into its canonical endpoint-nested form U^* by a finite sequence of inverse right-endpoint swap operations. Consider the intervals in U ordered by their left endpoints $l_1 \leq \cdots \leq l_{|U|}$. Let $r'_1, \ldots, r'_{|U|}$ be their corresponding right endpoints. If $U \neq U^*$[3], the right endpoints are not perfectly sorted, meaning there exists an adjacent inversion: an index i such that $r'_i > r'_{i+1}$. Because $l_i \leq l_{i+1} \leq r'_{i+1} < r'_i$, the interval $[l_{i+1}, r'_{i+1}]$ is strictly contained in $[l_i, r'_i]$. An inverse right-endpoint swap operation replaces them with $[l_i, r'_{i+1}]$ and $[l_{i+1}, r'_i]$. This strictly decreases the number of inversions in the sequence of right endpoints. Since the number of inversions is at most $O(|U|^2)$, this process must terminate, which occurs precisely when the right endpoints become sorted, i.e., when the resulting family is U^*. Reversing this process provides the required sequence of right-endpoint swap operations from U^* to U. □

Let $U^* = U_0 \to U_1 \to \cdots \to U_T = U$ be the transformation sequence of right-endpoint swap operations from Lemma 7. We create a sequence of families $P_t \subseteq U_t$, initialized with $P_0 = P^*$, such that $|P_t| = |P^*|$ for every $t \in \{0, 1, \ldots, T\}$. For each transition from U_t to U_{t+1} replacing $\{I_1, I_2\} \subseteq U_t$ with $\{I_\cup, I_\cap\} \subseteq U_{t+1}$ we proceed as follows:

- **Case 1:** If both $I_1, I_2 \in P_t$, we set $P_{t+1} = (P_t \setminus \{I_1, I_2\}) \cup \{I_\cap, I_\cup\}$. The point-wise coverage contribution of both pairs is identical, i.e., $c_{\{I_1, I_2\}}(i) = c_{\{I_\cup, I_\cap\}}(i)$ for every $i \in [n]$, so the point-wise coverage of P_t and P_{t+1} is the same, hence $F(P_{t+1}) = F(P_t)$.
- **Case 2:** If neither is in P_t, we set $P_{t+1} = P_t$, so $F(P_{t+1}) = F(P_t)$.
- **Case 3:** If exactly one is in P_t (call it I_a), we set $P_{t+1} = (P_t \setminus \{I_a\}) \cup \{I_\cup\}$. Because $I_\cup$ contains I_a, the coverage at every point $i \in [n]$ does not decrease, i.e., $c_{I_\cup}(i) \geq c_{I_a}(i)$, so $F(P_{t+1}) \geq F(P_t)$.

Since $|P_T| = |P^*|$, the constructed sequence of families $P_t \subseteq U_t$ satisfies:

$$\begin{aligned} F(P_T) &\geq \cdots \geq F(P_1) \geq F(P_0) = F(P^*) \\ &\overset{(4)}{\geq} (1 - \theta_{P_T}) \cdot F(S) + \theta_{P_T} \cdot F(S'), \qquad \text{where } \theta_{P_T} \coloneqq \tfrac{|P_T| - |S|}{|S'| - |S|}. \end{aligned} \tag{5}$$

Family $P \coloneqq P_T$ corresponds to a subset of C. Since $F(P) = \mathrm{score}_w(P)$, P fulfills all constraints of the theorem statement: P satisfies the concavity inequality due to Equation (5); $|P| \in \{|A^*|, |B^*|\} = \{\lfloor u/2 \rfloor, \lceil u/2 \rceil\}$; and $P = R \cup R' \cup (S \cap S')$ with $R \coloneqq P \cap S \subseteq S \setminus S'$ and $R' \coloneqq P \cap S' \subseteq S' \setminus S$. This concludes the proof for the case $S \cap S' = \emptyset$.

The case $S \cap S' \neq \emptyset$. It remains to consider the general case. The idea is to include all elements of $S \cap S'$ in P and apply the case $S \cap S' = \emptyset$ to the sets $S \setminus (S \cap S')$ and $S' \setminus (S \cap S')$. The formal proof of this case is deferred to Appendix E.

This completes the proof of the theorem. □

For two committees S and S' given, by applying Theorem 5 recursively, in polynomial time we can construct a family of committees of all intermediate sizes $|S|, |S| + 1, \ldots, |S'|$ that satisfy the concavity property from Theorem 5. In order to construct a committee of a given size p, $|S| < p < |S'|$ it is enough to apply Theorem 5 iteratively at most $O(\log(|S'| - |S|))$ times (in the binary-search manner). We state it as a separate proposition below and prove it for completeness.

[3] Note that every interval family U has a unique canonical form U^*.

Proposition 8 (∗). *Let A, w, S and S' be as in Theorem 5. Then, for any integer k such that $|S| < k < |S'|$, there is a polynomial-time algorithm that computes a committee $P = R \cup R' \cup (S \cap S')$ of size k such that $R \subseteq S \setminus S'$, $R' \subseteq S' \setminus S$ and*

$$\mathrm{score}_w(P) \geq (1-\theta) \cdot \mathrm{score}_w(S) + \theta \cdot \mathrm{score}_w(S'), \qquad \text{where } \theta := \tfrac{|P|-|S|}{|S'|-|S|}.$$

Applying Theorem 5 to optimal committees S and S' of sizes $k-1$ and $k+1$, respectively, yields a committee P of size k such that $\mathrm{score}_w(P) \geq 0.5 \cdot (\mathrm{score}_w(S) + \mathrm{score}_w(S'))$. Since $\textsc{Thiele}(k) \geq \mathrm{score}_w(P)$, and $\textsc{Thiele}(k-1) = \mathrm{score}_w(S), \textsc{Thiele}(k+1) = \mathrm{score}_w(S')$ this establishes the concavity of $\textsc{Thiele}(k)$.

Corollary 9. *For a given Voter Interval approval profile A and a collection of Thiele sequences w, the maximum total score achieved by a winning committee of size k under the* Generalized Thiele *rule is a concave function of k, i.e., for every $k \in \{1, \ldots, m-1\}$ we have*

$$\textsc{Thiele}(k) \geq 0.5 \cdot (\textsc{Thiele}(k-1) + \textsc{Thiele}(k+1)).$$

4 Polynomial-time algorithm

In this section we prove our main result which implies a polynomial-time solvability of any Thiele voting rule, in particular PAV, on VI.

Theorem 10. *There is a polynomial-time algorithm for* Generalized Thiele *on Voter Interval profiles.*

Proof. We describe how to compute an optimal solution to Generalized Thiele instance (A, k, w), where A is VI, in polynomial time. Excluding trivial cases, we assume that $k \in \{1, 2, \ldots, m-1\}$.

Standard ILP. We use a standard Integer Linear Programming (ILP) formulation of the problem [Pet18, PL20, SVX22, LS26]:

$$\begin{aligned}
\text{maximize} \quad & \textstyle\sum_{v \in V} \sum_{i \in [k]} w_i^v \cdot x_{v,i} && && (6)\\
\text{subject to} \quad & \textstyle\sum_{c \in C} y_c = k && && (7)\\
& \textstyle\sum_{i \in [k]} x_{v,i} \leq \sum_{c \in A_v} y_c && \forall v \in V && (8)\\
& x_{v,i} \in \{0,1\} && \forall v \in V, i \in [k]\\
& y_c \in \{0,1\} && \forall c \in C,
\end{aligned}$$

where selection variable $y_c = 1$ means selecting candidate c to the committee and counting variable $x_{v,i} = 1$ means that voter v approves at least i selected committee members. Cardinality constraint (7) forces selecting exactly k candidates. The constraint (8) for $v \in V$ transfers the number of selected approved candidates $\sum_{c \in A_v} y_c$ into counting variables $x_{v,i}$ which are used in the objective function (6).

Lagrangian relaxation. We modify the ILP by removing the cardinality constraint $\sum_{c \in C} y_c = k$ (Equation (7)) and adding penalty $\lambda \cdot (k - \sum_{c \in C} y_c)$ to the objective function, for some value of Lagrange multiplier $\lambda \geq 0$. This operation (called Lagrangian relaxation of a constraint) moves a constraint into the objective by penalizing violating the constraint, where the penalty has linear dependence on the original constraint (see, e.g., [JV01, CGL+25]). Additionally, we remove the constant term $\lambda \cdot k$ from the objective. Since the cardinality constraint is removed, we also extend the range of the counting variables $x_{v,i}$ from $i \in [k]$ to $i \in [m]$, allowing committees of any size to be evaluated. We call the resulting integer linear program as LR and we use notation $\mathrm{LR}(\lambda)$ when we mean LR with a specific value of Lagrange multiplier:

$$\begin{aligned}
\text{maximize} \quad & \textstyle\sum_{v \in V} \sum_{i \in [m]} w_i^v \cdot x_{v,i} - \lambda \cdot \sum_{c \in C} y_c && && (9)\\
\text{subject to} \quad & \textstyle\sum_{i \in [m]} x_{v,i} \leq \sum_{c \in A_v} y_c && \forall v \in V\\
& x_{v,i} \in \{0,1\} && \forall v \in V, i \in [m]\\
& y_c \in \{0,1\} && \forall c \in C
\end{aligned}$$

The LR removes a problematic cardinality constraint: adding this constraint can destroy total unimodularity. On VI profiles, the remaining constraint matrix is TU (as argued in [Pet18]), and thus the LR has an integral optimum and can be found in polynomial time.

Lagrange multipliers. Corollary 9 says that the optimal score $\textsc{Thiele}(t)$ achieved by a committee of size t is a concave function on VI profiles. Therefore, every $\lambda \in [\textsc{Thiele}(k+1) - \textsc{Thiele}(k), \textsc{Thiele}(k) - \textsc{Thiele}(k-1)] \subseteq [0, n]$ is a slope of an affine function $i \mapsto \textsc{Thiele}(k) + \lambda \cdot (i-k)$ that passes through the point $(k, \textsc{Thiele}(k))$ and upper-bounds $\textsc{Thiele}(i)$ for all $i \in \{0, 1, \ldots, m\}$, i.e.,

$$\textsc{Thiele}(k) + \lambda \cdot (i-k) \geq \textsc{Thiele}(i),$$

This is equivalent to

$$\textsc{Thiele}(k) - \lambda \cdot k \geq \textsc{Thiele}(i) - \lambda \cdot i$$

which implies that the maximum value of $\mathrm{LR}(\lambda)$ objective (9), for every $\lambda \in [\textsc{Thiele}(k+1) - \textsc{Thiele}(k), \textsc{Thiele}(k) - \textsc{Thiele}(k-1)] \subseteq [0, n]$, is achieved for a solution of size k. We will search for a fixed Lagrangian multiplier from this interval, i.e., $\lambda_k := \textsc{Thiele}(k) - \textsc{Thiele}(k-1)$.

Binary search for λ_k. In order to find $\lambda_k = \textsc{Thiele}(k) - \textsc{Thiele}(k-1) \in [0, n]$ we use binary search over $\lambda \in [0, n]$. This leads us either to an optimal solution of size k or to two solutions S, S' such that $|S| < k < |S'|$. This type of Lagrange-multiplier search, leading either to a size-k solution or to two bracketing solutions, is standard, for example, Lagrangian relaxation of the cardinality constraint in the k-MEDIAN problem is equivalent to the UNCAPACITATED FACILITY LOCATION problem with uniform opening cost. In this problem one searches for multipliers that bracket k open facilities before combining the resulting solutions (see, e.g., [JV01, CGL+25]). For completeness, we provide the full binary-search argument in Appendix F. In particular, it shows that the number of binary-search iterations is at most linear in the input size, under our standard encoding in which all weights are given as rational numbers with numerators and denominators encoded in binary.

Interpolation between two solutions. If the binary search does not return a solution of size k, then it produces two solutions S, S' such that $|S| < k < |S'|$. By concavity, this implies that $\textsc{Thiele}(t)$ is linear on the whole interval of sizes between $|S|$ and $|S'|$, and in particular

$$\textsc{Thiele}(a) - \lambda_k \cdot a = \textsc{Thiele}(k) - \lambda_k \cdot k, \qquad \text{for every } a \in \{|S|, |S|+1, \ldots, |S'|\}. \tag{10}$$

Now, applying Proposition 8 to committees S and S', we can construct a committee W of size exactly k (after at most $\log(|S'| - |S|) \leq \log(m)$ calls to Theorem 5) which for $\theta := \frac{k-|S|}{|S'|-|S|}$ satisfies the following:

$$\begin{aligned}
\mathrm{score}_w(W) &\geq (1-\theta) \cdot \mathrm{score}_w(S) + \theta \cdot \mathrm{score}_w(S') = (1-\theta) \cdot \textsc{Thiele}(|S|) + \theta \cdot \textsc{Thiele}(|S'|) \\
&\overset{(10)}{=} (1-\theta) \cdot (\textsc{Thiele}(k) + \lambda_k \cdot (|S| - k)) + \theta \cdot (\textsc{Thiele}(k) + \lambda_k \cdot (|S'| - k)) \\
&= \textsc{Thiele}(k) + \lambda_k \cdot \big(|S| - k + \theta \cdot (|S'| - |S|)\big) = \textsc{Thiele}(k).
\end{aligned}$$

This inequality means that W is an optimal solution and this finishes the proof. □

5 AI assistance

In this section, we describe how Gemini (Deep Think), an AI-assistance tool [Tea25], helped in achieving the results presented in this paper. Analogous descriptions of AI-assisted research in theoretical computer science have appeared recently [WCAJ+26, Knu26]. We were already working on this particular open problem and sought AI assistance to make progress on it. Our goal was not to systematically study AI-assisted problem solving across different problems. We report two separate interactions with Gemini that contributed to our approach.

Initial interactions: Arriving at Lagrangian relaxation. As a first step, we use Gemini Pro to generate the main ideas for the algorithm. In particular, we simply prompt the model to solve the problem, using the full problem description. Originally, the model proposed an incorrect polynomial-time algorithm based on Maximum Flow. Fortunately, once we point out to the model that this algorithm is incorrect, the model immediately suggests the Lagrangian relaxation approach, which is essentially the same as the one we eventually used. However, once we ask the model to provide the proof of correctness, it provides an incorrect proof. In particular, the concavity argument is incorrect. After several attempts to prompt the model to correct this proof, the model still fails.

The sequence of interactions we have with the model—with full prompts but with responses abbreviated—is given in Appendix H.

Final step: Proof of concavity of the objective. As concavity of the objective function was a crucial missing part of the Lagrangian relaxation approach, in a second step, we reformulate the concavity property in a self-contained manner. Using this reformulation, we prompt Gemini Deep Think. In the very first response, the model gives us the correct proof of the statement, which is the basis for our main concavity theorem (Theorem 5). We note that our final theorem is slightly more general than the version we asked the model to prove. Specifically, we asked the model only to combine two solutions whose sizes differ by exactly two, without requesting a polynomial-time algorithm. Moreover, the concavity formulation assumes that the two solutions are disjoint, since their common elements might be included in the combined solution. Formally, the solution created by the model might use the same interval twice, which is not allowed in the final solution. The model did not specifically discuss this issue. Nevertheless, our final proof is essentially the same as produced by the model.

The exact user prompt and response for this second step are given in Appendix G.

6 Conclusion

In this work we gave a polynomial-time algorithm for computing an optimal size-k committee for Thiele voting rules on the Voter Interval domain, thereby resolving a 10-year-old open problem [EL15, ELP17, Pet18, YW18, PL20, GBSF21, LS23, YW23, ELP25, LS26]. We also extended the result to the Generalized Thiele rule. A key ingredient is our concavity theorem for VI profiles, which we view as a structural result of independent interest.

Indeed, Generalized Thiele can also be interpreted as a heterogeneous variant of the Concave Coverage problem, where each element has its own non-decreasing concave utility function [BFF21]. Concave Coverage has been studied in connection with applications beyond voting, including resource allocation and vehicle–target assignment [BFF21], as well as list decoding and combinatorial auctions [BFGG22]. While previous work typically considers a common concave function (possibly scaled by element weights), our algorithm allows different concave functions for different elements, provided that the selectable sets form intervals over a common ordering. Identifying natural non-voting applications satisfying this interval structure is an interesting direction for future work.

An interesting direction for future work is to obtain a purely combinatorial algorithm for Thiele rules on the Voter Interval domain (and, analogously, on the Candidate Interval domain [Pet18, PL20]), avoiding linear-programming solvers. Achieving this may require further structural insights beyond concavity, which could deepen our understanding of why these domains admit efficient exact algorithms and may be of independent interest.

Finally, our results illustrate how human–AI collaboration can help identify useful structural conjectures and guide proof development in theoretical computer science. We hope this experience will encourage further systematic study of such workflows on other open problems in theoretical computer science.

Acknowledgments and Disclosure of Funding

Krzysztof Sornat was partially supported by National Science Centre, Poland, UMO-2025/58/A/ST6/00371. For the purpose of Open Access, the author has applied a CC-BY public copyright licence to any Author Accepted Manuscript (AAM) version arising from this submission. Krzysztof Sornat was partially supported by the funds allocated by the Polish Ministry of Science and Higher Education to AGH University of Krakow. During early stages of this project, Krzysztof Sornat was supported by the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No 101002854).

The authors declare no competing interests.

References

[AGG+15] 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.

[ALSKV26a] Dimitris Avramidis, Alexandra Lassota, Ulrike Schmidt-Kraepelin, and Adrian Vetta. Computing Thiele Rules on Interval Elections and their Generalizations. In *Proceedings of the 40th Annual Conference on Neural Information Processing Systems (NeurIPS 2026)*, 2026. To appear.

[ALSKV26b] Dimitris Avramidis, Alexandra Lassota, Ulrike Schmidt-Kraepelin, and Adrian Vetta. Computing Thiele Rules on Interval Elections and their Generalizations. Technical Report arXiv:2605.03067v1, arXiv.org, May 2026. Version 1 appeared on arXiv on May 6, 2026.

[BBC+24] Niclas Boehmer, Markus Brill, Alfonso Cevallos, Jonas Gehrlein, Luis Sánchez Fernández, and Ulrike Schmidt-Kraepelin. Approval-Based Committee Voting in Practice: A Case Study of (over-)Representation in the Polkadot Blockchain. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI 2024)*, pages 9519–9527, 2024.

[BFF21] 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)*, pages 9:1–9:17, 2021.

[BFGG22] Siddharth Barman, Omar Fawzi, Suprovat Ghoshal, and Emirhan Gürpinar. Tight Approximation Bounds for Maximum Multi-Coverage. *Math. Program.*, 192(1):443–476, 2022.

[BFK+20] Robert Bredereck, Piotr Faliszewski, Andrzej Kaczmarczyk, Dusan Knop, and Rolf Niedermeier. Parameterized Algorithms for Finding a Collective Set of Items. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2020)*, pages 1838–1845, 2020.

[BSS18] 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)*, pages 26:1–26:14, 2018.

[BSU13] Nadja Betzler, Arkadii Slinko, and Johannes Uhlmann. On the Computation of Fully Proportional Representation. *J. Artif. Intell. Res.*, 47:475–519, 2013.

[CC83] 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.

[CE21] 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.

[CGL+25] Vincent Cohen-Addad, Fabrizio Grandoni, Euiwoong Lee, Chris Schwiegelshohn, and Ola Svensson. A (2+ϵ)-Approximation Algorithm for Metric k-Median. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing (STOC 2025)*, pages 615–624, 2025.

[DBW+25] Chris Dong, Martin Bullinger, Tomasz Wąs, Larry Birnbaum, and Edith Elkind. Selecting Interlacing Committees. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025)*, pages 630–638, 2025.

[DMMS20] 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.

[EL15] Edith Elkind and Martin Lackner. Structure in Dichotomous Preferences. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 2019–2025, 2015.

[ELP17] 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.

[ELP25] Edith Elkind, Martin Lackner, and Dominik Peters. Preference Restrictions in Computational Social Choice: A Survey. Technical Report arXiv:2205.09092v2, arXiv.org, May 2025.

[GBSF21] Michał Tomasz Godziszewski, Paweł Batko, Piotr Skowron, and Piotr Faliszewski. An Analysis of Approval-Based Committee Rules for 2D-Euclidean Elections. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI 2021)*, pages 5448–5455, 2021.

[GJS+25] Sushmita Gupta, Pallavi Jain, Souvik Saha, Saket Saurabh, and Anannya Upasana. More Efforts Towards Fixed-Parameter Approximability of Multiwinner Rules. In *Proceedings of the 34th International Joint Conference on Artificial Intelligence (IJCAI 2025)*, pages 3891–3899, 2025.

[JV01] Kamal Jain and Vijay V. Vazirani. Approximation Algorithms for Metric Facility Location and k-Median Problems Using the Primal-Dual Schema and Lagrangian Relaxation. *J. ACM*, 48(2):274–296, 2001.

[Kar32] Jovan Karamata. Sur une inégalité relative aux fonctions convexes. (French). *Publ. Math. Univ. Belgrade*, 1:145–148, 1932.

[Knu26] Donald E. Knuth. Claude's cycles. Manuscript, Stanford Computer Science Department, February 2026. Dated 28 February 2026; revised 16 March 2026.

[LS21] Martin Lackner and Piotr Skowron. Consistent Approval-based Multi-winner Rules. *J. Econ. Theory*, 192:105173, 2021.

[LS23] Martin Lackner and Piotr Skowron. *Multi-Winner Voting with Approval Preferences*. Springer, 2023.

[LS26] Alexandra Lassota and Krzysztof Sornat. Algorithms for Structured Elections under Thiele Voting Rules. In *Proceedings of the 40th AAAI Conference on Artificial Intelligence (AAAI 2026)*, pages 17084–17092, 2026.

[MOA11] Albert W. Marshall, Ingram Olkin, and Barry C. Arnold. *Inequalities: Theory of Majorization and Its Applications*. Springer Series in Statistics. Springer, 2 edition, 2011.

[MS26] Pasin Manurangsi and Krzysztof Sornat. Polynomial-Time Algorithm for Thiele Voting Rules with Voter Interval Preferences. Technical Report abs/2604.05953, arXiv.org, April 2026. Version 1 appeared on arXiv on April 8, 2026.

[Pet18] Dominik Peters. Single-Peakedness and Total Unimodularity: New Polynomial-Time Algorithms for Multi-Winner Elections. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI 2018)*, pages 1169–1176, 2018.

[PL20] Dominik Peters and Martin Lackner. Preferences Single-Peaked on a Circle. *J. Artif. Intell. Res.*, 68:463–502, 2020.

[PS22] Grzegorz Pierczyski and Piotr Skowron. Core-Stable Committees Under Restricted Domains. In *Proceedings of the 18th International Conference on Web and Internet Economics (WINE 2022)*, pages 311–329, 2022.

[Sch99] Alexander Schrijver. *Theory of Linear and Integer Programming*. Wiley, 1999.

[SFL16] 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.

[SVX22] 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.

[Tea25] Gemini Team. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. *CoRR*, abs/2507.06261, 2025.

[Thi95] Thorvald Thiele. Om flerfoldsvalg. In *Oversigt over det Kongelige Danske Videnskabernes Selskabs Forhandlinger (in Danish)*, pages 415–441. København: A.F. Høst, 1895.

[WCAJ+26] David P. Woodruff, Vincent Cohen-Addad, Lalit Jain, Jieming Mao, Song Zuo, MohammadHossein Bateni, Simina Branzei, Michael P. Brenner, Lin Chen, Ying Feng, Lance Fortnow, Gang Fu, Ziyi Guan, Zahra Hadizadeh, Mohammad T. Hajiaghayi, Mahdi JafariRaviz, Adel Javanmard, Karthik C. S., Ken ichi Kawarabayashi, Ravi Kumar, Silvio Lattanzi, Euiwoong Lee, Yi Li, Ioannis Panageas, Dimitris Paparas, Benjamin Przybocki, Bernardo Subercaseaux, Ola Svensson, Shayan Taherijam, Xuan Wu, Eylon Yogev, Morteza Zadimoghaddam, Samson Zhou, Yossi Matias, James Manyika, and Vahab Mirrokni. Accelerating Scientific Research with Gemini: Case Studies and Common Techniques. Technical Report arXiv:2602.03837 [cs.CL], arXiv.org, February 2026.

[YW18] Yongjie Yang and Jianxin Wang. Parameterized Complexity of Multi-Winner Determination: More Effort Towards Fixed-Parameter Tractability. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2018)*, pages 2142–2144, 2018.

[YW23] Yongjie Yang and Jianxin Wang. Parameterized Complexity of Multiwinner Determination: More Effort Towards Fixed-Parameter Tractability. *Auton. Agents Multi Agent Syst.*, 37(2):28, 2023.

A Detailed Comparison with Independent Work

We provide a more detailed comparison of our approach with the work of Avramidis, Lassota, Schmidt-Kraepelin, and Vetta [ALSKV26a, ALSKV26b], who independently obtained a polynomial-time algorithm for Thiele rules on Voter Interval profiles using a different approach.

This alternative approach exploits the domination relation between candidates: c dominates c' if the set of voters supporting c is a strict superset of the set of voters supporting c'. The domination relation was previously studied, for example, in [DBW+25, LS26]. In particular, it was shown in [LS26], using a domination-based exchange argument, that an optimal committee for a Thiele rule can always be chosen to be non-dominated, and that every optimal committee is non-dominated for strictly decreasing Thiele weights.

Avramidis et al. [ALSKV26a] extend this exchange argument to fractional solutions of the standard LP relaxation. Specifically, the algorithm iteratively shifts selection weight from dominated candidates to dominating ones, without decreasing the objective value. Candidates assigned value 1 are then fixed as winners, while those assigned value 0 are discarded. The fixed winners are then removed from the instance, and their contribution to the total score is accounted for by shifting each voter's Thiele weight sequence according to the number of fixed winners they approve, a reduction previously used in [SVX22, Theorem 12]. The remaining candidates, assigned strictly fractional LP values, form a domination-free instance, which is CI due to [DBW+25, Proposition 4.3]. This residual instance can therefore be solved in polynomial time using the algorithm of [Pet18]. The algorithm of Avramidis et al. [ALSKV26a] also implies that the standard LP relaxation admits an integral optimal solution.

This iterative rounding technique extends beyond VI to the Voter-Candidate Interval (VCI) domain, introduced under the name 1D-VCR in [GBSF21] and later studied under the name VCI [DBW+25]. The VCI domain strictly generalizes both VI and CI. The algorithm further extends to the more general Linearly Consistent (LC) domain [PS22].

Our TU-based approach does *not* directly extend to VCI profiles, since VCI approval matrices need not be TU, unlike those of VI profiles. For example, the following matrix, which also appears in [LS26, Eq. (3)], is the approval matrix of a VCI profile, as shown in [GBSF21, Example 1 in the Appendix of the extended version] (up to reordering voters and candidates), but has determinant -2:

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

In contrast, the approach presented in our paper relies on the total unimodularity of VI approval matrices and a concavity theorem for VI profiles (Theorem 5). The latter is a key ingredient of our algorithm and may also be of independent interest. It allows us to combine two committees of different sizes into a committee of any intermediate size whose total score is at least the corresponding linear interpolation of their scores.

Thus, although both approaches rely on linear programming, they use it in different ways. Avramidis et al. [ALSKV26a] work directly with the standard LP relaxation and establish the existence of an integral optimum, whereas we use a Lagrangian relaxation of the cardinality constraint to obtain a new ILP that has a TU constraint matrix, and then use our concavity theorem to recover an optimal committee of the required size.

B Proof of Lemma 1

Lemma 1. *A family U of intervals over $(1, 2, \ldots, n)$ is endpoint-nested, if and only if, there exists an ordering $(I_1, I_2, \ldots, I_{|U|})$ of intervals from U such that: $\min(I_j) \leq \min(I_{j+1})$ and $\max(I_j) \leq \max(I_{j+1})$, for every $j \in [|U| - 1]$.*

Proof. ($\Longrightarrow$) We define the ordering $\succeq$ of intervals from U as follows: for $I, I' \in U$ we set $I \succ I'$ when either: 1) $\min(I) < \min(I')$ and $\max(I) \leq \max(I')$, or 2) $\min(I) = \min(I')$ and $\max(I) < \max(I')$. Additionally, $I, I' \in U$ are in the same equivalence class with respect to $\succeq$ if $I = I'$. We note that the case $\min(I) < \min(I')$ and $\max(I) > \max(I')$ cannot occur, since it would imply $I' \subseteq I$, contradicting endpoint-nestedness of U: neither $\min(I) = \min(I')$ nor $\max(I) = \max(I')$.

Now, we show that the ordering $(I_1, I_2, \ldots, I_{|U|})$ of intervals from U defined by $\succeq$ (i.e., $I_j \succeq I_{j+1}$) has the desired property. We consider any $j \in [|U| - 1]$. From the conditions defining $\succeq$, we have that $\min(I_j) < \min(I_{j+1})$ and $\max(I_j) \leq \max(I_{j+1})$, or $\min(I_j) = \min(I_{j+1})$ and $\max(I_j) < \max(I_{j+1})$, or $\min(I_j) = \min(I_{j+1})$ and $\max(I_j) = \max(I_{j+1})$. In all cases it is true that $\min(I_j) \leq \min(I_{j+1})$ and $\max(I_j) \leq \max(I_{j+1})$, which finishes the proof of the first implication.

($\Longleftarrow$) Let $(I_1, I_2, \ldots, I_{|U|})$ be an ordering of intervals from U such that: $\min(I_j) \leq \min(I_{j+1})$ and $\max(I_j) \leq \max(I_{j+1})$, for every $j \in [|U| - 1]$. Consider any $I_j, I_{j'} \in U$ where $j < j'$. We have $\min(I_j) \leq \min(I_{j'})$ and $\max(I_j) \leq \max(I_{j'})$. Thus, if $I_j \subseteq I_{j'}$, we must have $\min(I_j) = \min(I_{j'})$. Similarly, if $I_{j'} \subseteq I_j$, then $\max(I_j) = \max(I_{j'})$. Hence, U is endpoint-nested. □

C Proof of Theorem 3

Theorem 3. *An Endpoint-Nested approval profile is Candidate Interval.*

Proof. Let A be an EN approval profile and let $(v_1, v_2, \ldots, v_n)$ be the corresponding ordering of voters. Since EN is VI and we fixed the ordering of voters, instead of $(v_1, v_2, \ldots, v_n)$ and V we will use simply voters' indices, i.e., $(1, 2, \ldots, n)$ and $[n]$ to denote the order of voters and the set of voters, respectively.

For any arbitrary voter $i \in [n]$, we have $c \in A_i$ if and only if $\min(V_c) \le i \le \max(V_c)$. Using Lemma 1 we can also order candidates into sequence $(c_1, c_2, \dots, c_m)$ such that $\min(V_{c_j}) \le \min(V_{c_{j+1}})$ and $\max(V_{c_j}) \le \max(V_{c_{j+1}})$, for every $j \in [m-1]$. It means that both left and right endpoint sequences are non-decreasing. This implies that the condition $\min(V_{c_j}) \le i$ defines a prefix of indices $\{1, \dots, p\}$, and the condition $\max(V_{c_j}) \ge i$ defines a suffix of indices $\{q, \dots, m\}$. Thus, the candidates approved by voter i correspond exactly to a contiguous block of indices $\{q, q+1, \dots, p\}$, therefore A is CI. □

D Proof of Proposition 8

Proposition 8. *Let A, w, S and S' be as in Theorem 5. Then, for any integer k such that $|S| < k < |S'|$, there is a polynomial-time algorithm that computes a committee $P = R \cup R' \cup (S \cap S')$ of size k such that $R \subseteq S \setminus S'$, $R' \subseteq S' \setminus S$ and*

$$\mathrm{score}_w(P) \ge (1-\theta) \cdot \mathrm{score}_w(S) + \theta \cdot \mathrm{score}_w(S'), \qquad \textit{where } \theta := \tfrac{|P|-|S|}{|S'|-|S|}.$$

Proof. As in the proof of Theorem 5, we identify committees with their corresponding indexed families of intervals and use the notation $F(X) = \mathrm{score}_w(X)$ for such families.

We define a recursive procedure on pairs (X, Y) of families of intervals over $(1, 2, \dots, n)$ satisfying $S \cap S' \subseteq X \cap Y$, $X \cup Y \subseteq S \cup S'$, and $|X| \le k \le |Y|$. The procedure maintains the following invariant (the inequality from the proposition statement) for both families X and Y:

$$F(Z) \ge (1-\theta_Z) \cdot F(S) + \theta_Z \cdot F(S'), \qquad \text{where } Z \in \{X, Y\} \text{ and } \theta_Z := \frac{|Z| - |S|}{|S'| - |S|}. \tag{11}$$

Initially, we call the procedure on (S, S') and the invariant holds with equality. We apply Theorem 5 to X and Y. It returns a family $M = X' \cup Y' \cup (X \cap Y)$ such that $X' \subseteq X \setminus Y, Y' \subseteq Y \setminus X$, and $|M| \in \left\{ \left\lfloor \frac{|X|+|Y|}{2} \right\rfloor, \left\lceil \frac{|X|+|Y|}{2} \right\rceil \right\}$. Moreover,

$$F(M) \ge (1-\eta) \cdot F(X) + \eta \cdot F(Y), \qquad \text{where } \eta := \frac{|M| - |X|}{|Y| - |X|}. \tag{12}$$

Since $S \cap S' \subseteq X \cap Y$, we also have $S \cap S' \subseteq M$, and clearly $M \subseteq X \cup Y \subseteq S \cup S'$. Substituting the invariant bounds for $F(X)$ and $F(Y)$ in Equation (12) yields

$$F(M) \overset{(11),(12)}{\ge} (1-\eta) \cdot \big((1-\theta_X) \cdot F(S) + \theta_X \cdot F(S')\big) + \eta \cdot \big((1-\theta_Y) \cdot F(S) + \theta_Y \cdot F(S')\big). \tag{13}$$

A direct calculation shows that $(1-\eta) \cdot \theta_X + \eta \cdot \theta_Y = \frac{|M|-|S|}{|S'|-|S|}$, and therefore

$$F(M) \overset{(13)}{\ge} (1-\theta_M) \cdot F(S) + \theta_M \cdot F(S'), \qquad \theta_M := \frac{|M| - |S|}{|S'| - |S|}.$$

Thus the invariant (11) is preserved if we replace one of the endpoints by M: if $k < |M|$, we recurse on (X, M); if $k > |M|$, we recurse on (M, Y); otherwise, if $|M| = k$, we simply return $P := M$.

In each recursive step, the interval of feasible sizes shrinks from length $|Y| - |X|$ to at most $\left\lceil \frac{|Y|-|X|}{2} \right\rceil$. Hence after at most $O(\log(|S'| - |S|))$ recursive calls we reach a family M^* of size exactly k. By construction, $S \cap S' \subseteq M^* \subseteq S \cup S'$. Thus, if we define $P := M^*$ and $R := M^* \cap (S \setminus S')$, $R' := M^* \cap (S' \setminus S)$, then $P = R \cup R' \cup (S \cap S')$, with $R \subseteq S \setminus S'$ and $R' \subseteq S' \setminus S$ as required. Finally, the invariant (11) for the family M^* implies the required interpolation bound for the committee $P := M^*$, since $F(P) = \mathrm{score}_w(P)$. This finishes the proof. □

E Proof of the case $S \cap S' \neq \emptyset$ of Theorem 5

Proof of the case $S \cap S' \neq \emptyset$. We use the interval representation and the notation f_i, $c_X(i)$, and $F(X)$ introduced in the main text part of the proof (Theorem 5).

We consider the general case when $S \cap S' \neq \emptyset$. Let $D \coloneqq S \cap S'$ and for every $i \in [n]$ we define

$$g_i(x) \coloneqq f_i(x + c_D(i)).$$

Since each f_i is non-decreasing and concave, so is each g_i. Let $S_0 \coloneqq S \setminus D$ and $S'_0 \coloneqq S' \setminus D$. Then $S_0 \cap S'_0 = \emptyset$ and $|S_0| < |S'_0|$. Applying the disjoint-case construction to S_0 and S'_0 with functions $g_1, \ldots, g_n$, we obtain a family $P_0 \subseteq S_0 \cup S'_0$ such that

$$|P_0| \in \left\{ \left\lfloor \frac{|S_0| + |S'_0|}{2} \right\rfloor, \left\lceil \frac{|S_0| + |S'_0|}{2} \right\rceil \right\} \tag{14}$$

and, for $G(X) \coloneqq \sum_{i=1}^{n} g_i(c_X(i))$ for a family X of intervals over $(1, 2, \ldots, n)$, we have

$$G(P_0) \geq (1 - \theta_{P_0}) \cdot G(S_0) + \theta_{P_0} \cdot G(S'_0), \qquad \text{where } \theta_{P_0} \coloneqq \frac{|P_0| - |S_0|}{|S'_0| - |S_0|}. \tag{15}$$

Now let $P \coloneqq P_0 \cup D = P_0 \cup (S \cap S')$, so P has the required form and size since

$$\begin{aligned} |P| = |P_0| + |D| &\overset{(14)}{\in} \left\{ \left\lfloor \frac{|S_0| + |S'_0| + 2|D|}{2} \right\rfloor, \left\lceil \frac{|S_0| + |S'_0| + 2|D|}{2} \right\rceil \right\} \\ &= \left\{ \left\lfloor \frac{|S| + |S'|}{2} \right\rfloor, \left\lceil \frac{|S| + |S'|}{2} \right\rceil \right\}. \end{aligned}$$

By the definition of $g_i(\cdot)$ and $c_X(i)$ we have

$$F(P) = F(P_0 \cup D) = \sum_{i=1}^{n} f_i(c_{P_0 \cup D}(i)) = \sum_{i=1}^{n} f_i(c_{P_0}(i) + c_D(i)) = \sum_{i=1}^{n} g_i(c_{P_0}(i)) = G(P_0) \tag{16}$$

Using the analogous chain of equalities for $S = S_0 \cup D$ and $S' = S'_0 \cup D$, we obtain

$$F(S) = G(S_0) \qquad \text{and} \qquad F(S') = G(S'_0) \tag{17}$$

This implies

$$F(P) \overset{(16)}{=} G(P_0) \overset{(15)}{\geq} (1 - \theta_{P_0}) \cdot G(S_0) + \theta_{P_0} \cdot G(S'_0) \overset{(17)}{=} (1 - \theta_{P_0}) \cdot F(S) + \theta_{P_0} \cdot F(S'). \tag{18}$$

Moreover,

$$\theta_{P_0} \overset{(15)}{=} \frac{|P_0| - |S_0|}{|S'_0| - |S_0|} = \frac{(|P_0| + |D|) - (|S_0| + |D|)}{(|S'_0| + |D|) - (|S_0| + |D|)} = \frac{|P| - |S|}{|S'| - |S|}. \tag{19}$$

Hence

$$F(P) \overset{(18),(19)}{\geq} (1 - \theta) \cdot F(S) + \theta \cdot F(S'), \qquad \text{where } \theta \coloneqq \frac{|P| - |S|}{|S'| - |S|}.$$

This completes the proof for the case $S \cap S' \neq \emptyset$. □

F Binary search for λ_k in the proof of Theorem 10

In order to find $\lambda_k = \textsc{Thiele}(k) - \textsc{Thiele}(k-1) \in [0, n]$ we use binary search over $\lambda \in [0, n]$. This leads us either to an optimal solution of size k or to two solutions S, S' such that $|S| < k < |S'|$. This type of Lagrange-multiplier search, leading either to a size-k solution or to two bracketing solutions, is standard, for example, Lagrangian relaxation of the cardinality constraint in the k-MEDIAN problem is equivalent to the UNCAPACITATED FACILITY LOCATION problem with uniform opening cost. In this problem one searches for multipliers that bracket k open facilities before combining the resulting solutions (see, e.g., [JV01, CGL+25]). For completeness, we provide the full binary-search argument below.

For a tested value λ, we solve $\mathrm{LR}(\lambda)$ (in polynomial time) and obtain an optimal committee W_λ. If $|W_\lambda| = k$, we can stop and output W_λ. Otherwise, we do *not* interpret $|W_\lambda| \neq k$ as meaning that λ is already incorrect—it may happen that $\lambda = \lambda_k$ and $\mathrm{LR}(\lambda_k)$ has optimal committees of different sizes, and the solver returns one of them. Instead, we maintain a bracketing interval $[\lambda^-, \lambda^+]$ with

the property that $\max\{T(\lambda^-)\} \geq k$ and $\min\{T(\lambda^+)\} \leq k$, where $T(\lambda)$ denotes the set of sizes of optimal committees for $\mathrm{LR}(\lambda)$. Initially, this holds for $\lambda^- = 0$ and $\lambda^+ = n$. At each step we test $\lambda = \frac{\lambda^- + \lambda^+}{2}$; if $|W_\lambda| > k$ we set $\lambda^- \leftarrow \lambda$, and if $|W_\lambda| < k$ we set $\lambda^+ \leftarrow \lambda$.

This update rule remains valid since as λ increases, larger committees are penalized more by the term $-\lambda \cdot \sum_{c \in C} y_c$, and thus both $\min\{T(\lambda)\}$ and $\max\{T(\lambda)\}$ are monotone non-increasing functions of λ. Consequently, observing an optimal solution of size strictly greater than k certifies that the considered multiplier λ is (weakly) too small, while observing an optimal solution of size strictly smaller than k certifies that the considered multiplier λ is (weakly) too large, which justifies maintaining a bracketing interval in the binary search.

We continue the binary search until the length of the search interval is below additive precision ε (as defined below), i.e., $\lambda^+ - \lambda^- \leq \varepsilon$. (Recall that if at any point we obtain an optimal committee W_λ with $|W_\lambda| = k$, we stop and output it.)

To bound the number of iterations, recall that all weights $(w_i^v)_{i \in [m], v \in V}$ are given as rational numbers, with numerators and denominators encoded in binary. Let D be a common denominator of all input weights (e.g., the least common multiple of their denominators). Then every committee total score is a multiple of $1/D$. Every breakpoint slope λ at which the set of optimal sizes $T(\lambda)$ changes is of the form $\textsc{Thiele}(t) - \textsc{Thiele}(t-1)$, for $t \in [m]$, therefore it is also a multiple of $1/D$. Hence any two distinct breakpoints differ by at least $1/D$ and it suffices to run the binary search to additive precision $\frac{1}{2D}$. In particular, letting B denote the total bit-length of the denominators in the input, we have $D \leq 2^B$ and may take $\varepsilon := 1/2^{B+1}$, which yields $O(\log(n/\varepsilon)) = O(\log n + B)$ iterations. Since the input size is $\Omega(nm + B)$, the number of binary-search iterations is at most linear in the input size.

Now, we describe the next steps of the algorithm (assuming that during the binary-search procedure a committee of size k has not been found). Let $\bar{\lambda} := (\lambda^- + \lambda^+)/2$ and solve $\mathrm{LR}(\bar{\lambda})$, obtaining a committee $W_{\bar{\lambda}}$. If $|W_{\bar{\lambda}}| = k$, we stop and output $W_{\bar{\lambda}}$.

Otherwise, since $\lambda^+ - \lambda^- \leq \varepsilon$ and ε is chosen smaller than the minimum distance between two distinct breakpoints, the interval $[\lambda^-, \lambda^+]$ contains at most one breakpoint. Moreover, by construction of the bracketing interval, the unique breakpoint in $[\lambda^-, \lambda^+]$ must be λ_k, and there is no open interval of multipliers for which an LR-optimal committee has size exactly k. Equivalently, k lies on a linear segment of $\textsc{Thiele}(\cdot)$ (i.e., $|T(\lambda_k)| > 1$).

In this case we perturb the multiplier and solve $\mathrm{LR}(\bar{\lambda} + \varepsilon)$ and $\mathrm{LR}(\bar{\lambda} - \varepsilon)$, obtaining two LR-optimal committees S, S' such that $|S| < k < |S'|$. Intuitively, this perturbation selects the two endpoints of the size-interval $T(\lambda_k)$: decreasing λ slightly weakens the size penalty and yields a larger optimal committee, while increasing λ slightly strengthens the penalty and yields a smaller optimal committee.

G Human–AI interaction log I: Proof of concavity

Below, we include the exact user prompt and response from the model that provides the proof of concavity (Theorem 5). We stress that this is a single call to Gemini Deep Think without further interaction. The response is a correct proof of the required statement.

User Prompt

Consider the following objective. We have non-decreasing functions $f_1, \ldots, f_n$ from non-negative integers to non-negative reals such that they are concave. For given a multiset of intervals $S = \{I_1, \ldots, I_k\}$, its *total value* is computed as follows: For every $i \in [n]$, let c_i denote the number of intervals containing i. Then, let the total value be $F(S) = f_1(c_1) + \cdots + f_k(c_k)$. Prove the following statement: Given two multiset of intervals S, S' of size $k-1, k+1$ respectively (where k is a positive integer). Then, we can always find a sub-multiset $S'' \subseteq S \cup S'$ of size k such that $F(S'') \geq (F(S) + F(S'))/2$.

AI Response

To prove this statement, we can use a constructive combinatorial approach. The problem asks us to find a sub-multiset $S'' \subseteq S \cup S'$ of size k whose total value is at least the average of the values of S and S'.
Let $U = S \cup S'$ be the multiset union of the two given collections of intervals. The size of U is $(k-1) + (k+1) = 2k$.

Step 1: The Canonical Uncrossed Multiset U^*. First, we extract all $2k$ left endpoints and $2k$ right endpoints from the intervals in U. We sort both multisets independently in non-decreasing order:
Left endpoints L: $l_1 \leq l_2 \leq \cdots \leq l_{2k}$ Right endpoints R: $r_1 \leq r_2 \leq \cdots \leq r_{2k}$ We define a canonical multiset of intervals $U^* = \{J_m = [l_m, r_m]\}_{m=1}^{2k}$ by pairing the m-th smallest left endpoint with the m-th smallest right endpoint.
Validity: For any m, there are exactly $m-1$ left endpoints strictly less than l_m. Thus, at most $m-1$ intervals in U can finish strictly before l_m. This implies at least $2k-(m-1)$ right endpoints in U must be $\geq l_m$. Consequently, the m-th smallest right endpoint r_m must be $\geq l_m$. Therefore, all intervals in U^* are strictly valid. Identical Total Coverage: Because U and U^* share the exact same sets of endpoints L and R, their total pointwise coverage is identical. For any point $x \in [n]$, $c_{U^*}(x) = c_U(x) = c_S(x) + c_{S'}(x)$.

Step 2: Alternating Partition of U^*. Consider the ordered sequence of intervals $J_1, J_2, \ldots, J_{2k}$ in U^*. For any arbitrary point x, an interval J_m covers x if and only if $l_m \leq x \leq r_m$. Because l_m is non-decreasing, the condition $l_m \leq x$ is satisfied by a prefix of indices $\{1, 2, \ldots, p\}$. Because r_m is non-decreasing, $r_m \geq x$ is satisfied by a suffix of indices $\{q, q+1, \ldots, 2k\}$. Therefore, the subset of intervals in U^* covering x corresponds exactly to a contiguous subsegment of indices $\{q, q+1, \ldots, p\}$.
We partition U^* into two sub-multisets of size k by alternating the sorted indices: $A^* = \{J_1, J_3, J_5, \ldots, J_{2k-1}\}$, $B^* = \{J_2, J_4, J_6, \ldots, J_{2k}\}$. Because any contiguous block of integers is split as evenly as mathematically possible into evens and odds, the number of covering intervals at point x distributed to A^* and B^* will differ by at most 1:

$$|c_{A^*}(x) - c_{B^*}(x)| \leq 1$$

Step 3: Majorization and Lower Bound. For every point x, the integers $c_{A^*}(x)$ and $c_{B^*}(x)$ sum to $c_S(x) + c_{S'}(x)$ and differ by at most 1. This means they form the most balanced possible pair of integers for that sum.
By the Hardy-Littlewood-Pólya majorization inequality, a more balanced distribution maximizes the sum of any concave functions. Since each f_x is concave:

$$f_x(c_{A^*}(x)) + f_x(c_{B^*}(x)) \geq f_x(c_S(x)) + f_x(c_{S'}(x))$$

Summing this across all points $x \in [n]$ yields:

$$F(A^*) + F(B^*) \geq F(S) + F(S')$$

Since A^* and B^* both have a size of exactly k, the Pigeonhole Principle dictates that at least one of them (assume A^* without loss of generality) satisfies:

$$F(A^*) \geq \frac{F(S) + F(S')}{2}$$

Step 4: Transformation Back to U. We have found a size-k subset A^* from M^* that achieves the required threshold, but we need a subset S'' drawn from our original multiset U.
Any original configuration U can be reached from the canonical U^* via a finite sequence of "crossing" operations. A crossing operation takes two crossing intervals $I_1 = [l_a, r_a]$ and $I_2 = [l_b, r_b]$ (where $l_a \leq l_b \leq r_a \leq r_b$) and replaces them with their union $I_\cup = [l_a, r_b]$ and intersection $I_\cap = [l_b, r_a]$. (This is equivalent to reversing the standard "uncrossing" algorithm, meaning it takes adjacent inversions of the right-endpoint permutation and sorts them back into the configuration found in U).
Let $U^* = M_0 \to M_1 \to \cdots \to M_N = U$ be this sequence. We can maintain a sub-multiset $Z_i \subseteq M_i$ of size k at every step starting with $Z_0 = A^*$:

If Z_i contains both I_1, I_2, we place both $I_\cup, I_\cap$ into Z_{i+1}. The sum of their point-wise coverages is perfectly identical since $1_{I_1} + 1_{I_2} = 1_{I_\cup} + 1_{I_\cap}$. If Z_i contains neither, we put neither into Z_{i+1}. If Z_i contains exactly one of $\{I_1, I_2\}$, we strictly place the union $I_\cup$ into Z_{i+1}. Because the union strictly covers the spans of both I_1 and I_2, the coverage at any point either strictly increases or remains the same. By induction, we arrive at a size-k sub-multiset $Z_N \subseteq U$. Because coverage never decreases at any step and the given functions f_x are non-decreasing, evaluations only increase or stay constant:

$$F(Z_N) \geq F(A^*) \geq \frac{F(S) + F(S')}{2}$$

Setting $S'' = Z_N$ fulfills all constraints, concluding the proof.

H Human–AI interaction log II: Arriving at Lagrangian relaxation

In this section, we include the sequence of interactions with Gemini Pro model that lead to the formulation of the Lagrangian relaxation approach together with the concavity theorem. At the beginning, the model suggested a different (incorrect) algorithm using Maximum Flow, but after pointing out that its algorithm is incorrect, the model converges to the Lagrangian relaxation approach. However, as noted earlier, the proof given by the model is incorrect. While we have attempted to make the model corrects its mistake, the proof remains wrong until the very end.

Since the model's responses are quite long, we abbreviate part of the responses (that we believe are less relevant)—this is denoted by "[...] (ABBREVIATED)" in the responses below.

We also note that there is in fact an error in the problem formulation in the very first user prompt where $\sum_{j\in[m]} \sum_{i\in[w_{c(j)}]} w_i$ should instead be $\sum_{j\in[m]} \sum_{i\in[c(j)]} w_i$. This is a genuine mistake on our part. Despite this typo, the model immediately starts working on the correct version of the problem.

User Prompt

Consider the following problem: We have intervals $[l_1, r_1], \ldots, [l_n, r_n]$ where l_i, r_i are integers in $\{0, \ldots, m\}$. Furthermore, we also have scores $w_1 \geq \cdots \geq w_k \geq 0$. The goal is to pick k intervals that maximizes the following objective $\sum_{j\in[m]} \sum_{i\in[w_{c(j)}]} w_i$ where $c(j)$ denotes the number of times j is covered by the picked intervals. (Note that $c(j)$ is always a number from $0, \ldots, k$.)
Can you prove that this problem is NP-hard?

AI Response

Based on the problem statement provided, the problem is **not NP-hard**. In fact, it can be solved in **polynomial time**.
The problem describes a variation of the **Maximum Coverage Problem** specialized for **Intervals**, with an objective function characterized by **diminishing returns** (concavity). While the general Maximum Coverage problem (on arbitrary sets) is NP-hard, the interval structure and the concavity of the weights allow for efficient solutions using **Min-Cost Flow** or **Linear Programming**.

[...] (ABBREVIATED)

User Prompt

Ok, this is good. Can you please help write a full algorithm together with a full formal proof of correctness? You may assume that m is not too large and just create a graph on the m time steps directly without identifying the unique endpoints first.

AI Response

Here is the full algorithm and a formal proof of correctness.

Problem Restatement. We are given n intervals $[l_1, r_1], \ldots, [l_n, r_n]$ within the range $\{0, \ldots, m\}$. We have weights $w_1 \geq \cdots \geq w_k \geq 0$. We must select a subset of indices $S \subseteq \{1, \ldots, n\}$ with $|S| = k$ to maximize:

$$\sum_{j \in [m]} \sum_{i=1}^{c(j)} w_i$$

where $c(j)$ is the number of selected intervals covering point j.

Part 1: The Algorithm (Min-Cost Flow Construction). We reduce this problem to a **Min-Cost Max-Flow** problem on a directed acyclic graph.

1. Graph Construction. Let $G = (V, E)$ be a directed graph.

- **Nodes:** Create $m + 2$ nodes labeled $0, 1, \ldots, m, m + 1$.
- Node 0 acts as the **Source** (S).
- Node $m + 1$ acts as the **Sink** (T).
- The nodes $1, \ldots, m$ represent the time steps.
- **Edges:** We construct two types of edges: **Interval Edges** and **Backbone (Coverage) Edges**.

A. Interval Edges (Representing choice of intervals).
For each interval $I_u = [l_u, r_u]$ in the input (where $u \in \{1, \ldots, n\}$):

- Add a directed edge from l_u to $r_u + 1$.
- **Capacity:** 1
- **Cost:** 0
- *Intuition:* Flowing through this edge represents "picking" interval u. It skips the "costly" backbone edges between l_u and $r_u + 1$.

B. Backbone Edges (Representing coverage count).
For each position $j \in \{0, \ldots, m\}$, add k parallel edges from node j to node $j + 1$. Let's index these parallel edges by $t \in \{1, \ldots, k\}$.

- **Capacity:** 1 (per parallel edge)
- **Cost:** $-w_t$
- *Intuition:* These edges represent the benefit of covering point j. If 1 unit of flow goes through the backbone at j, we gain w_1 (cost $-w_1$). If 2 units go through, we gain $w_1 + w_2$ (cost $-w_1 - w_2$).

2. The Flow Requirement. We require a total flow of exactly k units from Source (0) to Sink ($m + 1$).

3. Execution.

1. Construct the graph as described.
2. Run a standard **Min-Cost Max-Flow** algorithm (e.g., successive shortest path using Bellman-Ford or SPFA if weights are negative, or use potentials with Dijkstra) to send exactly k units of flow from 0 to $m + 1$.
3. The minimum cost found will be negative. The maximum objective value for the original problem is $-(MinCost)$.

4. Extracting the Solution. To identify which intervals to pick:

- Inspect the **Interval Edges**. If the edge corresponding to interval $[l_u, r_u]$ carries flow 1, include interval u in your solution set S.

[...] (ABBREVIATED)

User Prompt

Your algorithm is obviously incorrect since it allows more than k intervals to be picked. Please fix this.
If you cannot fix it, try proving that it is NP-hard instead.

AI Response

You are absolutely correct. My previous construction was flawed because it failed to strictly enforce the "exactly k intervals" constraint (it allowed "stitching" disjoint intervals into a single unit of flow).
The problem is **not NP-hard**. It is solvable in **polynomial time** ($O(Poly(n, m) \cdot \log(\text{Precision}))$).
The correct approach requires handling the "exactly k" constraint using **Lagrangian Relaxation** (often called the WQS Binary Search or "Alien's Trick" in competitive programming) coupled with a **Min-Cost Circulation** algorithm.
Here is the corrected algorithm and the logic behind it.

The Core Logic: Lagrangian Relaxation. The constraint "pick exactly k intervals" is hard to model directly in flow networks because "count" is a global property. However, we can move this constraint into the objective function.
Instead of fixing k, let's assign a **cost** λ to the act of picking *any* interval. We then solve the simpler unconstrained problem:

Subproblem: Pick *any* set of intervals to maximize:

$$\left(\sum_{j} \sum_{i=1}^{c(j)} w_i \right) - \lambda \cdot (\text{number of picked intervals})$$

Intuitively:

- If λ is very high (expensive), the optimal solution picks few intervals.
- If λ is low (cheap), the optimal solution picks many intervals.
- Since the marginal gain of adding intervals is diminishing (concavity), the number of intervals picked is monotonic with respect to λ. We can **binary search** for the λ that results in picking exactly k intervals.

The function $G(k) =$ Optimal Score with k intervals is concave. This allows the binary search to find the optimal solution effectively.

The Algorithm

1. Binary Search for λ. We binary search for a penalty value $\lambda \in [-M, M]$ (where M is a sufficiently large number). In each step of the binary search, we solve the **Min-Cost Circulation** problem described below.

2. Network Construction (Min-Cost Circulation). For a fixed λ, we construct a graph to find the optimal subset of intervals.

- **Nodes:** Create nodes $0, 1, \ldots, m, m+1$ representing the timeline points.
- **Timeline Edges (Left-to-Right):** These represent the "state" of being covered.
- For each $j \in \{0, \ldots, m\}$, add edges from $j \to j+1$.
- Since the benefit is concave ($w_1 \geq w_2 \ldots$), we model this with **Convex Cost** flow.
- Add n parallel edges from $j \to j+1$:
  - Edge 1: Capacity 1, Cost $-w_1$.
  - Edge 2: Capacity 1, Cost $-w_2$.
  - $\ldots$ and so on.
- *Note:* Maximizing positive weight is equivalent to minimizing negative cost.
- **Interval Edges (Right-to-Left):** These represent the decision to "pick" an interval.
- For each available interval i given as $[l_i, r_i]$, add a directed edge from $(r_i + 1) \to l_i$.
- **Capacity:** 1 (each interval can be picked at most once).
- **Cost:** λ.

3. Solving the Circulation. Run a **Min-Cost Circulation** algorithm. This algorithm finds a flow that satisfies flow conservation at every node and minimizes total cost.

- A **circulation** decomposes into a set of cycles.
- A cycle here corresponds to:
  1. Using an interval edge $(r_i + 1) \to l_i$ (Paying λ).
  2. Flowing back through the timeline $l_i \to \cdots \to r_i + 1$ (Collecting benefits $-w_1, -w_2 \ldots$).
- The algorithm will only saturate a cycle if the benefit (sum of w's) exceeds the cost λ.

4. Update Step. Let k' be the number of interval edges saturated (carrying flow) in the optimal circulation.

- If $k' > k$: The penalty λ is too low. Increase λ.
- If $k' < k$: The penalty λ is too high. Decrease λ.
- If $k' = k$: We found the optimal set.

The final answer is:

$$\text{Objective} = -(\text{MinCost from Circulation}) + k \cdot \lambda$$

Proof of Correctness (Sketch).

1. **Total Unimodularity (TU):** The constraint matrix of the interval problem (an interval matrix) is Totally Unimodular. This ensures that the Linear Programming relaxation of the problem has an integral optimal solution.
2. **Concavity:** The function $f(x) =$ max coverage with x intervals is concave (discrete concave) due to the diminishing returns of overlaps ($w_i \geq w_{i+1}$) and the TU property.
3. **WQS Binary Search Validity:** The concavity ensures that for any k, there exists a "slope" λ such that the unconstrained optimization with penalty λ yields an optimal solution with exactly k items (or allows us to interpolate between k and $k+1$).

4. **No Stitching:** The "stitching" issue of the previous algorithm is solved by the topology. Flow travels $L \to R$ on the timeline and $R \to L$ on interval edges. To reuse the timeline flow (stitching), one would need to jump from the end of one interval to the start of another *against* the flow direction, which is impossible without using another costly interval edge or traversing the timeline (which correctly accounts for coverage).

Thus, the problem is solvable in polynomial time using Min-Cost Circulation within a binary search.

User Prompt

Can you give some more justification regarding concavity? I actually don't see how it holds.

AI Response

This is an excellent question. The claim of concavity is the cornerstone of the polynomial-time solution (specifically for the WQS binary search), so it requires a rigorous proof rather than just an appeal to "diminishing returns."
The concavity holds because the problem can be transformed into a **Min-Cost Flow problem with a parameter k as the flow demand**. A fundamental property of Min-Cost Flow networks is that the minimum cost is a **convex function** of the flow amount.

[...] (ABBREVIATED)

User Prompt

Your proof is wrong: We already discussed this but the minimum cost flow formulation you used in the proof is incorrect because it does *not* take the cardinality constraint into account. Can you just try to prove this from first principle? Namely, just try to argue, if we have two solutions for $k-1$ and $k+1$, how do we combine them to get the solution for k? Again, be completely formal and self-contained.

AI Response

You are asking for a direct, first-principles justification for why the algorithm works. The core of the correctness relies on the **concavity** of the optimal objective function $F(k)$ (the max score with k intervals).
If $F(k)$ is concave (i.e., $2F(k) \geq F(k-1) + F(k+1)$), then the local marginal gains are diminishing, and the **WQS Binary Search** (Lagrangian Relaxation) is guaranteed to find the global optimum.
Below is the formal proof of concavity. It avoids flow networks and relies on the geometric properties of intervals—specifically, the **Total Unimodularity (TU)** of the interval matrix.

Problem Setup

Let $\mathcal{I}$ be a collection of n intervals. Let A be the $m \times n$ coverage matrix, where $A_{ji} = 1$ if interval i covers point j, and 0 otherwise. Let $w : \mathbb{Z}_{\geq 0} \to \mathbb{R}$ be a function defining the score of covering a point c times:

$$w(c) = \sum_{t=1}^{\min(c,k)} w_t$$

Since $w_1 \geq w_2 \geq \ldots$, the function $w(c)$ is **concave**. The total objective for a selection vector $x \in \{0,1\}^n$ is:

$$\text{Obj}(x) = \sum_{j=1}^{m} w((Ax)_j)$$

Let $F(k)$ be the maximum objective value subject to $\sum x_i = k$.

The Claim. We prove that $F(k)$ is concave. Specifically, for any k:

$$F(k-1) + F(k+1) \leq 2F(k)$$

The Proof

Step 1: Combining Solutions. Let x be an optimal solution for $k-1$ (so $\sum x_i = k-1$). Let z be an optimal solution for $k+1$ (so $\sum z_i = k+1$).
Consider the sum vector $h = x + z$. Note that $h \in \{0,1,2\}^n$. The total number of intervals in h is $(k-1)+(k+1) = 2k$. The coverage depth at point j for this combined multiset is $C_j = (Ax)_j + (Az)_j$.
We wish to decompose h into two new integer vectors $y_1, y_2 \in \{0,1\}^n$ such that:

1. $y_1 + y_2 = h$ (They partition the multiset h).
2. $\sum y_{1,i} = k$ and $\sum y_{2,i} = k$ (Both are feasible for problem size k).
3. Their coverage profiles are "balanced" to exploit the concavity of w.

Step 2: The Splitting Lemma (via Total Unimodularity).

Lemma 11. *Let M be a matrix that is Totally Unimodular (TU). For any integer vector h, there exist integer vectors y_1, y_2 such that $y_1 + y_2 = h$ and for every row r:*

$$|(My_1)_r - (My_2)_r| \leq 1$$

Application to our problem: The constraint matrix describing our problem consists of: 1. The interval coverage matrix A (which has consecutive ones in columns). 2. The cardinality constraint (a row of all 1s). It is a known result that an interval matrix augmented with a row of 1s is **Totally Unimodular**.
Applying the lemma to $h = x + z$: We can find y_1, y_2 such that $y_1 + y_2 = h$, and:

- **Cardinality Balance:** $|\sum y_{1,i} - \sum y_{2,i}| \leq 1$. Since the total sum is $2k$, the sums must be exactly k and k. Thus, y_1, y_2 are valid feasible solutions for the size k problem.
- **Coverage Balance:** For every point j, the coverages $c_1 = (Ay_1)_j$ and $c_2 = (Ay_2)_j$ satisfy $|c_1 - c_2| \leq 1$.

Since $c_1 + c_2 = C_j$ (total coverage at j), the condition $|c_1 - c_2| \leq 1$ implies that c_1 and c_2 are the floor and ceiling of the average:

$$\{c_1, c_2\} = \left\{ \left\lfloor \frac{C_j}{2} \right\rfloor, \left\lceil \frac{C_j}{2} \right\rceil \right\}$$

Step 3: Discrete Concavity Argument. We now compare the total scores. We know that for any concave function w and any integers a, b:

$$w(a) + w(b) \leq w\left(\left\lfloor \frac{a+b}{2} \right\rfloor\right) + w\left(\left\lceil \frac{a+b}{2} \right\rceil\right)$$

(Intuitively: pushing values closer to the mean increases the sum for concave functions).
Applying this to every point j:

$$w((Ax)_j) + w((Az)_j) \leq w((Ay_1)_j) + w((Ay_2)_j)$$

Summing over all points j:

$$\text{Obj}(x) + \text{Obj}(z) \leq \text{Obj}(y_1) + \text{Obj}(y_2)$$

Since x and z were optimal for $k-1$ and $k+1$:

$$F(k-1) + F(k+1) \leq \text{Obj}(y_1) + \text{Obj}(y_2)$$

Since y_1 and y_2 are valid solutions for k, their scores cannot exceed optimal $F(k)$:

$$\text{Obj}(y_1) \leq F(k) \quad \text{and} \quad \text{Obj}(y_2) \leq F(k)$$

Combining these inequalities:

$$F(k-1) + F(k+1) \leq 2F(k)$$

Conclusion. This proves that the objective function $F(k)$ is concave. Consequently, the WQS Binary Search algorithm (Lagrangian Relaxation) is mathematically justified and will correctly find the global maximum in polynomial time.

User Prompt

You claim that an interval matrix augmented with a row of 1s is Totally Unimodular. This is clearly wrong as can be seen by the following matrix [[1, 0, 0, 1], [0, 1, 0, 1], [0, 0, 1, 1], [1, 1, 1, 1]] which has determinant 2. Can you fix this issue?

AI Response

You are absolutely correct. The matrix you provided is a valid counterexample. The "interval matrix augmented with an all-ones row" is **not** totally unimodular (TU) in general. I apologize for that error.
While the specific matrix formulation involving the cardinality row is not TU, the **Maximum Weight k-Interval Coverage problem is still solvable in polynomial time**.
The correctness of the algorithm relies on the **concavity of the objective function** $F(k)$. We can prove this concavity via a reduction to **Min-Cost Flow**, relying on the fact that the **node-arc incidence matrix** of a network is always TU (unlike the interval coverage matrix with a cardinality constraint).
Here is the corrected formal proof that resolves the issue by shifting the perspective from generic linear programming to network flow theory.

Addressing the Matrix Issue. The user correctly identified that the constraint matrix formed by interval coverage constraints plus a global cardinality constraint is not necessarily Totally Unimodular (TU). Therefore, we cannot rely on the standard LP relaxation of that specific matrix formulation.
However, we can model the problem exactly using a **Network Flow** formulation. The constraint matrix of a network flow problem (the node-arc incidence matrix) is always TU. The cardinality constraint k is handled as the **flow demand**, which acts as the right-hand side of the linear system rather than a row in the matrix itself. This preserves the integrality of the solution.

Formal Proof

Theorem 12. *Let $F(k)$ be the maximum score achievable using exactly k intervals. The function $F(k)$ is concave.*

Proof. We construct a Min-Cost Flow network $G = (V, E)$ such that the minimum cost of sending k units of flow is exactly $-F(k) + C_{constant}$. Since the minimum cost of a flow network with convex edge costs is a convex function of the flow amount, this implies $F(k)$ is concave.

Network Construction. Create a graph with nodes $0, 1, \ldots, m, m+1$.

- **Source/Sink:** We inject k units of flow at node 0 and extract k units at node $m+1$.
- **Interval Arcs (Selection):** For every interval i given as $[l_i, r_i]$, create a directed edge u_i from l_i to $r_i + 1$.
  - Capacity: 1.
  - Cost: 0.
  - Interpretation: Flow $x_i = 1$ on this arc means interval i is selected.
- **Timeline Arcs (Non-coverage):** For every $j \in \{0, \ldots, m\}$, create a directed edge v_j from j to $j + 1$.
  - Capacity: k.
  - Cost Function $\Phi_j(y)$: Depends on flow y through this arc.

Flow Conservation and Costs. Let y_j be the flow on the timeline arc $(j, j+1)$. Let S_j be the set of intervals covering point j. In the network, these correspond to interval arcs that "jump over" the timeline segment $(j, j+1)$. Consider the cut separating $\{0, \ldots, j\}$ from $\{j+1, \ldots, m+1\}$. The total flow across this cut must be k. The edges crossing this cut are exactly the timeline arc v_j and the interval arcs covering j.

$$y_j + \sum_{i \in S_j} x_i = k$$

Thus, the number of active intervals covering j is $c(j) = \sum x_i = k - y_j$. The flow y_j on the timeline represents the "missed" coverage opportunities (or the number of empty slots).
We want to maximize $\sum_{t=1}^{c(j)} w_t$. In terms of y_j, we maximize $\sum_{t=1}^{k-y_j} w_t$. This is equivalent to minimizing the negative score. We define the cost function $\Phi_j(y_j)$ for the timeline arc as:

$$\Phi_j(y_j) = -\sum_{t=1}^{k-y_j} w_t$$

Convexity of Edge Costs. To use the properties of Min-Cost Flow, the edge cost functions $\Phi_j(y)$ must be convex. Let us examine the marginal cost $\Delta\Phi_j(y) = \Phi_j(y+1) - \Phi_j(y)$.

$$\Phi_j(y) = -(w_1 + \cdots + w_{k-y})$$
$$\Phi_j(y+1) = -(w_1 + \cdots + w_{k-y-1})$$
$$\Delta\Phi_j(y) = \Phi_j(y+1) - \Phi_j(y) = w_{k-y}$$

The sequence of marginal costs for $y = 0, 1, \ldots$ is $w_k, w_{k-1}, \ldots, w_1$. Since the problem specifies non-increasing weights $w_1 \geq w_2 \geq \cdots \geq w_k$, the sequence of marginal costs is **non-decreasing**. A discrete function with non-decreasing marginal differences is **convex**.

Conclusion.

1. The constraint matrix of this problem is the node-arc incidence matrix of the graph G, which is **Totally Unimodular**. Thus, for integer demand k, there exists an integral optimal flow.
2. A standard result in network flow theory states that the optimal objective value of a Min-Cost Flow problem with convex edge costs is a **convex function** of the flow amount k.
3. Let $C(k)$ be this minimum cost. Our original maximization objective is $F(k) = -C(k)$.
4. Since $C(k)$ is convex, $F(k)$ is **concave**.

This validates the WQS Binary Search approach used in the algorithm. □

[...] (ABBREVIATED)

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims made in the abstract and introduction match the results proved later in the paper. In particular, the stated algorithmic and structural contributions are formalized in the main theorems and supported by the corresponding proofs.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are made explicit through the assumptions in the preliminaries and in the theorem statements. In particular, the paper specifies the classes of approval profiles and voting rules covered by the results, and states the provable asymptotic running time of the algorithm.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: For each theoretical result, the paper states the assumptions and provides a complete formal proof. The main contributions are proved in the main body, while only auxiliary or less central proofs are deferred to the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [N/A]

Justification: The paper does not include experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [N/A]

   Justification: The paper does not include experiments requiring code.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [N/A]

   Justification: The paper does not include experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [N/A]

   Justification: The paper does not include experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [N/A]

   Justification: The paper does not include experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics, including the anonymity requirements for submission. The paper is theoretical in nature and does not involve human subjects, personal data, or sensitive datasets.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no direct societal impact of the work performed, as the paper provides foundational theoretical results in computational social choice. Its contribution is to better understand the computational and structural properties of the considered voting rules.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks, as it does not release any data, trained models, or systems with potential for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: The paper does not use existing assets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The paper includes a dedicated section describing how LLMs contributed to the development of the proof ideas and methodology. The relevant prompts and responses are provided in the appendix. The authors independently verified the arguments and rewrote the proofs and take full responsibility for the correctness of the results.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.