# Practical approach to 2-Euclidean Preferences

Michal Dvořák
Czech Technical University in Prague
Prague, Czech Republic

Jan Pokorný
Czech Technical University in Prague
Prague, Czech Republic

Dušan Knop
Czech Technical University in Prague
Prague, Czech Republic

Martin Slávik
Czech Technical University in Prague
Prague, Czech Republic

## ABSTRACT

An election is a pair $(C, V)$ of candidates and voters. Each vote is a ranking (permutation) of the candidates. An election is $d$-Euclidean if there is an embedding of both candidates and voters into $\mathbb{R}^d$ such that voter $v$ prefers candidate $a$ over $b$ if and only if $a$ is closer to $v$ than $b$ is to $v$ in the embedding. For $d \geq 2$ the problem of deciding whether $(C, V)$ is $d$-Euclidean is $\exists\mathbb{R}$-complete. In this paper, we propose a practical approach to recognizing and refuting 2-Euclidean preferences. We design a new class of forbidden substructures that works very well on practical instances. We utilize the framework of integer linear programming (ILP) and quadratically constrained programming (QCP). We also introduce reduction rules that simplify many real-world instances significantly. Our approach beats the previous algorithm of Escoffier, Spanjaard and Tydrichová [Algorithmic Recognition of 2-Euclidean Preferences, ECAI 2023] both in number of resolved instances and the running time. In particular, we were able to lower the number of unresolved PrefLib instances from 343 to 60. Moreover, 98.7% of PrefLib instances are resolved in under 1 second using our approach.

## KEYWORDS

2-Euclidean Preferences; heuristics; ILP; QCP; Reduction Rules

**ACM Reference Format:**
Michal Dvořák, Jan Pokorný, Dušan Knop, and Martin Slávik. 2026. Practical approach to 2-Euclidean Preferences. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/TNYC5178

## 1 INTRODUCTION

The complexity of recognition problems of graph classes and geometrically inspired objects is very classical in computer science. These problems exhibit a wide range of complexities, from polynomial time to NP-complete, and even $\exists\mathbb{R}$-complete; see, e.g., [4, 9, 11, 14, 30, 32, 33, 39, 41, 42, 44, 45]. While some of these classes arise from one-dimensional objects, many are based on higher-dimensional structures.

In voting theory, preference restrictions studied over the past decades have predominantly focused on "linearly ordered" profiles, such as single-peaked or single-crossing; see Elkind, Lackner, and Peters [19]. Note that these profiles are inherently one-dimensional in a certain sense. Elkind, Lackner, and Peters also highlight that multidimensional domain restrictions present many challenging research questions. Geometric representations help visualize both the profiles and the election results [16]. Peters [43] showed that recognizing two-dimensional profiles is $\exists\mathbb{R}$-complete under the standard Euclidean metric.

To the best of our knowledge, no direct algorithmic advantage has been shown to arise from higher-dimensional profiles, despite significant efforts within the community. For example, the determination of Kemeny winner was recently shown to be NP-complete even on two-dimensional profiles [21]. However, if the Kemeny consensus is additionally restricted to be embeddable within the given profile, it can be found in polynomial time for any fixed dimension $d$ and is a 2-approximation of the unrestricted Kemeny consensus [27].

Many mathematical objects (graphs, elections, matrices, ...) allow characterization by a certain set of forbidden substructures. A classical example is the result of Kuratowski [34] characterizing planar graphs as those that do not contain subdivisions of $K_{3,3}$ or $K_5$ as subgraphs. Less known examples include the characterization of of interval graphs by infinite families of forbidden induces subgraphs by Lekkerkerker and Boland [35] or characterization of the consecutive 1's property for binary matrices by Tucker [46].

In the area of voting theory, certain properties of elections are similarly characterized by forbidden substructures. An example is the result of Ballester and Haeringer [1] characterizing the single-peaked elections in terms of two forbidden substructures. Another result of this kind is due to Bredereck, Chen and Woeginger [8] who characterized single-crossing elections by forbidding two particular patterns to occur as a subelection.

The power of forbidden substructure characterizations extends beyond theoretical classifications. Some of these characterizations can be even used to design an efficient algorithm recognizing objects with desired properties. For example the characterization of interval graphs can be directly applied algorithmically [36].

However, in some cases, the 'characterizations' are incomplete, meaning they do not contain all possible forbidden substructures that would disqualify an object from possessing the desired property. In this case, an algorithm based on these forbidden substructures is only able to detect that the object does not satisfy the given property. If no such substructure is found by the algorithm, the algorithm cannot conclude anything about the object as it may



*Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus.*  https://doi.org/10.65109/TNYC5178

contain another unidentified disqualifying substructure. This limitation occurs in particular when dealing with $d$-Euclidean elections. For $d \geq 2$, the problem of deciding whether an election is $d$-Euclidean is $\exists\mathbb{R}$-complete [43] and there cannot be finite characterization of $d$-Euclidean elections and unless $\mathsf{coNP} \subseteq \exists\mathbb{R}$ there isn't even polynomial-time recognizable characterization (see [43] for more details). Interestingly, Chen, Pruhs, and Woeginger [13] showed that already 1-Euclidean preferences cannot be characterized by finitely many forbidden substructures. On the other hand, 1-Euclidean elections are recognizable in polynomial time [15, 18, 31].

Theory suggests that it is impossible to give a good characterizing set of forbidden substructures for 2-Euclidean elections. However, many real-world instances are not 2-Euclidean (at least 91.5% of PrefLib [22]). This result was obtained by exhaustively trying all subelections and simply using the fact that 2-Euclidean elections can contain roughly $O(|C|^4)$ votes (generally, an election can contain up to $|C|!$ votes) or the characterization of 2-Euclidean profiles on at most 4 candidates [29].

However, many of the instances also fail to be 2-Euclidean due to a much simpler and quickly recognizable forbidden configuration. We demonstrate this in our paper. One such example arises from the construction of Bogomolnaïa and Laslier [7] which we refer to as the 3-8 *pattern* (see Section 2.2). Another such class of forbidden configuration arises from the convex hull of the voters (see Section 3). This phenomenon illustrates the gap between worst-case theoretical limitations and practical application, where even an incomplete characterization may suffice to solve most practical instances.

## 1.1 Related Work

Peters [43] showed that there exist $d$-Euclidean elections that inherently require exponentially many bits to even represent any of their $d$-Euclidean embeddings. Bennet and Hays [3, 28] studied what is the sufficient dimension $d$ for an election to be $d$-Euclidean. They also showed that the maximum number of voters with distinct preference ranking in a $d$-dimensional election with $m$ candidates is equal to $\sum_{k=m-d}^{m} |s(m,k)|$, where $s(m,k)$ are the Stirling numbers of the first kind. The same result was later obtained by Good and Tideman [24]. Bogomolnaïa and Laslier [7] studied lower bounds on $d$ based on the election to guarantee a $d$-Euclidean embedding. Kamiya, Takemura, and Terao [29] established the number of maximal $d$-Euclidean profiles if the number of candidates $m$ satisfies $m = d-2$ and they were able to enumerate them for $m = 4$. A simpler geometrical proof of this characterization for $m = 4$ was later given by Escoffier, Spanjaard and Tydrichová [23]. Bulteau and Chen [10] showed that any election with at most 2 voters is 2-Euclidean and for 3 voters they show that any election with at most 7 candidates is 2-Euclidean. There are also results considering similar topics in a different metric than the standard Euclidean, e.g., the $\ell_1$ and $\ell_\infty$ metrics [12, 23].

The most important previous work for us is the paper of Escoffier, Spanjaard and Tydrichová [22]. In their work, they propose an algorithm (partially) deciding whether given election is 2-Euclidean. Their algorithm consists of two phases. In the first phase, the aim is to find a no-certificate based on the maximal non-2-Euclidean elections on 4 candidates. This is achieved by testing every possible mapping to candidates in the maximal profiles. In the second phase, they aim to find an embedding using a certain randomized procedure. If no embedding is found within a specified time limit, the algorithm reports `Unknown`. Their experiments show that many real-world elections in the PrefLib dataset [40] are not 2-Euclidean (i.e., the algorithm finishes within the first phase). We will refer to this particular algorithm as the `EST` algorithm.

Apart from QCP used in the `EST` algorithm, there were previously more works using constrained-based techniques for recognizing restricted preference domains. In particular, Magiera and Faliszewski [37] showed that recognizing top-monotonic preference profiles can be reduced to 2SAT. Elkind et al. [20] provide an extensive survey of algorithms and complexity results for recognizing restricted preference domains.

## 1.2 Our contribution

We introduce new, efficiently recognizable class of forbidden substructures for 2-Euclidean elections based on the convex hull of the voters. Our experiments suggest that, in most real-world cases, using the convex hull along with the 3-8 pattern (see Section 2.2) is enough to conclude that an election is not 2-Euclidean.

Next, we utilize the framework of reduction rules that simplify the input instance and remove the 'trivial' parts. In particular, this improves the number of classified instances simply by using our reduction rules and then running previously known algorithms (e.g., the `EST` algorithm).

Finally, we study the graph-theoretical properties of the embedding graph based on a possible embedding of the election. We use integer linear program to verify that a given election cannot be 2-Euclidean by falsifying these properties. On the other side, we propose an enhancement of the QCP approach used by Escoffier, Spanjaard and Tydrichová [22] and in fact show that it can be modified to provide the embedding for previously unclassified instances that are (now known to be) 2-Euclidean. Using our methods, we were able to classify more than 82% of previously unclassified instances of PrefLib. Previously [22], there were 343 unclassified instances from PrefLib. Using our approach, we reduce this number to 60. For most previously classified instances we also improve the running time of finding the appropriate yes- or no-certificate by orders of magnitude. In particular, 98.7% of instances of PrefLib are solved with our approach under 1 second.

## 1.3 Paper Organization

In Section 2 we review basic definitions and notation used throughout the paper. Section 3 focuses on the convex hull. The reduction rules are shown in Section 4. In Section 5 we introduce the graph-theoretical framework and the ILP approach. In Section 6 we discuss the QCP approach. Lastly, we comment on implementation details and experiments in Section 7 and we conclude the paper with open questions and future research directions in Section 8. Statements where proofs or details are omitted due to space constraints are marked with ⋆. The omitted details are available in the long version [17].

# 2 PRELIMINARIES

For $i, j \in \mathbb{Z}_0^+$ we let $[i, j] = \{x \in \mathbb{Z}_0^+ \mid i \leq x \wedge x \leq j\}$ and $[j] = [1, j]$. An *election* is a pair $(C, V)$ of *candidates* and *voters*. Each voter casts a *vote* which is a permutation (or ranking) of $C$, that is, a bijection $v\colon C \to [|C|]$, or equivalently a strict linear order $\succ_v$ on $C$ given by $a \succ_v b \Leftrightarrow v(a) < v(b)$. We are not dealing with preference aggregation, hence we do not distinguish between vote and voter. We say that *voter $v$ prefers $a$ over $b$* if $a \succ_v b$. The *position* of candidate $c$ in a vote $v$, denoted by $\text{pos}_v(c)$ is defined to be $|\{d \in C \mid d \succ_v c\}|+1$. An election $(C', V')$ is a *subelection* of $(C, V)$ (or $(C', V')$ is *contained* in $(C, V)$) if there is an injective function $f\colon C' \cup V' \to C \cup V$ such that $f(C') \subseteq C$, $f(V') \subseteq V$, and $\forall a, b \in C', \forall v \in V'$ : $a \succ_v b \Leftrightarrow f(a) \succ_{f(v)} f(b)$. For election $(C, V)$ and $C' \subseteq C$ the *subelection of $(C, V)$ induced by $C'$* is the election $(C', V[C'])$, where $V[C']$ is the set of all votes in $V$ restricted to the candidates in $C'$. We say that election $(C, V)$ is 2*-Euclidean* if there exists an embedding (i.e., an injective function) $\gamma\colon C \cup V \to \mathbb{R}^2$ such that $\forall v \in V, a, b \in C$ we have $a \succ_v b \Rightarrow \ell_2(\gamma(v), \gamma(a)) < \ell_2(\gamma(v), \gamma(b))$, where $\ell_2$ is the two-dimensional euclidean distance. We refer to $\gamma$ as 2*-Euclidean* embedding of $(C, V)$. Let $\gamma$ be a 2-Euclidean embedding of $(C, V)$ and $a, b \in C$. The perpendicular bisector of the line segment with endpoints $\gamma(a)$ and $\gamma(b)$ is denoted by $\beta^{\gamma}_{a,b}$. The bisector $\beta^{\gamma}_{a,b}$ splits the plane into two open half-planes $H^{\gamma}_{a,b}$, and $H^{\gamma}_{b,a}$, where $\gamma(a) \in H^{\gamma}_{a,b}$ and $\gamma(b) \in H^{\gamma}_{b,a}$. The set of all bisectors $\mathcal{A}^{\gamma} = \{\beta^{\gamma}_{a,b} \mid a, b \in C, a \neq b\}$ induces an arrangement of lines whose cells are refered to as *regions*. We let $R^{\gamma}(v) = \bigcap_{a \succ_v b} H^{\gamma}_{a,b}$ be the region corresponding to the vote $v$. Note that $(C, V)$ is 2-euclidean if and only if there is an embedding $\gamma\colon C \to \mathbb{R}^2$ such that $R^{\gamma}(v) \neq \emptyset$ for $v \in V$. We use the same letter $\gamma$ to refer to an embedding of the candidate set $C$.

Candidates $a, b \in C$ are *consecutive* in vote $v$ if $|v(a) - v(b)| = 1$. A transposition $\tau_{a,b}\colon C \to C$ is a *consecutive swap* with respect to $v$ if $a, b$ are consecutive in $v$. For two votes $u, v$ we denote by $d^{\text{swap}}(u, v)$ their *swap distance* which is the minimum number of consecutive swaps needed to obtain $v$ from $u$. We denote the set of all possible votes over $C$ (i.e., permutations of $C$) by $\mathcal{S}_C$.

## 2.1 Nice embeddings

To prove certain properties of embeddings, it is essential to have somewhat 'well-behaved' embedding. We show that we can always assume a *nice* 2-Euclidean embedding of a 2-Euclidean election.

*Definition 2.1.* An embedding of candidates $\gamma\colon C \to \mathbb{R}^2$ is *nice* if there are no two parallel bisectors of the form $\beta^{\gamma}_{a,b}, \beta^{\gamma}_{c,d}$ for some $a, b, c, d \in C$. A 2-Euclidean embedding $\gamma$ of an election $(C, V)$ is *nice* if $\gamma$ restricted to $C$ is nice and it has no collinear triplet of voters.

Theorem 2.2 ($\star$). *Every* 2*-Euclidean election admits a nice* 2*-Euclidean embedding.*

## 2.2 Forbidden substructure on 3 voters (3-8 pattern)

It has been shown by Bulteau and Chen [10] that any election with $|C| \leq 2$ or $|C| \leq 3 \wedge |V| \leq 7$ is 2-Euclidean. Note that there exists an election with 3 voters and 8 candidates that is not 2-Euclidean. This counterexample, given by Bogomolnaïa and Laslier [7, Proposition 11], gives rise to a class of counterexamples for any fixed dimension $d \geq 2$. For $d = 2$ the pattern is as follows (see $(*)$). The set of voters is $V = \{v_1, v_2, v_3\}$ and the candidate set corresponds to the powerset of $V$. That is $C = \{c_\emptyset, c_1, c_2, c_3, c_{12}, c_{13}, c_{23}, c_{123}\}$. The preferences are as follows. The candidate $c_\emptyset$ is the 'central' candidate and is ranked fifth in each vote. For each index $i \in [3]$ we have $c_I \succ_{v_i} c_\emptyset$ if and only if $i \in I$ and the remaining preferences may be arbitrary. We will refer to this pattern as the 3-8 *pattern.*

| | | | |
|---|---|---|---|
| $v_1$ : | $c_{123} \succ c_{12} \succ c_{13} \succ c_1 \succ$ | $c_\emptyset$ | $\succ c_2 \succ c_{23} \succ c_3$ |
| $v_2$ : | $c_2 \succ c_{23} \succ c_{123} \succ c_{12} \succ$ | $c_\emptyset$ | $\succ c_{13} \succ c_1 \succ c_3$ |
| $v_3$ : | $c_{13} \succ c_3 \succ c_{123} \succ c_{23} \succ$ | $c_\emptyset$ | $\succ c_1 \succ c_2 \succ c_{12}$ |

$(*)$

Lemma 2.3 ($\star$). *Given an election $(C, V)$, we can in $O(|V|^3|C|^2)$ time decide whether $(C, V)$ contains the* 3-8 *pattern.*

# 3 CONVEX HULL

Suppose that $(C, V)$ is 2-Euclidean and let $\gamma$ be a nice 2-Euclidean embedding of $(C, V)$. Since there are finitely many voters, the convex hull $\text{conv}(\gamma(V))$ is a convex polygon. Let $\gamma(V) \cap \partial\text{conv}(\gamma(V)) = \{p_1, p_2, \ldots, p_k\}$ be the list of vertices of the polygon sorted in counterclockwise order. Note that since $\gamma$ is nice, no two consecutive sides of the polygon are parallel. Two points $p_i, p_j$ are *consecutive* if $|j - i| = 1 \vee \{i, j\} = \{1, k\}$. We refer to voters $v$ with $\gamma(v) = p_i$ as being *on the convex hull* and two voters corresponding to two consecutive points as *consecutive.*

*Definition 3.1.* Let $V' \subseteq V$ be a set of voters and $a, b \in C$ two candidates. We say that $V'$ is *controversial for $a$ over $b$* if and only if all voters in $V'$ prefer $a$ over $b$ while all voters in $V \setminus V'$ prefer $b$ over $a$. We say that $V'$ is *controversial* if it is controversial for $a$ over $b$ for some $a, b \in C$. A voter $v$ is controversial if the singleton $\{v\}$ is controversial.

Lemma 3.2 ($\star$). *Let $(C, V)$ be* 2*-Euclidean election and $\gamma$ be a nice* 2*-Euclidean embedding of $(C, V)$. If $v \in V$ is controversial, then $v$ is on the convex hull.*

Lemma 3.3 ($\star$). *Let $(C, V)$ be* 2*-Euclidean election and $\gamma$ be a nice* 2*-Euclidean embedding of $(C, V)$. Let $u, v \in V$ be two voters such that the three sets $\{u, v\}, \{u\}, \{v\}$ are controversial. Then $u$ and $v$ are consecutive on the convex hull.*

For an election $(C, V)$ we define the *controversity graph $\mathcal{CG}(C, V)$* as follows. The vertex set $V(\mathcal{CG}(C, V))$ consists of controversial voters and there is an edge $\{u, v\} \in E(\mathcal{CG}(C, V))$ between two controversial voters if and only if the set $\{u, v\} \subseteq V$ is controversial. Rephrasing in terms of the convex hull and Lemmata 3.2 and 3.3, the vertices of $\mathcal{CG}(C, V)$ are the voters that are necessarily on the convex hull and the edges correspond to consecutiveness on the convex hull for a nice 2-Euclidean embedding $\gamma$ of $(C, V)$ (if it exists). This allows us to give a simple graph-theoretical characterization of $\mathcal{CG}(C, V)$.

Theorem 3.4. *Let $(C, V)$ be* 2*-Euclidean election and let $G = \mathcal{CG}(C, V)$ be the controversity graph for $(C, V)$. Then $\Delta(G) \leq 2$ and if $G$ contains a cycle, then it is connected.*

Proof. Since $(C, V)$ is 2-Euclidean, by Theorem 2.2 there is a nice 2-Euclidean embedding of $(C, V)$. Suppose that there is a vertex $v \in V(G)$ of degree at least 3 and let $v_1, v_2, v_3$ be three distinct neighbors of $v$ in $G$. Note that this implies that the pairs

$(v, v_1), (v, v_2), (v, v_3)$ are consecutive on the convex hull. However, this is impossible. Hence $\Delta(G) \leq 2$.

For the second property suppose that $G$ contains a cycle $Y$. Then this cycle uniquely determines the convex hull $\text{conv}(\gamma(V))$ and any other voter cannot be on the boundary. Hence if $G$ was disconnected and $v$ was a voter not on $Y$, then $v$ is, by Lemma 3.2, on the convex hull. Contradiction. □

Theorem 3.4 gives us a refutation procedure for 2-Euclideanness of an election. Given $(C, V)$, construct the controversity graph $G = \mathcal{CG}(C, V)$. If $G$ does not satisfy the properties of Theorem 3.4, then $(C, V)$ is not 2-Euclidean.

## 4 REDUCING THE NUMBER OF CANDIDATES

In this section, we describe how to reduce the number of candidates in an instance while preserving 2-Euclideanness. We utilize the framework of *reduction rules*. Formally a reduction rule is an algorithm that, given an instance $(C, V)$, outputs a new instance $(C', V')$ that is 2-Euclidean if and only if $(C, V)$ is 2-Euclidean. The goal of a reduction rule is to simplify the structure of the input instance, for example decrease the number of candidates or voters. This will be the case in this section.

We emphasize that all our reduction rules are constructive. That is, given either of $(C, V)$ or $(C', V')$ our proofs also provide an algorithm how to construct an embedding of one election given the embedding of the other one, i.e., we provide the so-called *solution lifting* algorithm.

### 4.1 Removing candidates that behave similarly

We begin with a simple observation that a candidate that is always ranked the last in all votes can be safely removed. This is because this candidate can be always embedded 'far enough' from all other candidates and voters.

Reduction Rule 1. *Let $(C, V)$ be the input election. If there is a candidate $a \in C$ such that for all $b \in C \setminus \{a\}$ and all voters $v \in V$ we have $b \succ_v a$, then output the subelection induced by $C \setminus \{a\}$.*

We now generalize Reduction Rule 1 to allow for the removal of more candidates who are ranked last. For this purpose we introduce new terminology and notation. Let $(C, V)$ be an election. For $C' \subseteq C$ and $V' \subseteq V$ let $\text{pos}_{V'}(C') = \bigcup_{v \in V'} \bigcup_{c \in C'} \{\text{pos}_v(c)\}$. For a set of indices $I \subseteq [|C|]$ let $\text{pos}^{-1}(I) = \{c \in C \mid \exists v \in V : \text{pos}_v(c) \in I\}$. We say that a set of candidates $S \subseteq C$ induces a *block* if the set of positions where $S$ occurs in a vote is the same for each vote and moreover constitutes an interval. More formally, a nonempty $S \subseteq C$ is a *block* (in $(C, V)$) if there are two indices $i, j \in [|C|]$, $i \leq j$, such that $\text{pos}_V(S) = [i, j]$. We equivalently refer to a block by the discrete interval $[i, j]$ such that the set of candidates occuring between positions $i$ and $j$ is the same for all votes. Formally $[i, j]$ is a block (in $(C, V)$) if $|\text{pos}^{-1}([i, j])| = |[i, j]| = j - i + 1$. We say that $S$ is a *tail block* if $\max \text{pos}_V(S) = |C|$, that is $j = |C|$.

*Example 4.1.* If $a \in C$ is a candidate ranked the last in all votes, then $\{a\}$ is a tail block of size 1. If $|V| = 1$, then any subinterval $I \subseteq [1, |C|]$ is a block. The interval $[1, |C|]$ is always a block in any election $(C, V)$.

Let $(C, V)$ be an election and $C_1, C_2 \subseteq C$ two sets of candidates. We say that $C_1$ *copies* $C_2$ if there is a bijection $f : C_2 \to C_1$ such that for all $a, b \in C_2$ and all votes $v \in V$ we have $a \succ_v b$ if and only if $f(a) \succ_v f(b)$.

Reduction Rule 1+. *Let $(C, V)$ be the input election. If there is a tail block $S$ of size at most 3 and there is $S' \subseteq C \setminus S$ that copies $S$, then output the subelection induced by $C \setminus S$.*

Proposition 4.2 (★). *Reduction Rule 1+ is correct.*

We proceed to generalize this rule even further and introduce the notion of block decomposition. With Reduction Rule 1+ we are able to remove tail blocks of size at most 3. With block decomposition we will be able to remove such blocks even if they are not tail.

*Definition 4.3 (Block decomposition).* Let $(C, V)$ be an election and $k$ a positive integer. We say that a sequence of blocks $\mathcal{I} = ([i_1, j_1], [i_2, j_2], \ldots, [i_t, j_t])$ in $(C, V)$ forms a *$k$-block decomposition (of $(C, V)$)* if

(1) all blocks $[i_\ell, j_\ell]$ are of size at most $k$,
(2) the blocks are consecutive, that is, for all $\ell \in [t-1]$ we have $j_\ell + 1 = i_{\ell+1}$, and
(3) $[i_t, j_t]$ is a tail block.

Note that it is possible that $i_1 \neq 1$. Observe that some elections may admit multiple $k$-block decompositions for some $k$. For example the single interval $([1, |C|])$ is always a $|C|$-block decomposition for any election $(C, V)$. And if $|V| = 1$, then there are as many $k$-block decompositions of $(C, V)$ as there are ways to partition the set $[|C|]$ into nonempty intervals of size at most $k$. However, if we require that the decomposition contains maximum number of blocks, then it is unique (Lemma 4.4) and we refer to it as the *maximal $k$-block decomposition* of $(C, V)$.

Lemma 4.4 (★). *Let $(C, V)$ be an election and $k$ a positive integer. Then the $k$-block decomposition containing maximum number of blocks is unique.*

Reduction Rule 1++. *Let $(C, V)$ be the input election and let $\mathcal{I} = (I_1, I_2, \ldots, I_t)$ be its maximal 3-block decomposition. Denote $S_\ell = \text{pos}^{-1}(I_\ell)$. If there is a block $I_\ell$ such that there is $S' \subseteq C \setminus S_\ell$ that copies $S_\ell$, then output the subelection induced by $C \setminus S_\ell$.*

Proposition 4.5 (★). *Reduction Rule 1++ is correct.*

### 4.2 Removing candidates ranked next to each other

This reduction rule deals with the scenario when there are two candidates that are ranked tightly next to each other in each vote. That is, there are two candidates $b, c$ such that for every $v \in V$ we have $b \succ_v c$ and there is no $d$ and voter $v$ such that $b \succ_v d \succ_v c$. Moreover, for the reduction rule to work there must be a candidate $a$ that is ranked before $b$ in all votes. Symbolically, each vote may be described by the regular expression ?$a$?$bc$?, where ? denotes arbitrary (possibly empty) sequence of candidates (other than $a, b, c$).

Reduction Rule 2. *Let $(C, V)$ be the input election. If there are candidates $b, c \in C$ such that:*

(1) *for all votes $v$ we have $b \succ_v c$,*

(2) *for all votes $v$ there is no $d \in C \setminus \{b, c\}$ such that $b \succ_v d$ and $d \succ_v c$, and*
(3) *there exists candidate $a$ such that $a \succ_v b$ for all votes $v$,*

*then output the subelection induced by $C \setminus \{b\}$.*

Proposition 4.6 (★). *Reduction Rule 2 is correct.*

### 4.3 Limitation of generalization of reduction rules

We show that our reduction rules are optimally constrained. In particular, we prove that Reduction Rule 1+ cannot be applied even to tail blocks of size 2 when no copy of the tail block exists (Proposition 4.7). We then further explore the generalization of Reduction Rule 1+ and show that it is not possible to remove arbitrarily large tail blocks even with copies (Proposition 4.8). Finally, we show that condition 3 in Reduction Rule 2 cannot be omitted (Proposition 4.9).

Proposition 4.7 (★). *There exists an election $(C, V)$ with $|C| = 14$, $|V| = 4$ with tail block $S \subseteq C$ of size 2 and the subelection induced by $C \setminus S$ is 2-Euclidean while $(C, V)$ is not 2-Euclidean.*

Proposition 4.8 (★). *There exists an election $(C, V)$ with $|C| = 14$, $|V| = 7$ with tail block $S \subseteq C$ of size 7 and the set $S' = C \setminus S$ copies $S$ but the subelection induced by $C \setminus S$ is 2-Euclidean and $(C, V)$ is not.*

We know that we can always remove tail blocks with copies of size at most 3 by Reduction Rule 1+ and Proposition 4.8 gives an upper bound on tail block size that certainly cannot be removed safely even with copies. We conjecture that Reduction Rule 1+ cannot be extended to remove more than 3 candidates.

Proposition 4.9 (★). *There exists an election $(C, V)$ with $|C| = 8$, $|V| = 3$ with two candidates $b, c \in C$ such that:*

(1) *for all votes $v$ we have $b \succ_v c$,*
(2) *for all votes $v$ there is no $d \in C \setminus \{b, c\}$ such that $b \succ_v d$ and $d \succ_v c$,*
(3) *$(C, V)$ is not 2-Euclidean, and*
(4) *the subelection induced by $C \setminus \{b\}$ is 2-Euclidean.*

*In other words, we cannot omit condition 3 in Reduction Rule 2.*

## 5 REFUTING THE EXISTENCE OF AN EMBEDDING WITH AN ILP

The property of being 2-Euclidean imposes restriction on the election itself but also on the embedding. A trivial observation is that whenever $(C, V)$ admits a 2-Euclidean embedding $\gamma$, then $\gamma$ must have at least $|V|$ nonempty regions, one for each $v \in V$.

It is not hard to observe that for $|C| = 3$ the embedding that embeds the 3 candidates to the vertices of a non-degenerate triangle induces $6 = 3!$ distinct regions. Thus, any election with at most 3 candidates is 2-Euclidean. In general, an election with $m$ candidates can have up to $m!$ votes. However, these must correspond to some nonempty regions induced by the bisectors. Bennet and Hays [3] gave a recursive formula to compute the maximum number of nonempty regions induced by an embedding of $m$ candidates in $d$ dimensions. For $d = 2$ the upper bound is as follows.

Corollary 5.1. *Let $\gamma\colon C \to \mathbb{R}^2$ be an embedding of candidates. Let $\mathcal{R} = \{v \mid R^\gamma(v) \neq \emptyset\}$ be the set of nonempty regions of $\gamma$. Then $|\mathcal{R}| \leq \mathrm{ub}(|C|)$, where*

$$\mathrm{ub}(m) = \frac{m(3m-10)(m+1)(m-1)}{24} + m(m-1) + 1. \qquad \text{(UB)}$$

A trivial consequence of Corollary 5.1 is the following. Whenever $|V| > \mathrm{ub}(|C|)$, then $(C, V)$ is clearly not 2-Euclidean as otherwise a 2-Euclidean embedding would induce too many regions. Corollary 5.1 is one of many combinatorial properties that the election must satisfy in order to even have a chance to be 2-Euclidean. Towards designing an efficient refutation procedure, we derive additional properties that only depend on the combinatorial structure of the election itself or hold for *any* 2-Euclidean embedding of the given election.

For convenience, we rephrase the geometrical terminology of bisectors and regions into the language of plane graphs and introduce the *embedding graph* for a 2-Euclidean embedding of an election in Section 5.1. We prove important properties of the embedding graph in Section 5.1.1 and then utilize them to design an integer linear program refuting the existence of 2-Euclidean embedding in Section 5.2. To avoid degenerate or trivial cases, we will assume, without loss of generality, that $|C| \geq 4$. We already know that any election with at most three candidates is 2-Euclidean.

### 5.1 Embedding Graph

Let $\gamma\colon C \to \mathbb{R}^2$ be a nice embedding of candidates. The arrangement of lines $\mathcal{A}^\gamma$ induces a plane graph $\mathcal{P}^\gamma = (V(\mathcal{P}^\gamma), E(\mathcal{P}^\gamma))$ (the *primal graph*) as follows (refer to Figure 1 for an illustration). Let $I^\gamma = \{p \in \mathbb{R}^2 \mid \exists \beta_1, \beta_2, \in \mathcal{A}^\gamma : \beta_1 \neq \beta_2, p \in \beta_1 \cap \beta_2\}$ be the set of all pairwise intersection points of distinct bisectors. Let $B^\gamma$ be an open ball containing $I^\gamma$. We let $V(\mathcal{P}^\gamma) = I^\gamma \cup (\bigcup \mathcal{A}^\gamma \cap \partial B^\gamma)$. The arcs $E(\mathcal{P}^\gamma)$ are either the straight line segments (parts of bisectors) or circular arcs (parts of $\partial B^\gamma$) connecting two intersection points. Observe that the bounded faces of $\mathcal{P}^\gamma$ are exactly the regions $R^\gamma(v)$ (the outer regions are clipped by $B^\gamma$). Moreover the circle $\partial B^\gamma$ is partitioned by the bisectors into several circular arcs and there is a one to one correspondence between these circular arcs and the outer regions of the embedding.

The *embedding graph for $\gamma$*, denoted by $\mathcal{D}^\gamma$, is the weak dual of the plane graph $\mathcal{P}^\gamma$, i.e., $\mathcal{D}^\gamma$ has a vertex for each bounded face of $\mathcal{P}^\gamma$ and there is an edge between two faces $F_1, F_2$ for each common boundary edge of $F_1$ and $F_2$. Each bounded face of $\mathcal{P}^\gamma$ in turn corresponds to a nonempty region $R^\gamma(v)$. By a slight abuse of notation we identify vertices $v \in V(\mathcal{D}^\gamma)$ with the underlying permutation $v \in \mathcal{S}_C$ and with the nonempty region $R^\gamma(v)$. This correspondence allows us to rephrase Corollary 5.1 in terms of the embedding graph:

Corollary 5.2. *Let $(C, V)$ be a 2-Euclidean election, and $\gamma$ its 2-Euclidean embedding. Then the number of vertices of $\mathcal{D}^\gamma$ is at most $\mathrm{ub}(|C|)$.*

#### 5.1.1 *Properties of Embeddings and the Embedding Graph.*

We begin with a simple observation that the edges of the embedding graph correspond to consecutive swaps in the underlying permutations.

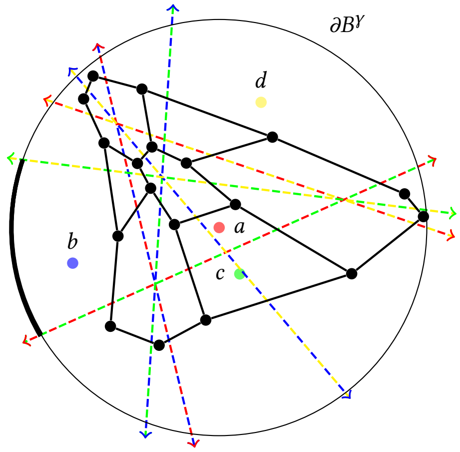


**Figure 1: The embedding graph $\mathcal{D}^{\gamma}$ (in black). The highlighted circular arc corresponds to the region $R^{\gamma}(bacd)$.**

Observation 1 (★). *In the embedding graph $\mathcal{D}^{\gamma}$, there is an edge between two vertices u and v if and only if $d^{\text{swap}}(u,v) = 1$.*

Observation 2 (★). *The embedding graph $\mathcal{D}^{\gamma}$ has no loops and multiedges.*

Let $G(\mathcal{S}_C)$ be a graph with vertex set $\mathcal{S}_C$ and two vertices share an edge if and only if they differ by a consecutive swap. It is not hard to see that $\mathcal{D}^{\gamma}$ is a subgraph of $G(\mathcal{S}_C)$. Moreover, it is a distance-preserving subgraph of $G(\mathcal{S}_C)$, i.e., for any two vertices $u, v \in V(H)$ we have $d_H(u,v) = d_G(u,v)$.

Lemma 5.3 (★). *The graph $\mathcal{D}^{\gamma}$ is distance-preserving subgraph of $G(\mathcal{S}_C)$ of minimum degree at least 2.*

Lemma 5.4 (★). *Let $(C,V)$ be a 2-Euclidean election and $\gamma$ any 2-Euclidean embedding of $(C,V)$ and let $G = G(\mathcal{S}_C)$. Let $u, v \in V$ be two votes and let $v_1, \dots, v_k \in N_G(u)$ be all the neighbors of u such that $d_G(v_i, v) = d_G(u,v) - 1$. Then there is index i such that $v_i \in V(\mathcal{D}^{\gamma})$.*

In other words, let $(C,V)$ is 2-Euclidean and $u, v \in V$ two votes and let $v_1, \dots, v_k$ be all the votes that differ from $u$ by a consecutive swap and $d^{\text{swap}}(v_i, v) = d^{\text{swap}}(u,v) - 1$. Then $R^{\gamma}(v_i) \neq \emptyset$ for some $i$.

Observation 3 (★). *Let C be a set of candidates and $\gamma$ an embedding of C. For each candidate $c \in C$ there is a vote $v \in \mathcal{S}_C$ such that $\text{pos}_v(c) = 1$ and $R^{\gamma}(v) \neq \emptyset$.*

We now turn our attention to the inner and outer regions of the embedding.

Observation 4 (★). *Let $\gamma\colon C \to \mathbb{R}^2$ be a nice embedding of a candidate set. Then there are exactly $2\binom{|C|}{2}$ nonempty outer regions.*

Lemma 5.5 (★). *Let $\gamma$ be an embedding of a candidate set C and let $v \in \mathcal{S}_C$. The following conditions are equivalent:*

(1) *The region $R^{\gamma}(v)$ is outer.*
(2) *Both $R^{\gamma}(v)$ and $R^{\gamma}(v^R)$ are nonempty.*

## 5.2 The algorithm

Let $(C,V)$ be 2-Euclidean and let $\gamma$ be its nice 2-Euclidean embedding. The properties derived in Section 5.1.1 impose constraints on both the embedding $\gamma$ and the graph $\mathcal{D}^{\gamma}$. Conversely, if can show, given an election $(C,V)$, that all the constraints cannot be satisfied, then clearly $(C,V)$ is not 2-Euclidean. For this purpose we formulate our problem as an integer linear program (ILP). The core variables of the program are binary variables $x_v$ and $\iota_v$ for each $v \in \mathcal{S}_C$ and the correspondence is as follows. If $(C,V)$ is 2-Euclidean and $\gamma$ is a nice 2-Euclidean embedding of $(C,V)$, then by setting $x_v = 1 \Leftrightarrow R^{\gamma}(v) \neq \emptyset$ and $\iota_v = 1 \Leftrightarrow (R^{\gamma}(v) \neq \emptyset \wedge R^{\gamma}(v)$ is an inner region), we obtain a feasible solution. We emphasize that the property of the ILP is meant to be only unidirectional. Not every feasible solution to the ILP should correspond to some embedding $\gamma$ with $R^{\gamma}(v) \neq \emptyset$ if and only if $x_v = 1$.

In the following paragraphs we describe how to transform the properties from Section 5.1.1 to ILP constraints.

*5.2.1 Basic constraints.* We begin with the obvious constraints:

$$x_v = 1 \qquad \forall v \in V. \tag{1}$$

Clearly for every $v \in V$ the region $R^{\gamma}(v)$ is nonempty, given that $(C,V)$ is 2-Euclidean. Next, there cannot be more than $\text{ub}(|C|)$ nonempty regions (see Corollary 5.1).

$$\sum_{v \in \mathcal{S}_C} x_v \leq \text{ub}(|C|) \tag{2}$$

Lemma 5.4 gives a constraint that can be interpreted as the formula $(x_u \wedge x_v) \Rightarrow \bigvee_{i=1}^{k} x_{v_i}$, where $v_1, \dots, v_k$ are all votes such that $d^{\text{swap}}(u, v_i) = 1$ and $d^{\text{swap}}(v_i, v) = d^{\text{swap}}(u,v) - 1$. This can be encoded as follows:

$$(1 - x_u) + (1 - x_v) + \sum_{i=1}^{k} x_{v_i} \geq 1 \qquad \forall \{u, v\} \subseteq \mathcal{S}_C. \tag{3}$$

We proceed by introducing a constraint for Observation 3.

$$\sum_{v \in \mathcal{S}_C, \text{pos}_v(c)=1} x_v \geq 1 \qquad \forall c \in C \tag{4}$$

*5.2.2 Inner and Outer regions.* Recall that the variables $\iota_v$ correspond to nonempty regions that are also inner. Hence we add the obvious constraint $\iota_v \Rightarrow x_v$ which is captured by the following linear constraint:

$$(1 - \iota_v) + x_v \geq 1 \qquad \forall v \in \mathcal{S}_C \tag{5}$$

Next constraint is due to Observation 4:

$$\sum_{v \in \mathcal{S}_C} (x_v - \iota_v) = 2\binom{|C|}{2} \tag{6}$$

Next, we apply Lemma 5.5:

$$\iota_v + \iota_{v^R} \leq 1 \qquad \forall v \in \mathcal{S}_C \tag{7}$$

By using the second part of Lemma 5.5 we can also see that if the region $R^{\gamma}(v)$ is outer, then the region $R^{\gamma}(v^R)$ is also nonempty and is outer. This fact can be captured by the formula $(x_v \wedge \neg\iota_v) \Rightarrow (x_{v^R} \wedge \neg\iota_{v^R})$ which is translated to the following linear constraints:

$$(1 - x_v) + \iota_v + x_{v^R} \geq 1 \qquad \forall v \in \mathcal{S}_C \tag{8}$$

$$(1 - x_v) + \iota_v + (1 - \iota_{v^R}) \geq 1 \qquad \forall v \in \mathcal{S}_C \tag{9}$$

Since minimum degree of $\mathcal{D}^{\gamma}$ is at least 2, we have:

$$\sum_{v_i \in N_{G(\mathcal{S}_C)}(v)} x_{v_i} \geq 2x_v + \iota_v \qquad \forall v \in \mathcal{S}_C \tag{10}$$

The description of the neighbors can be made more precise. Every outer region has exactly two neighboring regions that are also outer. For a vote $v$, consider the sum $S(v) = \sum_{v_i \in N_{G(\mathcal{S}_C)}} (x_{v_i} - \iota_{v_i})$. $S(v)$ counts the number of neighbors of $v$ that are also outer regions. Our aim is to express the following. If $x_v = 1$ and $\iota_v = 0$, then $S(v) = 2$. This can be encoded by the following:

$$S(v) \geq 2 \cdot (x_v - \iota_v) \qquad \forall v \in \mathcal{S}_C \qquad (11)$$

$$S(v) \leq 2 + \iota_v \cdot |C| \qquad \forall v \in \mathcal{S}_C \qquad (12)$$

Theorem 5.6 (★). *Let $(C, V)$ be an election. Suppose that $(C, V)$ is 2-Euclidean and let $\gamma$ be a nice 2-Euclidean embedding of $(C, V)$. Then by plugging in $x_v = 1$ if and only if $R^\gamma(v) \neq \emptyset$ and $\iota_v$ if and only if $x_v = 1$ and the region $R^\gamma(v)$ is an inner region, then all the constraints (1) up to (12) are satisfied.*

Note that the formulation of the ILP has $|C|!$ variables and may seem impractical at first sight. We propose a lazy approach applicable in practice which is described in the long version.

## 6 QCP APPROACH

In this section, we propose the Quadratically Constrained Program (QCP) approach, which arises from the definition of 2-Euclidean elections. By definition, an election $(C, V)$ is 2-Euclidean if and only if there exist points $\gamma(t) \in \mathbb{R}^2$ for each $t \in C \cup V$ such that, whenever $a \succ_v b$, we have $\ell_2(\gamma(v), \gamma(a)) < \ell_2(\gamma(v), \gamma(b))$.

The main challenge here is that QCP programs and in particular most practical QCP solvers require the inequalities to be non-strict. However, this is manageable by ensuring that the distances of voters from the bisectors are sufficiently large. This is dealt with by adding error terms into the inequalities. More details can be found in the long version.

The key improvement over the QCP approach of Escoffier, Spanjaard and Tydrichová [22] is the addition of the following constraints:

$$-x_{\max} \leq x_t \leq x_{\max} \qquad \forall t \in C \cup V$$

$$-y_{\max} \leq y_t \leq y_{\max} \qquad \forall t \in C \cup V$$

In other words, we bound the range of the coordinates of the embedding to the rectangle $2x_{\max} \times 2y_{\max}$. In practice, if the bounding box is too small, the QCP solver can quickly recognize infeasibility. In such case we keep doubling the bounding box until a solution is found.

## 7 EXPERIMENTS

We implemented[2] the algorithms `3-8` (based on Section 2.2), `Hull` (based on Section 3), `ILP` (based on Section 5), and `QCP` (based on Section 6). As the source of the experiments, we took all the complete strict order datasets (`.soc` files) from PrefLib [40] including datasets from [2, 5, 6, 25, 38]. As of October 2025 there were 7743 such instances in PrefLib. The experiments were carried out on a machine with `AMD EPYC 7742 64-Core Processor`. For each instance each heuristic (including the `EST` algorithm) was run on a single thread with time limit of 1 hour and memory limit of 32 GiB.

[1]The dataset 00062 - orderaltexpe was not present in PrefLib at the time of the paper [22] and it is also not counted towards the 212 newly resolved no-instances.

[2]The source code and related material is available at `zenodo.org/records/18541197`.

| dataset | unknown | non-2-Euclidean | 2-Euclidean |
|---|---|---|---|
| 00006 - skate | 0(-16) | 18(+14) | 2(+2) |
| 00015 - cleanweb | 6(-35) | 72(+34) | 1(+1) |
| 00032 - education | 0(-1) | 1(+1) | 0 |
| 00042 - boxing | 29(-60) | 11(+10) | 56(+50) |
| 00043 - cycling | 0(-2) | 121 | 2(+2) |
| 00044 - tabletennis | 0(-2) | 36(+2) | 0 |
| 00048 - spotifycountry | 0(-4) | 640(+3) | 2(+1) |
| 00049 - mylaps | 3(-47) | 584(+38) | 23(+9) |
| 00052 - f1seasons | 2(-6) | 61(+2) | 4(+4) |
| 00054 - weeksport | 0(-4) | 951(+4) | 0 |
| 00056 - seasonsport | 20(-106) | 3766(+104) | 193(+2) |
| 00062 - orderaltexpe[1] | 0 | 2(+2) | 0 |
| other | 0 | 1037 | 100 |
| total | 60(-283) | 7300(+212) | 383(+71) |

**Table 1: Solved instances per PrefLib dataset by using all solvers combined (including EST). The green numbers indicate the improvement over the EST algorithm alone.**

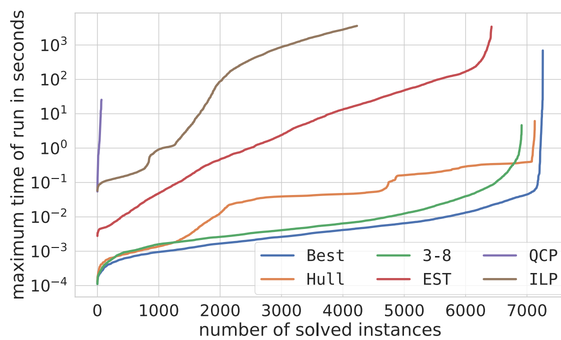


**Figure 2: The plot shows how many instances each solver solves within given time limit. Best represents ideal parallel execution, stopping all solvers once the first finishes.**

| | solved | fastest | max time (s) | median time (s) |
|---|---|---|---|---|
| `3-8` | 6914 | 5213 | 4.653348 | 0.004962 |
| `EST` | 6424 | 96 | 3422.947932 | 3.857373 |
| `Hull` | 7128 | 1908 | 6.150966 | 0.042965 |
| `ILP` | 4231 | 3 | 3598.020537 | 140.411390 |
| `QCP` | 68 | 39 | 25.507809 | 1.578934 |
| `trivial` | 424 | 424 | - | - |

**Table 2: Running times of individual algorithms. First column indicates the number of instances solved by the algorithm, second column indicates for how many instances this approach was the fastest and the last two columns show the maximum and median times in seconds.**

This memory limit was not enough for 235 instances (in all cases this was the case of the `EST` algorithm). For both ILP and QCP programs we used the Gurobi Optimizer [26] (note that for QCP it is the same as in the previous paper [22]). Table 1 shows the absolute number of solved instances divided among datasets of PrefLib by combining all approaches. Table 2 shows the running times of each approach separately. We denote by `trivial` the solver which (trivially) resolves exactly the instances with $|C| \leq 3$ or $|V| \leq 2$ or $|V| \leq 3 \wedge |C| \leq 7$ all of which are 2-Euclidean.

*Running times.* We evaluated the performance of a solver by counting the number of instances that are solvable within a given time (see Figure 2). Out of a total of 7683 instances we solved, only 39 of them took longer than one second to solve by the fastest solver. This corresponds to the `Best` curve in Figure 2. Note that

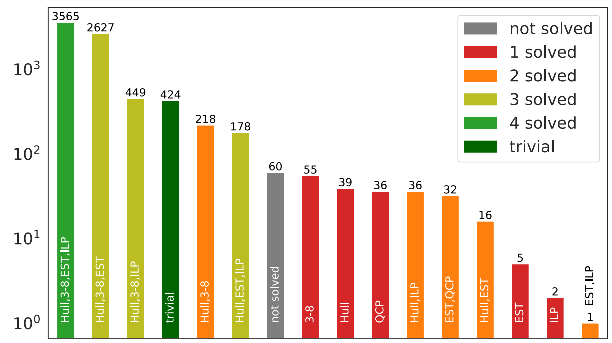


**Figure 3: Number of instances that were solved by exactly the solvers in the corresponding set.**

only the running time for successfully solved instances is displayed in Figure 2, rather than the time spent on instances that could not be decided. Experiments suggest, that `3-8` approach is the fastest for most instances. The only contender is the algorithm `Hull` as it solves 214 more instances and it also solves most of the instances under one second. The next fastest solver is the `EST` algorithm, although it is slower by few orders of magnitude. In few instances, it almost exceeds the time limit. However, based on the shape of its performance curve, it appears unlikely that many additional instances would be solved with `EST` algorithm in a reasonable time. In contrast, `ILP` is even slower by a few more orders of magnitude. However, its performance curve remains flat at the end, suggesting that if given additional time, the solver could potentially solve more instances. Lastly, `QCP`, which can only recognize yes-instances, solves only a small number of instances (since most PrefLib instances are not 2-Euclidean).

*Reduction rules.* The reduction rules were successfully applied to 802 instances. Together both reduction rules were used 979 times. Reduction Rule 1++ was applied 728 times and Reduction Rule 2 was applied 251 times. In total, 1729 candidates across all instances were removed, averaging more than 2 candidates per instances where any rule was applied. In most cases, the number of removed candidates was low – only 1 candidate was removed in 487 instances and only 2 candidates in 155 instances. However, in 70 instances, at least 5 candidates were removed; in 20 instances 10 or more candidates were removed, with the highest observed removal being 27 candidates in 2 instances.

*Evaluating Solver Necessity.* The number of solved instances and the running time alone do not fully capture the effectiveness of an approach. It is also crucial to consider whether a given approach can solve instances that other solvers cannot. In Figure 3, for a given set $S$ of solvers, we illustrate how many instances are resolved by all solvers in $S$. From the initial columns, we can observe that majority of the instances (over 7000) are either trivial or solvable by at least three different approaches. However, each solver is justified with at least one instance that no other solver can handle. Notably, all of the instances except 8 can be solved with the combination of `3-8`, `Hull` and `QCP`. For yes-instances, `QCP` recognizes all known cases except two solved only by `EST`.[3]

[3] In Figure 3 there are 5 instances solved only by `EST`, but 3 of them are no-instances.

*When to use which solver.* In some approaches we try all possible subsets of either votes (`Hull`, partially `3-8`) or candidates (`ILP`, partially `EST`). This theoretically identifies when a particular algorithm should be the best. In practice it is less clear which solver will be the fastest for a given number of candidates and voters. The `EST` solver is limited by design to a maximum of 9 candidates. By first reducing the instance size, we were able to solve 17 additional instances. However, the `EST` algorithm has not solved any yes-instance with more than 8 votes (after reductions). The `QCP` approach, on the other hand, was able to handle slightly larger instances, solving instances with up to 24 candidates and 12 votes. For small no-instances with at most 30 candidates, `Hull` outperforms `3-8`, but as instance sizes increase, `3-8` dominates due to its lower complexity ($O(|V|^3|C|^2)$, see Lemma 2.3).

*Unsolved instances.* There are still 60 unsolved instances in the PrefLib dataset. Of these, 16 instances contain 3 votes and at least 100 candidates. The other 44 unsolved instances are relatively small, they have at most 17 votes and at most 23 candidates. Minimal unsolved instances as a $(|C|, |V|)$ pair are of sizes $(12, 3)$, $(6, 6)$ and $(8, 5)$.

## 8 CONCLUSION

In this work we discussed a practical approach to recognizing 2-Euclidean preferences. We evaluated the performance of our approach on real-world datasets from PrefLib. We observed that most practical instances fail to be 2-Euclidean due to a very simple and quickly recognizable forbidden configuration contained in the instance. In particular, it is the forbidden pattern based on the convex hull of the voters or the 3-8 no-instance. Additionally, we designed an ILP that considers many more combinatorial properties of 2-Euclidean elections and attempts to falsify them. Towards recognizing yes-instances, we improved the QCP formulation that arises from the definition of 2-Euclidean elections. All our algorithms are also supported by reduction rules which remove trivial parts of the instance and reduce its size significantly. Using these techniques we were able to classify more than 82% of previously unclassified instances of PrefLib.

First open question that arises is how our approaches generalize to $d$-Euclidean elections for $d > 2$. For example, Reduction Rule 1+ generalizes well to $d > 2$ by simply replacing the size of the tail blocks by $d + 1$. On the other hand, it is not obvious how to generalize the convex hull approach to higher dimensions. How do controversial sets of voters of size greater than 2 contribute to the higher-dimensional convex hull? Does this characterization help significantly to provide a no-certificate for $d > 2$? How does the ILP approach generalize to higher dimensions?

Finally, it remains to classify the remaining 60 instances of PrefLib whether they are 2-Euclidean.

## ACKNOWLEDGMENTS

This work was co-funded by the European Union under the project Robotics and advanced industrial production (reg. no. CZ.02.01.01/00/22_008/0004590). Michal Dvořák, Jan Pokorný and Martin Slávik further acknowledge the support of the Grant Agency of the Czech Technical University in Prague, grant No. SGS23/205/OHK3/3T/18.

## REFERENCES


[1] Miguel A. Ballester and Guillaume Haeringer. 2011. A characterization of the single-peaked domain. *Social Choice and Welfare* 36, 2 (01 Feb 2011), 305–322. https://doi.org/10.1007/s00355-010-0476-3

[2] James Bennett, Stan Lanning, et al. 2007. The netflix prize. In *Proceedings of KDD cup and workshop*, Vol. 2007. 35.

[3] Joseph F. Bennett and William L. Hays. 1960. Multidimensional unfolding: Determining the dimensionality of ranked preference data. *Psychometrika* 25, 1 (01 Mar 1960), 27–43.

[4] Daniel Bertschinger, Nicolas El Maalouly, Linda Kleist, Tillmann Miltzow, and Simon Weber. 2023. The Complexity of Recognizing Geometric Hypergraphs. In *Graph Drawing and Network Visualization*, Michael A. Bekos and Markus Chimani (Eds.). Springer Nature Switzerland, 163–179.

[5] Bartłomiej Bindas, Krzysztof Kontek, Michał Ramsza, and Honorata Sosnowka. 2025. Serial Position Effect in Group Decision-Making. *Scientific Papers of Silesian University of Technology. Organization & Management/Zeszyty Naukowe Politechniki Slaskiej. Seria Organizacji i Zarzadzanie* 234 (2025).

[6] Niclas Boehmer and Nathan Schaar. 2022. Collecting, Classifying, Analyzing, and Using Real-World Elections. *CoRR* abs/2204.03589 (2022). https://doi.org/10.48550/arXiv.2204.03589 arXiv:2204.03589

[7] Anna Bogomolnaia and Jean-Francois Laslier. 2007. Euclidean preferences. *Journal of Mathematical Economics* 43, 2 (February 2007), 87–98.

[8] Robert Bredereck, Jiehua Chen, and Gerhard Woeginger. 2013. A characterization of the single-crossing domain. *Social Choice and Welfare* 41, 4 (October 2013), 989–998. https://doi.org/10.1007/s00355-012-0717-8

[9] Anna Bretscher, Derek Corneil, Michel Habib, and Christophe Paul. 2008. A Simple linear time LexBFS cograph recognition algorithm. *SIAM Journal on Discrete Mathematics* 22, 4 (2008), 1277 – 1296. https://doi.org/10.1137/060664690

[10] Laurent Bulteau and Jiehua Chen. 2022. 2-Dimensional Euclidean Preferences. arXiv:2205.14687 [cs.GT]

[11] Jean Cardinal, Stefan Felsner, Tillmann Miltzow, Casey Tompkins, and Birgit Vogtenhuber. 2018. Intersection Graphs of Rays and Grounded Segments. *Journal of Graph Algorithms and Applications* 22, 2 (2018), 273–294. https://doi.org/10.7155/jgaa.00470

[12] Jiehua Chen, Martin Nöllenburg, Sofia Simola, Anaïs Villedieu, and Markus Wallinger. 2022. Multidimensional Manhattan Preferences. arXiv:2201.09691 [cs.MA]

[13] Jiehua Chen, Kirk Pruhs, and Gerhard J. Woeginger. 2015. The one-dimensional Euclidean domain: Finitely many obstructions are not enough. *CoRR* abs/1506.03838 (2015). arXiv:1506.03838

[14] Derek G. Corneil. 2004. A simple 3-sweep LBFS algorithm for the recognition of unit interval graphs. *Discrete Applied Mathematics* 138, 3 (2004), 371 – 379. https://doi.org/10.1016/j.dam.2003.07.001

[15] J.P. Doignon and J.C. Falmagne. 1994. A Polynomial Time Algorithm for Unidimensional Unfolding Representations. *Journal of Algorithms* 16, 2 (1994), 218–233. https://doi.org/10.1006/jagm.1994.1010

[16] Saari Donald G. 2011. Chapter Twenty-Seven - Geometry of Voting. In *Handbook of Social Choice and Welfare*, Kenneth J. Arrow, Amartya Sen, and Kotaro Suzumura (Eds.). Handbook of Social Choice and Welfare, Vol. 2. Elsevier, 897–945. https://doi.org/10.1016/S0169-7218(10)00027-4

[17] Michal Dvořák, Dušan Knop, Jan Pokorný, and Martin Slávik. 2025. Practical approach to 2-Euclidean Preferences. arXiv:2502.07454 [cs.GT]

[18] Edith Elkind and Piotr Faliszewski. 2014. Recognizing 1-Euclidean Preferences: An Alternative Approach. In *Algorithmic Game Theory*.

[19] Edith Elkind, Martin Lackner, and Dominik Peters. 2022. *Preference restrictions in computational social choice: A survey*. Technical Report. arXiv preprint arXiv:2205.09092.

[20] Edith Elkind, Martin Lackner, and Dominik Peters. 2025. Preference Restrictions in Computational Social Choice: A Survey. arXiv:2205.09092 [cs.GT]

[21] Bruno Escoffier, Olivier Spanjaard, and Magdaléna Tydrichová. 2022. Weighted majority tournaments and Kemeny ranking with 2-dimensional Euclidean preferences. *Discret. Appl. Math.* 318 (2022), 6–12. https://doi.org/10.1016/J.DAM.2022.05.009

[22] Bruno Escoffier, Olivier Spanjaard, and Magdaléna Tydrichová. 2023. Algorithmic Recognition of 2-Euclidean Preferences. In *ECAI 2023 - 26th European Conference on Artificial Intelligence, September 30 - October 4, 2023, Kraków, Poland - Including 12th Conference on Prestigious Applications of Intelligent Systems (PAIS 2023) (Frontiers in Artificial Intelligence and Applications, Vol. 372)*. IOS Press, 637–644. https://doi.org/10.3233/FAIA230326

[23] Bruno Escoffier, Olivier Spanjaard, and Magdaléna Tydrichová. 2022. Euclidean preferences in the plane under $\ell_1$, $\ell_2$ and $\ell_\infty$ norms. arXiv:2202.03185 [math.MG]

[24] I.J Good and T.N Tideman. 1977. Stirling numbers and a geometric ,structure from voting theory. *Journal of Combinatorial Theory, Series A* 23, 1 (1977), 34–45. https://doi.org/10.1016/0097-3165(77)90077-2

[25] Paul E Green and Vithala R Rao. 1972. Applied multidimensional scaling: A comparison of approaches and algorithms. (1972).

[26] Gurobi Optimization, LLC. 2026. Gurobi Optimizer Reference Manual. https://www.gurobi.com

[27] Thekla Hamm, Martin Lackner, and Anna Rapberger. 2021. Computing Kemeny Rankings from d-Euclidean Preferences. In *Algorithmic Decision Theory - 7th International Conference, ADT 2021, Toulouse, France, November 3-5, 2021, Proceedings (Lecture Notes in Computer Science, Vol. 13023)*. Springer, 147–161. https://doi.org/10.1007/978-3-030-87756-9_10

[28] William L. Hays and Joseph F. Bennett. 1961. Multidimensional unfolding: Determining configuration from complete rank order preference data. *Psychometrika* 26, 2 (01 Jun 1961), 221–238. https://doi.org/10.1007/BF02289716

[29] Hidehiko Kamiya, Akimichi Takemura, and Hiroaki Terao. 2011. Ranking patterns of unfolding models of codimension one. *Adv. Appl. Math.* 47, 2 (2011), 379–400. https://doi.org/10.1016/J.AAM.2010.11.002

[30] Ross Kang and Tobias Müller. 2012. Sphere and Dot Product Representations of Graphs. *Discrete & Computational Geometry* 47, 3 (2012), 548–569. https://doi.org/10.1007/s00454-012-9394-8

[31] Vicki Knoblauch. 2010. Recognizing one-dimensional Euclidean preference profiles. *Journal of Mathematical Economics* 46, 1 (2010), 1–5. https://doi.org/10.1016/j.jmateco.2009.05.007

[32] Jan Kratochvíl. 1991. String graphs. II. Recognizing string graphs is NP-hard. *Journal of Combinatorial Theory, Series B* 52, 1 (1991), 67–78.

[33] Jan Kratochvíl and Jiří Matoušek. 1994. Intersection graphs of segments. *Journal of Combinatorial Theory. Series B* 62, 2 (1994), 289–315. https://doi.org/10.1006/jctb.1994.1071

[34] Casimir Kuratowski. 1930. Sur le problème des courbes gauches en Topologie. *Fundamenta Mathematicae* 15, 1 (1930), 271–283.

[35] C. Lekkeikerker and J. Boland. 1962. Representation of a finite graph by a set of intervals on the real line. *Fundamenta Mathematicae* 51, 1 (1962), 45–64.

[36] Nathan Lindzey and Ross M. McConnell. 2013. On Finding Lekkerkerker-Boland Subgraphs. *CoRR* abs/1303.1840 (2013). arXiv:1303.1840

[37] Krzysztof Magiera and Piotr Faliszewski. 2017. Recognizing Top-Monotonic Preference Profiles in Polynomial Time. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, Carles Sierra (Ed.). ijcai.org, 324–330. https://doi.org/10.24963/IJCAI.2017/46

[38] Andrew Mao, Ariel Procaccia, and Yiling Chen. 2013. Better human computation through principled voting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 27. 1142–1148.

[39] Jiří Matoušek. 2014. Intersection graphs of segments and $\exists\mathbb{R}$. *ArXiv 1406.2636* (2014).

[40] Nicholas Mattei and Toby Walsh. 2013. PrefLib: A Library of Preference Data HTTP://PREFLIB.ORG. In *Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT 2013) (Lecture Notes in Artificial Intelligence)*. Springer.

[41] Martin Milanič, Romeo Rizzi, and Alexandru I. Tomescu. 2014. Set graphs. II. Complexity of set graph recognition and similar problems. *Theoretical Computer Science* 547 (2014), 70–81. https://doi.org/10.1016/j.tcs.2014.06.017

[42] Tobias Müller, Erik Jan van Leeuwen, and Jan van Leeuwen. 2013. Integer representations of convex polygon intersection graphs. *SIAM Journal on Discrete Mathematics* 27, 1 (2013), 205–231. https://doi.org/10.1137/110825224

[43] Dominik Peters. 2017. Recognising Multidimensional Euclidean Preferences. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, Satinder Singh and Shaul Markovitch (Eds.). AAAI Press, 642–648. https://doi.org/10.1609/AAAI.V31I1.10616

[44] Marcus Schaefer, Eric Sedgwick, and Daniel Štefankovič. 2003. Recognizing string graphs in NP. *J. Comput. System Sci.* 67, 2 (2003), 365–380. https://doi.org/10.1016/S0022-0000(03)00045-X Special Issue on STOC 2002.

[45] Klaus Simon. 1991. A new simple linear algorithm to recognize interval graphs. In *Computational Geometry-Methods, Algorithms and Applications: International Workshop on Computational Geometry CG'91 Bern, Switzerland, March 21–22, 1991 Proceedings*. Springer, 289–308.

[46] Alan Tucker. 1972. A structure theorem for the consecutive 1's property. *Journal of Combinatorial Theory, Series B* 12, 2 (1972), 153–162. https://doi.org/10.1016/0095-8956(72)90019-6