Recognizing 1-Euclidean Preferences: An Alternative Approach

Edith Elkind[1] and Piotr Faliszewski[2]

[1] University of Oxford, Oxford, UK
[2] AGH University, Krakow, Poland

Abstract. We consider the problem of detecting whether a given election is 1-*Euclidean*, i.e., whether voters and candidates can be mapped to points on the real line so that voters' preferences over the candidates are determined by the Euclidean distance. A recent paper by Knoblauch [14] shows that this problem admits a polynomial-time algorithm. Knoblauch's approach relies on the fact that a 1-Euclidean election is necessarily single-peaked, and makes use of the properties of the respective candidate order to find a mapping of voters and candidates to the real line. We propose an alternative polynomial-time algorithm for this problem, which is based on the observation that a 1-Euclidean election is necessarily singe-crossing, and we use the properties of the respective voter order to find the desired mapping.

1 Introduction

There are many settings where agents express their preferences over a finite set of candidates by submitting full rankings of the candidates. Often, the set of candidates has a special structure, which influences the agents' preferences. For instance, it may be the case that voters and/or candidates can be mapped to points on the real line so that the agents' preferences are consistent with this mapping. Different instantiations of this idea give rise to such well-known preference domains as *single-peaked* preferences [4], *single-crossing* preferences [15], and 1-*dimensional Euclidean*, or 1-*Euclidean*, preferences [13].

In this paper, we study 1-Euclidean elections (though we will also discuss single-peaked and single-crossing elections, as all three domains are closely interrelated). These are elections that can be succinctly described by embedding both voters and candidates in the real line so that each voter prefers an alternative that is closer to her to the one that is further away. A typical situation that results in 1-Euclidean preferences is facility location on a line: there is a single facility (such as a bus stop, a playground, or a library) to be constructed in one of several possible locations along a street, and the voters (who are the residents of that street) want the facility to be located as close to them as possible. 1-Euclidean preferences can also arise in settings where the structure of the alternative space is not immediately obvious: for instance, political elections may turn out to be 1-Euclidean when the voters rank the candidates according to some combination of factors that happens to map onto the real line. It is then natural to ask whether, given an election, we can uncover its hidden metric structure, i.e., decide whether it is 1-Euclidean. This is the question that is the main focus of this paper.

R. Lavi (Ed.): SAGT 2014, LNCS 8768, pp. 146–157, 2014.

Before we discuss 1-Euclidean preferences in further detail, let us review the relationship between, on the one hand, 1-Euclidean preferences and, on the other hand, single-peaked and single-crossing preferences, as the nature of this relationship will play an important role for our algorithmic results. Recall that the agents' preferences are said to be *single-peaked* when the candidates can be ordered on the line so that each voter, when comparing two candidates located on the same side of her favorite point, prefers one that is closer to her top choice to the one that is further away from it; in contrast with 1-Euclidean domain, voter's preferences concerning two candidates located on different sides of her favorite point are unconstrained. On the other hand, a preference profile is said to be *single-crossing* if the voters can be ordered so that for each pair of candidates a, b, the "trajectories" of a and b in the voters' preferences cross at most once, i.e., if the first voter prefers a to b, then all voters who prefer a to b precede all voters who prefer b to a. Both of these domains have received a considerable amount of attention in social choice literature, as they have a number of desirable properties: for instance, both single-peaked and single-crossing elections are guaranteed to have a Condorcet winner, and admit a non-trivial strategyproof social choice rule [16,17,1]. Recently, it has also been shown that some of the algorithmic problems related to elections (such as winner determination, manipulation, bribery, and control) become easier if the voters' preferences belong to one of these domains [11,5,3,18]. Further, there are polynomial-time algorithms for checking whether an election is single-peaked [2,10] or single-crossing [6,9], and, if this is the case, finding an ordering of candidates or voters witnessing this. It is not hard to see that 1-Euclidean elections are both single-peaked and single-crossing [12]; however, the converse is not true, i.e., there are single-peaked single-crossing elections that are not 1-Euclidean [8,7].

It is fairly easy to see that, given an ordering of voters and candidates in an election E, we can efficiently check whether there is a mapping that places the voters and the candidates on the real line in a way that is consistent with this ordering and witnesses that E is 1-Euclidean; indeed, this question can be captured by a simple linear feasibility program (a variant of this observation is due to Knoblauch [14]; see also Proposition 3). Thus, checking whether an election is 1-Euclidean can be reduced to finding an appropriate ordering of voters and candidates. Since 1-Euclidean elections are single-peaked and single-crossing, given an input election E with a candidate set C and a voter set V, it is natural to first check whether E is single-peaked and single-crossing, and, if so, use the respective orderings of C and V to construct the required ordering of $C \cup V$. Indeed, a variant of this approach has been recently pursued by Knoblauch [14], who used an ordering of candidates witnessing that E is single-peaked as a starting point for her algorithm for checking whether E is 1-Euclidean. We discuss Knoblauch's algorithm in more detail in Section 5; at this point, we would like to mention that a single-peaked ordering of candidates is not unique, which causes considerable complications.

In contrast, in this paper we start with an ordering of voters witnessing that a given election $E = (C, V)$ is single-crossing, and show how to extend it to an ordering of $C \cup V$ witnessing that E is 1-Euclidean. The advantage of this approach is that, if E is single-crossing, there is effectively a unique ordering of voters that certifies this (this is shown in Proposition 1). As a result, we construct an algorithm for recognizing 1-Euclidean preferences that is arguably simpler than that of Knoblauch.

The rest of the paper is organized as follows. After introducing the necessary notation and formally defining single-peaked, single-crossing and 1-Euclidean preferences (Section 2), we state a few basic observations about the 1-Euclidean domain (Section 3), followed by a presentation of our algorithm (Section 4). We then provide an overview of Knoblauch's algorithm (Section 5). We conclude the paper by discussing topics for future research (Section 6).

2 Preliminaries

For every positive integer s, we let $[s]$ denote the set $\{1, \dots, s\}$. An *election* is a pair $E = (C, V)$ where $C = \{c_1, \dots, c_m\}$ is a set of candidates and $V = (v_1, \dots, v_n)$ is an ordered list of voters. Each voter $v_i \in V$ has a *preference order*, or *vote*, $\succ_i$, i.e., a linear order over C that ranks all the candidates from the most desirable one to the least desirable one. We refer to the list V as the *preference profile*. In what follows, we use the terms "election","preferences" and "preference profile" interchangeably.

We denote the most preferred candidate in a vote v_i by $\mathrm{top}(v_i)$. Given an election $E = (C, V)$ and a subset of candidates $D \subset C$, we denote by $V|_D$ the restriction of the preferences of the voters in V to D. Given two sets $A, B \subset C$, we write $\dots \succ A \succ B \succ \dots$ to denote a vote where all candidates in A appear above all candidates in B.

Euclidean, Single-Crossing and Single-Peaked Profiles. We will now define three important preference domains that will be considered in this paper.

Perhaps the most intuitive of the three is the domain of Euclidean preferences: both voters and candidates are identified with points on the real line (or, more generally, in $\mathbb{R}^d$), and the voters' preferences are determined by the Euclidean distance to the candidates.

Definition 1. *An election $E = (C, V)$ is said to be* d-Euclidean *if there is a mapping $x : C \cup V \to \mathbb{R}^d$ such that for every voter $v \in V$ and every pair of candidates $a, b \in C$ it holds that $a \succ_v b$ if and only if $\|x(v) - x(a)\|_d < \|x(v) - x(b)\|_d$, where $\|\cdot\|_d$ is the Euclidean norm on $\mathbb{R}^d$, i.e., for every vector $\mathbf{u} = (u_1, \dots, u_d) \in \mathbb{R}^d$ we have $\|\mathbf{u}\|_d = \left(u_1^2 + \dots + u_d^2\right)^{1/2}$.*

Note that in Definition 1 a voter cannot be equidistant from two distinct candidates a and b. One might argue that such a situation should be allowed, and the voter can then be indifferent between a and b, or break the tie arbitrarily. However, the former interpretation would not fit our model of preference orders being strict, and the latter would render the notion of d-Euclidean elections useless: Every election would be d-Euclidean (for each $d \in \mathbb{N}$), since we could map all voters and all candidates to a single point. One can deal with this objection by requiring that the positions of all candidates are distinct; the resulting model (and the associated algorithmic problem), while non-standard, deserves future study (see Section 6).

The notion of single-crossing preferences (sometimes also called *intermediate* preferences) dates back to the work of Mirrlees [15].

Definition 2. *An election $E = (C, V)$, where C is a set of candidates and $V = (v_1, \dots, v_n)$ is an ordered list of voters, is* single-crossing with respect to V *if for every pair of candidates $a, b \in C$ such that $a \succ_1 b$, there exists a $t \in [n]$ such that $\{i \in [n] \mid a \succ_i b\} = [t]$.*

Definition 2 refers to the ordering of the voters given by V. Alternatively, we could say that an election is single-crossing if the voters can be reordered so that the condition of Definition 2 is satisfied. However, from the algorithmic perspective, this distinction is not essential: one can compute an order of the voters that makes an election single-crossing or decide that such an order does not exist, in polynomial time [9,6].

Another relevant concept is that of single-peaked preferences [4].

Definition 3. *Let $\succ$ be a preference order over a candidate set C and let $\lhd$ be an order over C. We say that $\succ$ is* single-peaked with respect to $\lhd$ *if for every triple of candidates $a, b, c \in C$ such that $a \lhd b \lhd c$ or $c \lhd b \lhd a$ it holds that $a \succ b$ implies $b \succ c$. An election $E = (C, V)$ is* single-peaked with respect to an order $\lhd$ over C *if the preference order of every voter $v \in V$ is single-peaked with respect to $\lhd$. An election $E = (C, V)$ is* single-peaked *if there exists an order $\lhd$ over C with respect to which it is single-peaked.*

If E is single-peaked with respect to some order $\lhd$ then we call $\lhd$ a *societal axis* for E. There are polynomial-time algorithms that, given an election E, decide if it is single-peaked and if so, compute a societal axis for it [2,10].

It is not hard to show that 1-Euclidean elections are both single-peaked and single-crossing, see, e.g., [12]. However, there exist elections that are both single-peaked and single-crossing, but not 1-Euclidean [7,8].

3 Basic Observations

We will now present some simple observations that will be used in the analysis of our algorithm.

Our first observation is that for single-crossing elections there is effectively a unique ordering of voters that witnesses that it is single-crossing, up to flipping the entire ordering and reordering identical voters.

Proposition 1. *Consider an election $E = (C, V)$ that is single-crossing with respect to $V = (v_1, \dots, v_n)$. If the preferences of the voters in V are pairwise distinct, then the only other order of the voters witnessing that E is single-crossing is $(v_n, \dots, v_1)$.*

Proof. The proof is by induction on n. For $n = 1$ and $n = 2$, our claim is trivially true. Now suppose that it is true for $n - 1$, where $n \geq 3$; we will show that it is true for n.

Since $v_1 \neq v_2$, there is a pair of candidates a, b such that v_1 prefers a to b, but v_2 prefers b to a. Since E is single-crossing, all voters other than v_1 prefer b to a. Therefore, if E is single-crossing with respect to some order $\widehat{V}$, then voter v_1 has to be first or last in $\widehat{V}$.

Now, consider the election (C, V'), where $V' = (v_2, \dots, v_n)$. It is single-crossing and has $n - 1$ voters, so by the induction hypothesis the voters in V' have to be ordered as $(v_2, \dots, v_n)$ or $(v_n, \dots, v_2)$.

It remains to argue that E is not single-crossing with respect to $(v_2, \dots, v_n, v_1)$ and $(v_1, v_n, \dots, v_2)$. It suffices to consider the first of these two orderings. Since $n \geq 3$ and we assume that all preferences are pairwise distinct, there is a pair of candidates c, d such that v_2 prefers c to d, but v_n prefers d to c. Since the original election is single-crossing, it has to be the case that v_1 prefers c to d. But this means that E is not single-crossing with respect to $(v_2, \dots, v_n, v_1)$. $\square$

In contrast, there can be exponentially many axes witnessing that a given election is single-peaked [10]. For instance, a unanimous election where all voters order the candidates as $c_1 \succ \cdots \succ c_m$ is single-peaked with respect to 2^{m-1} axes (any subset of $\{c_2, \ldots, c_m\}$ can appear to the left of c_1 on the axis).

We will now list some useful properties of 1-Euclidean elections.

Proposition 2. *Let $E = (C, V)$ be a 1-Euclidean election with $V = (v_1, \ldots, v_n)$ such that all votes are pairwise distinct, and let $x : C \cup V \to \mathbb{R}$ be some mapping witnessing that E is 1-Euclidean. Assume without loss of generality that $x(v_1) < x(v_n)$. Suppose that $\mathrm{top}(v_1) = a$, $\mathrm{top}(v_n) = b$ (and hence $x(a) < x(b)$). Let*

$$C_L = \{c \in C \mid x(c) < x(a)\},$$
$$C_M = \{c \in C \mid x(a) \le x(c) \le x(b)\},$$
$$C_R = \{c \in C \mid x(c) > x(b)\}.$$

Then

(1) $C_M = \{c \in C \mid c \succ_1 b$ and $c \succ_n a\} \cup \{a, b\}$.
(2) For every pair of candidates $c, d \in C$, if $c \succ_1 d$, but $d \succ_n c$, then $c \in C_L \cup C_M$ and $d \in C_M \cup C_R$.

Proof. Obviously, $a, b \in C_M$. If $c \in C_M$ and $c \neq a, b$, then v_1 ranks c above b and v_n ranks c above a. On the other hand, if $c \in C_R$ then v_1 ranks c below b and if $c \in C_L$, then v_n ranks c below a. This proves our first claim.

To prove the second claim, consider a pair of candidates $c, d \in C$ such that v_1 prefers c to d, but v_n prefers d to c. If $c \in C_R$, we have $x(v_1) < x(v_n) \le x(c)$. Now, if v_1 prefers c to d, we have $x(c) < x(d)$, which implies that v_n, too, prefers c to d, a contradiction. If $d \in C_L$, we obtain a contradiction as well by a similar argument. □

Finally, once we have an ordering of candidates, finding a mapping x that witnesses that an election is 1-Euclidean and is consistent with this ordering reduces to solving a system of linear inequalities. A variant of this observation is due to Knoblauch [14]; however, Knoblauch's reduction produces a system of *strict* inequalities, and standard tools of linear programming are not directly applicable to such systems. The following proposition shows that for a fixed ordering of candidates our problem can be reduced to a system of *non-strict* inequalities.

Proposition 3. *There exists a polynomial-time algorithm that, given an election $E = (C, V)$ and an ordering of candidates $\lhd$, decides whether there exists a mapping $x : C \cup V \to \mathbb{R}$ that witnesses that E is 1-Euclidean and respects $\lhd$, i.e., such that for every pair of candidates $a, b \in C$ it holds that $x(a) < x(b)$ if and only if $a \lhd b$.*

Proof. We introduce a real variable x_v for each $v \in V$ and a real variable x_c for each $c \in C$; these variables encode the positions of voters and candidates on the real line. For every pair of candidates $a, b \in C$ such that $a \lhd b$ we introduce the inequality $x_a + 1 \le x_b$. Also, for every voter $v \in V$, if v prefers a to b, we introduce the inequality $x_v + 1 \le (x_a + x_b)/2$, and if v prefers b to a, we introduce the inequality $x_v \ge (x_a + x_b)/2 + 1$. Thus, altogether we introduce $(n + 1)m(m - 1)/2$ inequalities. It is easy to see that every feasible solution to this linear program (LP) describes a mapping x that respects $\lhd$ and witnesses that E is 1-Euclidean.

Conversely, if E is 1-Euclidean and this is witnessed by a mapping x that respects $\lhd$, this LP has a feasible solution. Indeed, set $\delta_1 = \min_{a,b\in C} |x(a) - x(b)|$, $\delta_2 = \frac{1}{2}\min_{a,b\in C,v\in V} |2x(v) - x(a) - x(b)|$. Note that $\delta_1 > 0$, since otherwise there would be a pair of candidates a, b with $x(a) = x(b)$ and then the voters would be indifferent between a and b. Similarly, $\delta_2 > 0$ since otherwise there would be a pair of candidates a, b and a voter v such that $|x(v) - x(a)| = |x(v) - x(b)|$, i.e., voter v would be indifferent between a and b. We can now set $\delta = \min\{\delta_1, \delta_2\}$ and $x_z = x(z)/\delta$ for all $z \in C \cup V$; it is easy to see that this provides a feasible solution to our LP. □

In what follows, we refer to the algorithm that takes a candidate set C, a voter list V and an ordering $\lhd$ of C as its input, and returns a mapping $x : C \cup V \to \mathbb{R}$ that corresponds to a feasible solution to our LP (or $\perp$ if this LP admits no feasible solution) as LP$(C, V, \lhd)$.

4 Algorithm

We are now ready to present our algorithm.

Theorem 1. *Given an election $E = (C, V)$ with $C = \{c_1, \ldots, c_m\}$, $V = (v_1, \ldots, v_n)$, we can decide in time polynomial in n and m whether E is 1-Euclidean, and, if so, construct a mapping x that witnesses this.*

Proof. We can assume without loss of generality that the voters' preferences are pairwise distinct. Indeed, if this is not the case, we can simply remove the "duplicate" voters: the resulting election is 1-Euclidean if and only if the original one is. Also, we can assume that $n > 1$, since otherwise the election is clearly 1-Euclidean.

We first verify that E is single-crossing and output $\perp$ (indicating that E is not 1-Euclidean) if this is not the case. From now on, we assume that E is single-crossing with respect to the voter order $(v_1, \ldots, v_n)$; note that by Proposition 1 this order is unique up to a reversal. We then execute Algorithm 1. This algorithm consists of three main stages. First, it colors the candidates red, green, blue, or grey based on the preferences of voter 1 and voter n (lines 2–14). We will argue that this is done in such a way that the set of red candidates is exactly C_M, blue candidates are contained in C_R, and green candidates are contained in C_L (see Proposition 2 for definitions of these sets). Then our algorithm defines a complete order $\lhd$ on the set C^+ that consists of all non-grey candidates (lines 16–24). This order is passed to the algorithm described in Proposition 3, which places the voters and the candidates in C^+ on the real line (lines 25–28). The result of this step is a mapping $x : V \cup C^+ \to \mathbb{R}$. Finally, the algorithm inserts the grey candidates. To this end, it partitions the grey and non-grey candidates into groups according to the order of their appearance in the preferences of the first voter (line 29). After placing the voters and the first group of each type (lines 32–33), it "stretches" x to ensure that different non-grey groups are well-separated (lines 34–39), and inserts the grey candidates into the appropriate spaces (lines 40–41).

Algorithm 1: 1-Euclidean

Input: a single-crossing election $E = (C, V)$.
Output: a mapping $y : C \cup V \to \mathbb{R}$ witnessing that E is 1-Euclidean, or $\perp$ if E is not 1-Euclidean.

```
$c^- \leftarrow \mathrm{top}(v_1)$, $c^+ \leftarrow \mathrm{top}(v_n)$;
foreach $c \in C$ do
    if ($c \succ_1 c^+$ and $c \succ_n c^-$) or $c \in \{c^-, c^+\}$ then
        $\gamma(c) \leftarrow$ red
    else
        $\gamma(c) \leftarrow$ grey
foreach $a, b \in C$ do
    if ($a \succ_1 b$ and $b \succ_n a$) then
        if $\gamma(a) =$ blue or $\gamma(b) =$ green then
            return $\perp$
        if $\gamma(a) =$ grey then
            $\gamma(a) \leftarrow$ green
        if $\gamma(b) =$ grey then
            $\gamma(b) \leftarrow$ blue
$C^+ \leftarrow \{c \mid \gamma(c) \neq \text{grey}\}$, $C^- \leftarrow \{c \mid \gamma(c) = \text{grey}\}$;
foreach $a, b \in C^+$ do
    if ($\gamma(a) =$ green and $\gamma(b) =$ red) or
       ($\gamma(a) =$ red and $\gamma(b) =$ blue) or
       ($\gamma(a) =$ green and $\gamma(b) =$ blue) then
        set $a \lhd b$
    if ($\gamma(a) = \gamma(b) =$ red and $a \succ_1 b$) or
       ($\gamma(a) = \gamma(b) =$ blue and $a \succ_1 b$) or
       ($\gamma(a) = \gamma(b) =$ green and $b \succ_n a$) then
        set $a \lhd b$
if LP$(C^+, V|_{C^+}, \lhd) = \perp$ then
    return $\perp$
else
    $x \leftarrow$ LP$(C^+, V|_{C^+}, \lhd)$
Represent $\succ_1$ as $F_1 \succ_1 G_1 \succ_1 \ldots \succ_1 F_k \succ_1 G_k$, where $F_i \subseteq C^+$, $G_i \subseteq C^-$, $F_i \neq \emptyset$
for all $i \in [k]$, $G_i \neq \emptyset$ for all $i \in [k-1]$, $G_i = \{g^i_1, \ldots, g^i_{s_i}\}$, where $g^i_1 \succ_1 \ldots \succ_1 g^i_{s_i}$
for all $i \in [k]$;
$x^L \leftarrow \min_{t \in V \cup F_1} x(t)$, $x^R \leftarrow \max_{t \in V \cup F_1} x(t)$;
$\Delta \leftarrow \max_{t,t' \in V \cup C^+} |x(t) - x(t')|$;
foreach $t \in V \cup F_1$ do $y(t) \leftarrow x(t)$
foreach $g^1_\ell \in G_1$ do $y(g^1_\ell) \leftarrow x^R + 6\Delta + \frac{\ell}{m}\Delta$
foreach $i = 2, \ldots, k$ do
    foreach $c \in F_i$ do
        if $x(c) < x^L$ then
            $y(c) \leftarrow x(c) - (i+1)^2\Delta$
        if $x(c) > x^R$ then
            $y(c) \leftarrow x(c) + (i+1)^2\Delta$
    foreach $\ell = 1, \ldots, s_i$ do
        $y(g^i_\ell) \leftarrow x^R + (i+1)^2\Delta + 2\Delta + \frac{\ell}{m}\Delta$
return $y$;
```

The algorithm then returns the resulting mapping y. Note that the algorithm returns $\perp$ if the input election is not single-crossing, or the coloring stage cannot be completed (line 10), or the linear program does not have a feasible solution (line 26).

We will now argue that this algorithm is correct. Suppose that E is 1-Euclidean, and consider an arbitrary mapping z with $z(v_1) < z(v_n)$ that witnesses this. Let $c^- = \mathrm{top}(v_1)$, $c^+ = \mathrm{top}(v_n)$, and observe that $z(c^-) < z(c^+)$. Let $C_L = \{c \mid z(c) < z(c^-)\}$, $C_M = \{c \mid z(c^-) \leq z(c) \leq z(c^+)\}$, $C_R = \{c \mid z(c) > z(c^+)\}$. The first claim of Proposition 2 implies that the set of all red candidates is exactly C_M. Further, the second claim of Proposition 2 implies that if the conditions in line 8 are satisfied then $a \in C_L \cup C_M$ and $b \in C_M \cup C_R$. This explains why our algorithm outputs $\perp$ in line 10: if we have already decided that a is blue (and hence $a \in C_R$) or that b is green (and hence $b \in C_L$), we obtain a contradiction. Also, it follows that every green candidate belongs to C_L and every blue candidate belongs to C_R. Indeed, when we color a green in line 12, we know from line 8 that $a \in C_L \cup C_M$, and if a were in C_M, it would have been colored red already; the argument for blue candidates is similar.

Now, suppose that the algorithm has not output $\perp$ in line 10, i.e., we have consistently colored some set of candidates C^+ red, green, and blue. Clearly, we have $z(a) < z(b) < z(c)$ for any $a \in C_L$, $b \in C_M$, $c \in C_R$. Moreover, if $a, b \in C_M \cup C_R$ then $z(a) < z(b)$ if and only if $a \succ_1 b$ and if $a, b \in C_L$ then $z(a) < z(b)$ if and only if $b \succ_1 a$. Thus, the ordering $\lhd$ constructed in lines 16–24 is consistent with z. Note that z is an arbitrary mapping witnessing that E is 1-Euclidean such that $z(v_1) < z(v_n)$; our argument shows that every such mapping orders C^+ in the same way.

We now invoke Proposition 3. If E is 1-Euclidean, then so is $(C^+, V|_{C^+})$, and our mapping z witnesses this fact (clearly, z is consistent with $\lhd$). Thus, LP$(C^+, V|_{C^+}, \lhd)$ outputs some such mapping $x : V \cup C^+ \to \mathbb{R}$. On the other hand, if LP$(C^+, V|_{C^+}, \lhd) = \perp$, then there is no mapping witnessing that E is 1-Euclidean that is consistent with $\lhd$, and hence, as argued above, E is not 1-Euclidean.

Now, suppose that LP$(C^+, V|_{C^+}, \lhd)$ returned a mapping $x : V \cup C^+ \to \mathbb{R}$ witnessing that $(C^+, V|_{C^+})$ is 1-Euclidean. In line 29 we represent the preference ordering of the first voter as an alternating sequence of non-grey and grey blocks; the first block is non-grey since $c^- = \mathrm{top}(v_1)$ is red. Since all blocks, except possibly the last grey block, are required to be non-empty, this representation is unique. If $G_1 = \emptyset$, we have $C = C^+$, so we are done. Thus, assume that $G_1 \neq \emptyset$. The following lemmas present some useful observations about the sets F_i and G_i for $i \in [k]$.

Lemma 1. *We have $F_1 \succ_i G_1 \succ_i \ldots \succ_i F_k \succ_i G_k$ for all $i \in [n]$. Moreover, if $a, b \in C^-$ then $a \succ_i b$ if and only if $a \succ_1 b$.*

Proof. Consider a candidate $a \in C^-$. Since a remained grey by the end of the coloring stage, v_1 and v_n agree on all comparisons involving a. Since E is single-crossing with respect to $(v_1, \ldots, v_n)$, this means that for every candidate $b \neq a$ either all voters prefer a to b or all voters prefer b to a. This immediately implies our second claim. For the first claim, consider a voter v_i and a pair of candidates $a \in F_j$, $b \in G_j$ for some $j \geq 1$. We have $a \succ_1 b$, so, by the argument above, $a \succ_i b$. Similarly, if $a \in G_j$, $b \in F_{j+1}$ for some j, $1 \leq j < k$, then $a \succ_1 b$, so, by the argument above, $a \succ_i b$. Now our claim follows by induction on j. $\square$

Lemma 2. *For all $j = 2, \dots, k$ and all $c \in F_j$ we have $x(c) \notin [x^L, x^R]$, where x^L, x^R are defined in line 30 of our algorithm.*

Proof. Fix a j with $2 \le j \le k$ and a candidate $c \in F_j$. Assume for the sake of contradiction that $x(c) \in [x^L, x^R]$. We consider the following four cases.

- $x^L = x(a)$, $x^R = x(b)$ for some $a, b \in F_1$. Then either $x(v_1) \le x(c)$, in which case v_1 prefers c to b, or $x(v_1) > x(c)$, in which case v_1 prefers c to a.
- $x^L = x(a)$, $x^R = x(v_i)$ for some $a \in F_1$, $v_i \in V$. Then $c \succ_i a$.
- $x^L = x(v_i)$, $x^R = x(b)$ for some $b \in F_1$, $v_i \in V$. Then $c \succ_i b$.
- $x^L = x(v_i)$, $x^R = x(v_\ell)$ for some $v_i, v_\ell \in V$. Pick some $a \in F_1$ (note that $F_1 \neq \emptyset$). If $x(a) < x(c)$, then v_ℓ prefers c to a, and if $x(a) > x(c)$, then v_i prefers c to a.

In all cases, we obtain a contradiction, since by Lemma 1 we have $a \succ_i c$ for all $a \in F_1$ and all $i \in [n]$. □

Now, consider the mapping y returned in line 42. Fix an arbitrary voter v_i and a pair of candidates $a, b \in C$. To complete the proof of correctness, we will show that $|y(v_i) - y(a)| < |y(v_i) - y(b)|$ if and only if $a \succ_i b$. Note that the quantity Δ defined in line 31 satisfies $\Delta > 0$, since otherwise it would be the case that $x(v_1) = x(v_n)$, in which case all voters have the same preference order, and we assumed that this is not the case.

It suffices to consider the following six cases; the remaining cases follow by the transitivity of $\succ_i$.

- $a, b \in F_1$. Then we have $y(a) = x(a)$, $y(b) = x(b)$, $y(v_i) = x(v_i)$. Since $a, b \in C^+$ and x is a witness that $(C^+, V|_{C^+})$ is 1-Euclidean, our claim follows.
- $a, b \in F_j$, $j \ge 2$. We have $y(v_i) \in [x^L, x^R]$, and $y(a), y(b) \notin [x^L, x^R]$ by Lemma 2. Thus,

  $$|y(v_i) - y(a)| = |x(v_i) - x(a)| + (j+1)^2\Delta,\ |y(v_i) - y(b)| = |x(v_i) - x(b)| + (j+1)^2\Delta$$

  (lines 35–39). Again, our claim follows, since $a, b \in C^+$ and x is a witness that $(C^+, V|_{C^+})$ is 1-Euclidean.
- $a, b \in G_j$, $j \ge 1$. Assume without loss of generality that $a \succ_i b$. Then by Lemma 1 we have $a \succ_1 b$ and hence $x^R < y(a) < y(b)$ (lines 33 and 41). Since $x(v_i) \le x^R$, our claim follows.
- $a \in F_1, b \in G_1$. By Lemma 1 we have $a \succ_i b$. On the other hand, $y(a) \in [x^L, x^R]$, $y(v_i) \in [x^L, x^R]$, whereas $y(b) > x^R + 6\Delta$. Since $|x^R - x^L| \le \Delta$, the claim follows.
- $a \in F_j$, $b \in G_j$, $j > 1$. By Lemma 1 we have $a \succ_i b$. Since $j > 1$, we have $x(a) \in [x^L - \Delta, x^R + \Delta] \setminus [x^L, x^R]$, so we have

  $$\begin{aligned} y(a) &\in [x^R + (j+1)^2\Delta, x^R + (j+1)^2\Delta + \Delta] \\ &\quad \cup [x^L - (j+1)^2\Delta - \Delta, x^L - (j+1)^2\Delta], \\ y(b) &\in [x^R + (j+1)^2\Delta + 2\Delta + \frac{\Delta}{m}, x^R + (j+1)^2\Delta + 3\Delta]. \end{aligned}$$

Since $y(v_i) \in [x^L, x^R]$, we have

$$|y(a) - y(v_i)| \le (j+1)^2\Delta + 2\Delta, \quad |y(b) - y(v_i)| > (j+1)^2\Delta + 2\Delta,$$

and our claim follows.

- $a \in G_j$, $b \in F_{j+1}$, $1 \le j < k$. Again, by Lemma 1 we have $a \succ_i b$, and $x(b) \in [x^L - \Delta, x^R + \Delta] \setminus [x^L, x^R]$, so

$$\begin{aligned} y(a) &\in [x^R + (j+1)^2\Delta + 2\Delta + \frac{\Delta}{m}, x^R + (j+1)^2\Delta + 3\Delta], \\ y(b) &\in [x^R + (j+2)^2\Delta, x^R + (j+2)^2\Delta + \Delta] \\ &\quad \cup [x^L - (j+2)^2\Delta - \Delta, x^L - (j+2)^2\Delta]. \end{aligned}$$

Hence,

$$|y(a) - y(v_i)| \le (j+1)^2\Delta + 4\Delta, \quad |y(b) - y(v_i)| \ge (j+2)^2\Delta;$$

since $(j+1)^2 + 4 < (j+2)^2$ for all $j \ge 1$, our claim follows.

It remains to analyze the running time of our algorithm. One can check whether an election is single-crossing and, if so, determine the voter order that witnesses this, in time $O(nm^2)$ [6]. Further, our procedures for coloring the candidate set and constructing the order $\lhd$ run in time $O(m^2)$. The algorithm described in the proof of Proposition 3 is based on solving a linear program with coefficients in $\{-1, -\frac{1}{2}, 0, \frac{1}{2}, 1\}$, $O(n+m)$ variables, and $O(nm^2)$ constraints. Finally, the mapping y is computed by performing a constant number of arithmetic operations for each voter or candidate, and these operations involve numbers that form a feasible solution to our linear program. Thus, the overall running time of our algorithm is polynomial in n and m, and is dominated by the time needed to solve the linear program. □

Remark 1. Note that, while we chose to place the candidates in G_i to the right of the candidates in F_i, we could have also placed them to the left of the candidates in F_i. Further, instead of dealing with an entire grey block in a single step, we could have processed the grey candidates one by one. This shows that, after the end of the coloring stage, we can arbitrarily color all grey candidates green or blue, use this coloring to construct an order $\lhd$ on C, and apply Proposition 3 to E and $\lhd$. While the resulting algorithm is simpler, it may require solving a much larger linear program.

Remark 2. The reader may wonder if stretching x (lines 34–39) is necessary to place the candidates in C^-: perhaps we can find suitable positions for them without modifying the positions of the candidates in C^+? The following example shows that this is not always the case. Consider an election $E = (C, V)$ with $C = \{a, b, c, d, e\}$, $V = (u, v)$, where u ranks the candidates as $a \succ b \succ c \succ d \succ e$ and v ranks the candidates as $b \succ a \succ c \succ e \succ d$. For this election we have $F_1 = \{a, b\}$, $G_1 = \{c\}$, $F_2 = \{d, e\}$ and the ordering $\lhd$ over C^+ is given by $d \lhd a \lhd b \lhd e$. A feasible solution to the corresponding linear program is $x_u = -2$, $x_v = 2$, $x_a = -6$, $x_b = 6$, $x_d = -12$, $x_e = 12$. Now, suppose that we want to place c on the real line without changing the positions of other points. Since our construction is symmetric, we can assume without loss of generality that c should be placed to the right of 0. Since v prefers a to c, it has to be the case that $x_c > 10$. However, this means that u prefers d to c, a contradiction.

However, one can eliminate the stretching steps by adding constraints saying that different non-grey blocks are well-separated to the linear program itself. Then each grey block can be simply placed in the middle of the respective interval.

5 An Overview of Knoblauch's Algorithm

The main difference between our algorithm and that of Knoblauch is that the latter uses a single-peaked ordering of the candidates as its starting point. It then partitions the candidates into groups so that, for each group, the ordering of the candidates in this group is the same (up to reversal) for all societal axes witnessing that the election is single-peaked. This partition is fairly straightforward to derive from the votes, and can be shown to be a refinement of the partition $\{F_1, G_1, \ldots, F_k, G_k\}$ implicitly constructed by our algorithm. The groups that correspond to subsets of C^+ are then "glued together", i.e., the algorithm defines an ordering on C^+; this procedure is the heart of the algorithm, and is quite complicated. From this point on, Knoblauch's algorithm proceeds in the same manner as our algorithm: it uses a linear program to embed C^+ and V into the real line, and then places the candidates from C^-. However, both of these steps are implemented somewhat differently. In more detail, Knoblauch's linear program only has variables for elements of C^+, and the number of inequalities in it is bounded by $O(nm^4)$; effectively, it is obtained from our linear program by variable elimination. While it uses strict inequalities, it is not hard to modify it so that only non-strict inequalities are used (see Proposition 3). Finally, Knoblauch's algorithm places the candidates in C^- one by one rather than in blocks; whenever a candidate in C^- is placed, some of the candidates in C^+ are shifted by an unspecified "large enough" amount. As a consequence, candidates in C^+ may be shifted multiple times.

In terms of performance, neither algorithm has a clear advantage over the other: the running time of both algorithms is dominated by solving a linear program, and the two linear programs are closely related. Thus, our main contribution is conceptual: we provide a quick and simple method for obtaining an ordering of the non-grey candidates that is based on the single-crossing property of 1-Euclidean elections. We find it remarkable that our algorithm does not use the fact that a 1-Euclidean election is single-peaked, whereas Knoblauch's paper does not mention single-crossing elections at all; thus, the two approaches provide very different perspectives on the problem at hand.

6 Future Work

We have presented an alternative algorithm for recognizing whether an election is 1-Euclidean. Both our algorithm and that of Knoblauch rely on solving a linear program. A natural question is whether this step can be eliminated, i.e., whether our problem admits a purely combinatorial algorithm.

Further, it would be interesting to see if our algorithm (or that of Knoblauch) can be extended to higher dimensions, i.e., the problem of recognizing whether an election is d-Euclidean for $d > 1$. We remark that, while d-Euclidean elections with $d > 1$ are not particularly attractive from a purely social choice-theoretic perspective (e.g., such elections are not guaranteed to have a Condorcet winner), it is plausible that they may admit efficient algorithms for problems in computational social choice that are known to be hard on the general domain.

Another promising direction is to explore whether our ideas can be used to identify elections that are close to being 1-Euclidean, for an appropriate notion of distance. A challenge that one would need to cope with in this context is that an "almost Euclidean" election need not be single-peaked or single-crossing. One can also consider a variant of the 1-Euclidean model where a voter can be equidistant from two different candidates, in which case she may prefer either of these candidates, but the positions of all candidates are required to be pairwise distinct, and ask whether preference profiles that are 1-Euclidean in this sense can be recognized in polynomial time.

References

1. Barberà, S., Moreno, B.: Top monotonicity: A common root for single peakedness, single crossing and the median voter result. Games and Economic Behavior 73(2), 345–359 (2011)
2. Bartholdi III, J., Trick, M.: Stable matching with preferences derived from a psychological model. Operations Research Letters 5(4), 165–169 (1986)
3. Betzler, N., Slinko, A., Uhlmann, J.: On the computation of fully proportional representation. Journal of Artificial Intelligence Research 47, 475–519 (2013)
4. Black, D.: On the rationale of group decision-making. Journal of Political Economy 56(1), 23–34 (1948)
5. Brandt, F., Brill, M., Hemaspaandra, E., Hemaspaandra, L.: Bypassing combinatorial protections: Polynomial-time algorithms for single-peaked electorates. In: AAAI 2010, pp. 715–722 (2010)
6. Bredereck, R., Chen, J., Woeginger, G.: A characterization of the single-crossing domain. Social Choice and Welfare 41(4), 989–998 (2013)
7. Chen, J., Pruhs, K., Woeginger, G.: Characterizations of the one-dimensional Euclidean domain. Manuscript (2014)
8. Elkind, E., Faliszewski, P., Skowron, P.: A characterization of the single-peaked single-crossing domain. In: AAAI 2014 (2014)
9. Elkind, E., Faliszewski, P., Slinko, A.: Clone structures in voters' preferences. In: ACM EC 2012, pp. 496–513 (2012)
10. Escoffier, B., Lang, J., Öztürk, M.: Single-peaked consistency and its complexity. In: ECAI 2008, pp. 366–370 (2008)
11. Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L., Rothe, J.: The shield that never was: Societies with single-peaked preferences are more open to manipulation and control. Information and Computation 209(2), 89–107 (2011)
12. Grandmont, J.M.: Intermediate preferences and the majority rule. Econometrica 46(2), 317–330 (1978)
13. Hotelling, H.: Stability in competition. Economic Journal 39, 41–57 (1929)
14. Knoblauch, V.: Recognizing one-dimensional euclidean preference profiles. Journal of Mathematical Economics 46, 1–5 (2010)
15. Mirrlees, J.: An exploration in the theory of optimal income taxation. Review of Economic Studies 38, 175–208 (1971)
16. Moulin, H.: On strategy-proofness and single peakedness. Public Choice 35(4), 437–455 (1980)
17. Saporiti, A.: Strategy-proofness and single-crossing. Theoretical Economics 4(2), 127–163 (2009)
18. Skowron, P., Yu, L., Faliszewski, P., Elkind, E.: The complexity of fully proportional representation for single-crossing electorates. In: Vöcking, B. (ed.) SAGT 2013. LNCS, vol. 8146, pp. 1–12. Springer, Heidelberg (2013)