Journal of Artificial Intelligence Research 66 (2019) 57-84 Submitted 01/2019; published 09/2019

Recognizing Top-Monotonic Preference Profiles in Polynomial Time

Krzysztof Magiera KRZYS.MAGIERA@GMAIL.COM
Piotr Faliszewski FALISZEW@AGH.EDU.PL
AGH University, Krakow, Poland

Abstract

We provide the first polynomial-time algorithm for recognizing if a profile of (possibly weak) preference orders is top-monotonic. Top-monotonicity is a generalization of the notions of single-peakedness and single-crossingness, defined by Barberà and Moreno. Top-monotonic profiles always have weak Condorcet winners and satisfy a variant of the median voter theorem. Our algorithm proceeds by reducing the recognition problem to the SAT-2CNF problem.

1. Introduction

The Condorcet paradox refers to situations where a group of individuals—such that each of them ranks a set of objects from the best to the worst one—has cyclic collective preference under majority voting. For example, consider objects a, b, and c (referred to as candidates) and three individuals, v_1, v_2, and v_3 (referred to as voters), with preference orders:

$$v_1 \colon a \succ b \succ c, \qquad v_2 \colon b \succ c \succ a, \qquad v_3 \colon c \succ a \succ b.$$

A majority of the voters (v_1 and v_3) prefers a to b, a majority of the voters (v_1 and v_2) prefers b to c, and a majority of the voters (v_2 and v_3) prefers c to a. The fact that a Condorcet paradox can occur in preference aggregation is unfortunate. Indeed, if cyclic collective preferences could be avoided, then it would often be quite clear how to aggregate voters' opinions. For example, in each election there would be a candidate (or, possibly, a group of candidates, if ties could happen) preferred by a majority of the voters to all the other ones, and equally preferred among themselves; such candidates are known as (weak) Condorcet winners.

One of the standard ways to avoid the Condorcet paradox is to restrict the domain of legal preference orders from all permutations of the candidates to some subset of such permutations. Indeed, a *Condorcet domain* is a set of allowed preference orders such that if one forms a preference profile by choosing orders only from this set, then there will certainly be a (weak) Condorcet winner; see, e.g., the overviews of Gaertner (2001) and Monjardet (2009), or the works of Clearwater, Puppe, and Slinko (2015) and Puppe and Slinko (2017). Two best-known examples of domain restrictions are the notions of *single-peakedness* (Black, 1958; Arrow, 1951) and *single-crossingness* (Mirrlees, 1971; Roberts, 1977) (formally, single-crossingness corresponds to a family of Condorcet domains). Under single-peaked preferences, we assume that the candidates can be arranged on a line, known as the *societal axis* (e.g., on the left-to-right political spectrum, or simply in the order of increasing numbers if we consider such an issue as, e.g., choosing the most comfortable temperature in a room). Then, a voter has single-peaked preferences if as we go along the societal axis, first this voter's appreciation for the candidates increases and then decreases.

On the other hand, a preference profile is single-crossing if the voters can be ordered in such a way that as we progress from one end of the voter spectrum to the other, then for each pair of candidates their relative order can change at most once. (The left-right political spectrum again provides an example: If the voters have views that can be arranged on the left-to-right spectrum, and the candidates can also be associated with such views, then for each two candidates a and b, such that a is more left-wing and b is more right-wing, the extreme-left voters certainly prefer a to b, extreme-right voters certainly prefer b to a, and as we progress from one extreme to the other, we expect only one swap.) Saporiti and Tohmé (2006) discuss several settings where single-crossing preferences arise in practice; the first motivating examples of Mirrlees (1971) and Roberts (1977) regarded taxation.

Not only is it known that the Condorcet paradox cannot occur if the voters have single-peaked or single-crossing preferences, but also many other negative effects cannot happen in these cases. For example, the famous impossibility theorems of Arrow (1951) and of Gibbard (1973) and Satterthwaite (1975) do not hold under these restrictions.[1]

In this paper we consider a far more recent domain restriction than either single-peakedness or single-crossingness, namely the notion of *top-monotonic* preferences of Barberà and Moreno (2011). Top-monotonicity has many advantages as, for example, it generalizes both the notions of single-peakedness and single-crossingness while still guaranteeing existence of (weak) Condorcet winners, and it is applicable to the settings where voters have weak preferences.[2] However, it also has drawbacks. One is that top-monotonic preferences are hard to define intuitively (and, indeed, this is why we refrain from describing them here on the intuitive level and point the reader to the formal definition in Section 2). Another one is that so far no polynomial-time algorithm for recognizing top-monotonic preferences was known, even though there is a number of algorithms for recognizing single-peaked preferences (Bartholdi & Trick, 1986; Escoffier, Lang, & Öztürk, 2008) and single-crossing preferences (Elkind, Faliszewski, & Slinko, 2012; Bredereck, Chen, & Woeginger, 2013; Cornaz, Galand, & Spanjaard, 2013). Our main contribution is providing the first polynomial-time algorithm for recognizing top-monotonic preference profiles. As there are natural domain restrictions for which the recognition problems are NP-hard (Peters, 2017), we believe that this contribution is imporant.

The first algorithmic study of top-monotonicity is due to Aziz (2014). While he did not give a recognition algorithm, he has shown that the problem of deciding if a profile of partial preference orders can be extended to a top-monotonic one is NP-hard; similar results were also obtained for single-peakedness (Lackner, 2014) and single-crossingness (Elkind et al., 2015). He also related the problem of recognizing top-monotonic profiles to the non-betweenness problem (Guttmann & Maucher, 2006). Our algorithm uses different ideas and is based on a somewhat intricate reduction to the SAT-2CNF problem (i.e., the problem of

1. Results of Arrow and of Gibbard and Satterthwaite are often interpreted as saying that in general no perfect voting rule exists. For single-peaked or single-crossing preferences, such a "perfect voting rule" simply elects the (weak) Condorcet winners.
2. Lackner (2014) and Elkind, Faliszewski, Lackner, and Obraztsova (2015) also apply notions of single-peakedness and single-crossingness to partial orders (including weak ones) and encounter some computational hardness results.

testing if a logical formula in conjunctive normal form with at most two literals per clause is satisfiable; SAT-2CNF is well known to be solvable in polynomial time; Krom, 1967). This is interesting both technically—indeed, noting that our approach can produce SAT-2CNF formulas requires some insight—and because many other methods for recognizing elections in restricted domains rely on the consecutive-ones problem[3]; see, e.g., the works of Peters and Lackner (2017) or Elkind and Lackner (2015). The SAT-2CNF problem might have a similar impact on the design of further algorithms recognizing domain restrictions and we recommend it as a useful tool for such tasks. As an example that our methodology indeed goes beyond top-monotonicity, after presenting our main result we show an analogous algorithm for recognizing single-peaked profiles.

We conclude by noting that there is yet another reason why having a polynomial time algorithm for recognizing top-monotonic preferences is important. Indeed, many NP-hard voting-related problems turn out to be polynomial-time solvable under various restricted domains. This was noted, e.g., by Conitzer (2009) for the case of vote elicitation, by Faliszewski, Hemaspaandra, Hemaspaandra, and Rothe (2011), Faliszewski, Hemaspaandra, and Hemaspaandra (2014) and by Magiera and Faliszewski (2017) for election control (i.e., for problems that model affecting the election result by changing its structure), by Brandt, Brill, Hemaspaandra, and Hemaspaandra (2015) for election bribery (i.e., for problems where we can change some number of votes to ensure a given candidate's victory), and—most importantly—by many researchers for the case of winner determination: As a few examples, we mention the results of Brandt et al. (2015) for all the Condorcet consistent rules, the results of Betzler, Slinko, and Uhlmann (2013), Cornaz, Galand, and Spanjaard (2012), and Skowron, Yu, Faliszewski, and Elkind (2015) for the Chamberlin–Courant rule, and the results of Peters (2018) for a number of multiwinner rules. Our work may inspire researchers to seek positive algorithmic consequences for top-monotonic preference profiles. Indeed, so far, the only such results are those that follow from the existence of a (weak) Condorcet winner.

The paper is organized as follows. In Section 2, we present necessary background regarding single-peaked, single-crossing, and top-monotonic profiles. Then, in Section 3, we show our algorithm for recognizing top-monotonic profiles and explain its workings. We first provide a convenient reformulation of the conditions for a profile to be top-monotonic, then show a variant of our algorithm for minimally rich profiles (where each alternative is ranked on top by at least one agent), and finally show the full algorithm. We finish the section by illustrating how our approach could be used for recognizing single-peaked profiles (while our algorithm is slower than the best algorithms for this task, it shows our ideas in a simpler setting). We conclude in Section 4.

2. Preliminaries

We mostly adopt the notation of Barberà and Moreno (2011). We let A be a (finite) set of alternatives (also called candidates) and we let $N = \{1, \dots, n\}$ be a set of n agents (also called voters). We denote the preference order of agent i over the set of alternatives by $\succcurlyeq_i$ (note that we take them to be *weak orders*). For each pair of alternatives $x, y \in A$, we write

3. In this problem, we are given a binary matrix and we ask if we can permute its rows so that in each column the entries with "ones" are consecutive (Booth & Lueker, 1976).

$x \succcurlyeq_i y$ if the i-th agent weakly prefers alternative x over y. We use the strict order ($\succ_i$) and equality ($\approx_i$) notations defined for each $x, y \in A$ as follows:

1. $x \succ_i y$ holds if $x \succcurlyeq_i y$ and it is not the case that $y \succcurlyeq_i x$;

2. $x \approx_i y$ holds if $x \succcurlyeq_i y$ and $y \succcurlyeq_i x$.

A preference profile is a collection of preference orders of the agents from N; we write $\succcurlyeq = (\succcurlyeq_1, \ldots, \succcurlyeq_n)$ to denote a preference profile of weak orders, and $\succ = (\succ_1, \ldots, \succ_n)$ for a profile of strict orders (i.e., one where there is no agent i and distinct candidates $x, y \in A$ such that $x \approx_i y$).

Before we define top-monotonic preferences, let us give formal definitions of single-peaked and single-crossing ones.

Definition 1 (Black, 1958; Arrow, 1951)**.** *A preference profile $\succ$ is single-peaked if there exists a linear order $>$ over the set of the alternatives (the societal axis), such that for each three alternatives x, y, and z, if it holds that either $x > y > z$ or $z > y > x$, then for each agent $i \in N$ we have that $x \succ_i y \implies y \succ_i z$.*

Intuitively put, the definition says that each agent i can choose his or her most preferred candidate arbitrarily, but then this agent must choose the following ones so that for each $j \in \{1, \ldots |A|\}$, the set of top j candidates according to i forms a consecutive block within the societal axis. In consequence, for each agent the candidate ranked last is either the maximum or the minimum element of $>$.

Definition 2 (Mirrlees, 1971; Roberts, 1977)**.** *A profile $\succ = (\succ_1, \ldots, \succ_n)$ is single-crossing with respect to an ordering of voters $(\succ_1, \ldots, \succ_n)$ if for each two alternatives x and y such that $x \succ_1 y$, there is a number $t_{x,y}$ such that $\{i \in N \mid x \succ_i y\} = \{1, \ldots, t_{x,y}\}$. Profile $\succ$ is single-crossing if it is single-crossing with respect to some ordering of the voters.*

That is, a profile is single-crossing if it is possible to order the voters so that, as we move along this order, the relative order of each two candidates changes at most once.

Example 1. *Consider candidate set $\{a, b, c, d\}$ and the profile of the following preference orders:*

$$a \succ_1 b \succ_1 c \succ_1 d, \qquad b \succ_2 a \succ_2 d \succ_2 c, \qquad d \succ_3 c \succ_3 b \succ_3 a.$$

It is single-crossing (for the natural order of the voters) but not single-peaked for any axis (it is well-known that in a single-peaked profile each candidate ranks one of the two extreme candidates from the societal axis last, but in our profile each of the three agents ranks a different candidate last).

For candidate set $\{a, b, c, d, e\}$, consider a profile with the following four preference orders:

$$c \succ_1 b \succ_1 d \succ_1 a \succ_1 e, \qquad c \succ_2 d \succ_2 b \succ_2 a \succ_2 e,$$
$$c \succ_3 b \succ_3 d \succ_3 e \succ_3 a, \qquad c \succ_4 d \succ_4 b \succ_4 e \succ_4 a.$$

It is single-peaked with respect to the axis $a > b > c > d > e$ (indeed, for each agent i and each $j \in \{1, \ldots, 5\}$, the top j candidates ranked by i form a consecutive block on this axis), *but it is not single-crossing (due to candidates a and e, a single-crossing ordering would have to put agent 1 next to agent 2 and agent 3 next to agent 4; due to candidates b and d, it would also have to put agent 1 next to agent 3, and agent 2 next to agent 4; fulfilling all these conditions simultaneously is impossible).*

We now move on to the definition of top-monotonic preferences (Barberà & Moreno, 2011). For all $i \in \mathbb{N}$ and for each $S \subseteq A$, we denote by $t_i(S)$ the set of top choices of the i-th agent among the alternatives from S. That is, $t_i(S) = \{x \in S \mid x \succcurlyeq_i y \text{ for all } y \in S\}$ and we call it the top of i in S according to $\succcurlyeq$. Let $T = \bigcup_{i \in N} t_i(A)$. For each preference profile $\succcurlyeq$, let $A(\succcurlyeq)$ be the family of sets containing A itself and all triples of distinct alternatives where each alternative is top in A for some agent $i \in N$ according to $\succcurlyeq$ (i.e., the triples in $A(\succcurlyeq)$ consist of the candidates from T, but $A(\succcurlyeq)$ also contains A).

Definition 3 (Barberà & Moreno, 2011). *A preference profile $\succcurlyeq$ is top-monotonic if there exists a linear order $>$ over the set of the alternatives, such that:*

(1) $t_i(A)$ is a finite union of closed intervals for all $i \in N$.[4]

(2) For all $S \in A(\succcurlyeq)$, for all $i, j \in N$ (including the case that $i = j$), all $x \in t_i(S)$, all $y \in t_j(S)$, and all $z \in S$, we have that:

$$[x > y > z \vee z > y > x] \implies \begin{cases} y \succcurlyeq_i z & \text{if } z \in t_i(S) \cup t_j(S) \\ y \succ_i z & \text{if } z \notin t_i(S) \cup t_j(S) \end{cases}$$

Definition 4. *A linear order $>$ over the set of alternatives is a* top-monotonic order *of a preference profile $\succcurlyeq$ if $\succcurlyeq$ is top-monotonic and $>$ fulfills condition (2) from Definition 3.*

Barberà and Moreno (2011) have shown that each single-peaked and each single-crossing profile is top-monotonic. On the other hand, the following example—taken from their work—shows that there is a profile of strict linear orders that is top-monotonic but neither single-peaked nor single-crossing.

Example 2 (Barberà & Moreno, 2011). *Consider candidate set $\{a, b, c, d\}$ and profile $\succ$ of three preference orders:*

$$a \succ_1 b \succ_1 c \succ_1 d, \qquad c \succ_2 d \succ_2 b \succ_2 a, \qquad d \succ_3 c \succ_3 a \succ_3 b.$$

Note that c is preferred to each other candidate by a majority of the agents. The profile is not single-peaked because there are three alternatives that are ranked last (a, b, and d). It is not single-crossing because agents 1 and 3 would have to be next to each other (because of the alternatives a and b), agents 2 and 3 would have to be next to each other (because of alternatives b and c), and agents 1 and 2 would have to be next to each other (because of alternatives c and d), which is impossible.

However, the profile is top-monotonic with respect to the order $a > b > c > d$. To see that the profile is top-monotonic, note that $A(\succ) = \{\{a, b, c, d\}, \{a, c, d\}\}$. We have to check

4. Since we assumed A to be a finite set, this part of the definition is always trivially satisfied. Barberà and Moreno consider also more general sets of alternatives and—to indicate the generality of their definition—we decided to keep this requirement in the text.

each $S \in A(\succ)$ and each two agents i and j. Let us take $S = \{a, c, d\}$, $i = 1$ and, $j = 2$. We have $x \in t_1(S) = \{a\}$, $y \in t_2(S) = \{c\}$, and we take $z = d$. It holds that $a > c > d$ and $z = d \notin t_1(S) \cup t_2(S) = \{a, c\}$, so it is required that $c \succ_1 d$, and this indeed holds. Checking top-monotonicity of the profile using the definition would require checking all the remaining combinations of S, i, j, and z.

We often need to refer to various definitions, lemmas, and conditions for particular instantiations of the object that they refer to. In such cases, we write $x \to y$ to mean that y takes the role of x. For example, if we wanted to speak of the definition of top-monotonic profiles for some specific agents k and ℓ taking the roles of agents i and j, we would indicate this by writing $i \to k$ and $j \to \ell$.

3. Results

The problem of determining a top-monotonic order of a preference profile is as follows: Given a finite set of alternatives A, a finite set of agents N, and a preference profile $\succcurlyeq$ (for the agents in N) over A, find a top-monotonic order of $\succcurlyeq$ or decide that no such order exists. In this section we provide an algorithm for this problem.

3.1 Interface to Top-Monotonicity

The following definition, and Lemma 1 a bit later, constitute an interface between the notion of top-monotonicity and our main algorithm, which reconstructs the top-monotonic order from allowed orders over triples of candidates. We follow the notation from Section 2; specifically, recall that $t_i(S)$ refers to the set of alternatives that are top of i in S and that T is the set of alternatives that are ever ranked on top by some agent.

Definition 5. *For each triple of distinct alternatives $S = \{x, y, z\} \subseteq A$ and each pair of agents $i, j \in N$ (including the case that $i = j$), we define* the set of legal orderings*, denoted as $L_S^{i,j}$, to be the set of ordered sequences of alternatives x, y, z, such that for each sequence $\sigma = (\sigma_1, \sigma_2, \sigma_3)$, where $\{\sigma_1, \sigma_2, \sigma_3\} = S$, all the following criteria are met:*

(1)

$$[\sigma_1 \in t_i(S) \text{ and } \sigma_2 \in t_j(S)] \implies \begin{cases} \sigma_2 \succcurlyeq_i \sigma_3 & \text{if } \sigma_3 \in t_i(S) \cup t_j(S) \\ \sigma_2 \succ_i \sigma_3 & \text{if } \sigma_3 \notin t_i(S) \cup t_j(S) \end{cases}$$

(2)

$$[\sigma_1 \in t_j(S) \text{ and } \sigma_2 \in t_i(S)] \implies \begin{cases} \sigma_2 \succcurlyeq_j \sigma_3 & \text{if } \sigma_3 \in t_i(S) \cup t_j(S) \\ \sigma_2 \succ_j \sigma_3 & \text{if } \sigma_3 \notin t_i(S) \cup t_j(S) \end{cases}$$

(3)

$$[\sigma_3 \in t_i(S) \text{ and } \sigma_2 \in t_j(S)] \implies \begin{cases} \sigma_2 \succcurlyeq_i \sigma_1 & \text{if } \sigma_1 \in t_i(S) \cup t_j(S) \\ \sigma_2 \succ_i \sigma_1 & \text{if } \sigma_1 \notin t_i(S) \cup t_j(S) \end{cases}$$

(4)

$$[\sigma_3 \in t_j(S) \text{ and } \sigma_2 \in t_i(S)] \implies \begin{cases} \sigma_2 \succcurlyeq_j \sigma_1 & \text{if } \sigma_1 \in t_i(S) \cup t_j(S) \\ \sigma_2 \succ_j \sigma_1 & \text{if } \sigma_1 \notin t_i(S) \cup t_j(S) \end{cases}$$

(5)

$$\left.\begin{array}{c}[\sigma_1 \in t_i(S) \ \textit{and}\ \sigma_3 \succ_i \sigma_2] \\ \vee \\ [\sigma_1 \in t_j(S) \ \textit{and}\ \sigma_3 \succ_j \sigma_2] \\ \vee \\ [\sigma_3 \in t_i(S) \ \textit{and}\ \sigma_1 \succ_i \sigma_2] \\ \vee \\ [\sigma_3 \in t_j(S) \ \textit{and}\ \sigma_1 \succ_j \sigma_2]\end{array}\right\} \implies \sigma_2 \notin T$$

To illustrate how the set of legal orderings is constructed, let us consider the following example. We take $S = \{x, y, z\}$ and two agents i and j with preference orders:

$$x \succ_i y \succ_i z \text{ and } z \succ_j x \succ_j y.$$

Now we need to consider six different orderings of candidates x, y, z. Let us consider ordering $\sigma = (x, z, y)$. We can see that it does not satisfy condition (1) as $x \in t_i(S)$, $z \in t_j(S)$ but $z \succ_i y$ is not true. Similarly, if we consider ordering $\sigma' = (x, y, z)$, and if we assume that $x, y, z \in T$, then we can see that it does not satisfy condition (5) as $z \in t_j(S)$ and $x \succ_j y$ (so the left-hand side of the implication is true) but $y \in T$. On the other hand, if we consider ordering $\sigma'' = (z, x, y)$ then we can see that it satisfies all five conditions and therefore is going to be a part of the set of legal orderings $L_S^{i,j}$. By analyzing the remaining three possible orderings we get that the complete set of legal orderings for the setup under consideration is $L_S^{i,j} = \{(z, x, y), (y, x, z)\}$.

Let Q^T be a family of *sets of legal orderings* for every triple x, y, z such that $x, y, z \in T$ and for every $i, j \in N$. Let Q^{NT} be another family of *sets of legal orderings*, for every triple x, y, z, where $x, y \in T$ and $z \in (A \setminus T)$, and every $i, j \in N$ such that $x \in t_i(A)$ and $y \in t_j(A)$. Note that for both definitions we allow for agents i and j to be the same. In other words, family Q^T regards sets of legal orderings for all agents and all triples of candidates that appear on top of some preference orders, whereas Q^{NT} is defined analogously, but for triples of candidates that contain two candidates from tops of some preference orders and one candidate that never appears on top (of any preference order).

Lemma 1. *Let A be a set of alternatives, N be a set of agents, and $\succcurlyeq$ be a preference profile over A. Let $Q = Q^T \cup Q^{NT}$. Preference profile $\succcurlyeq$ is top-monotonic with $>$ as the top-monotonic order if and only if for each set $X \in Q$, there exists an element $\sigma = (\sigma_1, \sigma_2, \sigma_3) \in X$ such that $\sigma_1 > \sigma_2 > \sigma_3$.*

Proof. We first focus on proving that if top-monotonic order $>$ exists for a given preference profile $\succcurlyeq$, then for each set X from Q there exists an element $\sigma = (\sigma_1, \sigma_2, \sigma_3)$ such that $\sigma_1 > \sigma_2 > \sigma_3$. Let us assume that preference profile $\succcurlyeq$ has a top-monotonic order $>$, but for some set $X' \in Q$ there is no such element $\sigma' = (\sigma'_1, \sigma'_2, \sigma'_3) \in X'$ such that $\sigma'_1 > \sigma'_2 > \sigma'_3$. Set X' may be a part of Q^T or Q^{NT}. Let us first assume that $X' \in Q^T$ and, so, each element $\sigma' \in X'$ is a triple of elements $x, y, z \in T$. We also assume that X' corresponds to a pair of agents $k, \ell \in N$. Without loss of generality we assume that $x > y > z$. Now, as per our assumption, sequence (x, y, z) is not a part of X', which means that one of the conditions from Definition 5 is not met for it. With set $S = \{x, y, z\}$ we consider each of the conditions from Definition 5:

(a) Since all x, y, z belong to T, condition (5) is not satisfied when the left-hand side of the implication is true. When $x \in t_k(S)$ and $z \succ_k y$, we get a contradiction with the assumption that $\succcurlyeq$ is top-monotonic, because then condition (2) from Definition 3 is violated for $i \rightarrow k$ and for some j such that $y \in t_j(S)$ (we know that such an agent j exists as $y \in T$). Since all the clauses from the left-hand side of condition (5) are symmetric up to the exchange of i with j and σ_1 with σ_3, we see that condition (5) is always satisfied for the sequence (x, y, z), assuming $\succcurlyeq$ is top-monotonic.

(b) If condition (1) is not met, then for $S = \{x, y, z\}$ we have:

$$x \in t_k(S) \text{ and } z \succ_k y$$
$$\textit{or}$$
$$x \in t_k(S) \text{ and } y \in t_\ell(S) \text{ and } z \notin t_k(S) \cup t_\ell(S) \text{ and } z \succcurlyeq_k y$$

However if that were true, $>$ could not be a top-monotonic order of the profile $\succcurlyeq$. The reason is that it directly violates condition (2) from Definition 3 (for $i \leftarrow k$ and $j \leftarrow \ell$). We therefore get a contradiction and condition (1) has to be satisfied for the selected sequence (x, y, z).

(c) Condition (2) is symmetric to the condition (1) with i and j swapped, and therefore it can be shown in a similar way that it has to be satisfied for a selected sequence (x, y, z). The same applies to condition (3) that is symmetric to condition (1) with σ_1 and σ_3 swapped. Lastly, we can apply the same methodology to condition (4), which can be constructed by swapping both i with j and σ_1 with σ_3 from condition (1).

In consequence, we see that if $X' \in Q^T$, then (x, y, z) has to be a part of X', which stands against our assumption that (x, y, z) is not to a part of X'. We therefore get that $X' \in Q^{NT}$. Let us assume that X' corresponds to agents $k, \ell \in N$ and a triple of alternatives x, y, w such that $x \in t_k(A)$, $y \in t_\ell(A)$ and $w \in A \setminus T$. Now we need to consider six possible cases for how agents x, y, w are ordered under $>$:

$$x > y > w, \quad y > x > w, \quad w > x > y, \quad w > y > x, \quad x > w > y, \quad y > w > x.$$

These six options map to the following sequences:

$$(x, y, w), \quad (y, x, w), \quad (w, x, y), \quad (w, y, x), \quad (x, w, y), \quad (y, w, x).$$

We set $S = \{x, y, w\}$. We note that $w \notin t_k(S)$ which stands true because x is top of agent k and we know that w is not top of any agent. Similarly, we note that $w \notin t_\ell(S)$. We make the following observations:

(I) If $x > w > y$ or $y > w > x$, we note that sequence $(x, w, y) \in X'$ or $(y, w, x) \in X'$ respectively. This is so because since $w \notin T$, condition (5) is satisfied. Also, all the remaining conditions from (1) to (4) are satisfied because $w \notin t_k(S)$ and $w \notin t_\ell(S)$.

(II) When $x > y > w$, we see that for the corresponding sequence (x, y, w) both conditions (3) and (4) are satisfied, because $w \notin t_k(S)$ and $w \notin t_\ell(S)$. Clearly, condition (2) is also satisfied because $y \succ_k w$ (which comes from the fact that $y \in t_k(A)$ and $w \notin t_k(A)$). Additionally, condition (1) is not satisfied if and only if $w \succcurlyeq_k y$ and condition (5) cannot be satisfied if and only if $w \succ_k y$. However, since $>$ is a top-monotonic order of the profile $\succcurlyeq$, from condition (2) in Definition 3 given $i \to k$, $j \to \ell$ and $S \to A$ (clearly $x \in t_i(S)$ and $y \in t_j(S)$ and $w \in S$), we get that if $x > y > w$, then it has to be that $y \succ_i w$ (because w is not a top for any agent). Therefore, we get a contradiction with both the assumption that $w \succ_k y$ (corresponding to condition (1)) and the assumption that $w \succcurlyeq_k y$ (corresponding to condition (5)). It means that conditions (1) and (5) are both satisfied, and therefore if $x > y > w$ then $(x, y, w) \in X'$.

(III) When $y > x > w$, we again use the fact that the rules from Definition 5 are symmetric up to the exchange of i and j, which—based on the above—proves that if $y > x > w$ then $(y, x, w) \in X'$. We follow the same methodology for the remaining orderings, $w > x > y$ and $w > y > x$, that are symmetric up to the exchange of σ_1 and σ_3.

We therefore see that it is impossible to find a set X' that would violate the assumption from Lemma 1 when $\succcurlyeq$ is top-monotonic.

The final step of this proof is to show that Lemma 1 is also true in the opposite direction. That is, when there exists a linear order $>'$ such that for each $X \in Q$ there exists an element $\sigma = (\sigma_1, \sigma_2, \sigma_3) \in X$, such that $\sigma_1 >' \sigma_2 >' \sigma_3$ then $\succcurlyeq$ is top-monotonic and that $>'$ is a top-monotonic order over $\succcurlyeq$. We assume that $>'$ exists and that $\succcurlyeq$ is not top-monotonic, and we will reach a contradiction, showing that $>'$ satisfies all the requirements to be a top-monotonic order over $\succcurlyeq$. If, as per our assumption, $\succcurlyeq$ is not top-monotonic then there exists a set $S \in A(\succcurlyeq)$, a pair of agents $i, j \in N$, and three (distinct) alternatives x, y, z, $x \in t_i(S)$, $y \in t_j(S)$ and $z \in S$, such that:

$$(x >' y >' z \vee z >' y >' x) \text{ and } (z \succ_i y \vee (z \approx_i y \text{ and } z \notin t_i(S) \cup t_j(S))). \tag{1}$$

Let $S' = \{x, y, z\}$. We note that $z \in t_i(S) \iff z \in t_i(S')$ because $S' \subseteq S$. Similarly, we see that $z \in t_j(S) \iff z \in t_j(S')$. If Eq. (1) is satisfied then there exists a set $X'' \in Q$ that contains either sequence (x, y, z) or (z, y, x) (indeed, this follows from the fact that either $x >' y >' z$ or $z >' y >' x$ holds). It is easy to note that when $z \in T$ then $X'' \in Q^T$ and when $z \notin T$ then $X'' \in Q^{NT}$. We now consider two cases depending on whether (x, y, z) or (z, y, x) is in X''.

(I) Let us first assume $(x, y, z) \in X''$. Now, from condition (1) of Definition 5, we see that if $z \notin t_i(S') \cup t_j(S')$, then $y \succ_i z$, which makes it impossible to satisfy Eq. (1) (the second segment of the equation requires that $z \succcurlyeq_i y$). On the other hand, when $z \in t_i(S') \cup t_j(S')$, then from condition (1) we get that $y \succcurlyeq_i z$, which again makes it impossible for Eq. (1) to be satisfied (here we refer to the fact that if $z \in t_i(S') \cup t_j(S')$ then $z \in t_i(S) \cup t_j(S)$).

(II) When $(z, y, x) \in X''$, then we follow the same methodology, leveraging the symmetry of Definition 5 with respect to the exchange of σ_1 with σ_2 (instead of condition (1) we use condition (3)).

We see that it is impossible for Eq. (1) to be satisfied for each of the aforementioned combinations of agents and alternatives, assuming that $>'$ exists. Therefore $>'$ is a top-monotonic order for $\succcurlyeq$.

□

The greatest advantage of using Lemma 1 and Definition 5 over directly applying Definition 3 is that it allows us to focus on sets of three candidates only (Definition 3 also uses the whole set A, as $A \in \mathcal{A}(\succcurlyeq)$). This property is crucial for our technique.

3.2 The Case of Minimally Rich Profiles

Before we get to our main theorem, we present a proof for a simpler variant of it. The full proof, presented in the next section, uses a very similar approach.

Preference profiles where each alternative is ranked first by at least one agent are known as *minimally rich*. We extend this definition to the case of weak orders in a natural way: A profile is minimally rich if every alternative is top of some agent. The notion of minimally rich profiles is very similar to that of narcissistic ones, introduced by Bartholdi and Trick (1986); the difference is that in the case of narcissistic profiles it is assumed that the sets of alternatives and voters are equal and each voter ranks him or herself first.[5]

Below we show an algorithm for recognizing minimally rich top-monotonic profiles. The main idea is to solve a 2CNF formula that encodes an order over the alternatives such that for each triple of alternatives and each two agents this order contains at least one of the legal orderings for these alternatives and voters. The case of minimally rich profiles is simpler than the general one because the family of sets of legal orderings is limited to Q^T (so one does not need to consider Q^{NT}) and it is easier to argue that the satisfying truth assignments encode orders over the alternatives (in particular, that these assignments encode transitive relations).

Theorem 2. *Let A be a set of alternatives, N be a set of agents, and $\succcurlyeq$ be a minimally rich preference profile. The problem of determining if a top-monotonic order of $\succcurlyeq$ exists (and computing it) is polynomial-time solvable.*

Proof. Due to Lemma 1, it suffices to demonstrate that finding an order $>$ over A, such that for each $X \in Q$ there exists $\sigma = (\sigma_1, \sigma_2, \sigma_3) \in X$ such that $\sigma_1 > \sigma_2 > \sigma_3$, can be done in polynomial time. From now on we focus on this task.

Since each of the alternatives is a top for some $i \in N$ (with respect to A), we have $T = A$ and $Q = Q^T$. Let us now consider some set of orderings from Q^T. As they all correspond to triples of alternatives from T, there are only a few possible cases of how they may relate to each other in preference orders of pairs of agents. Let us take some three alternatives $x, y, z \in T$; there are 21 different combinations of pairs of preference orders that we need to consider (see second column of Table 1). All these combinations have their entries in Table 1, precomputed according to Definition 5. To obtain a set of legal orderings for some triple of candidates S and some agents i and j, it suffices to assign these candidates to variables x, y, z and choose an entry in Table 1 corresponding to the preference orders of agents i and j.

5. In an early version of the paper we incorrectly referred to minimally rich profiles as narcissistic ones.

	Comb. of agents $i, j \in N$	Set of legal orderings	2CNF ordering formula
1	$x \succ_i y \succ_i z$ and $x \succ_j y \succ_j z$	$\{(y,x,z),(x,y,z),(z,x,y),(z,y,x)\}$	$(xz \vee zy) \wedge (yz \vee zx)$
2	$x \succ_i y \succ_i z$ and $x \succ_j y \approx_j z$	$\{(y,x,z),(x,y,z),(z,x,y),(z,y,x)\}$	$(xz \vee zy) \wedge (yz \vee zx)$
3	$x \succ_i y \succ_i z$ and $x \approx_j y \succ_j z$	$\{(y,x,z),(x,y,z),(z,x,y),(z,y,x)\}$	$(xz \vee zy) \wedge (yz \vee zx)$
4	$x \succ_i y \succ_i z$ and $x \approx_j y \approx_j z$	$\{(y,x,z),(x,y,z),(z,x,y),(z,y,x)\}$	$(xz \vee zy) \wedge (yz \vee zx)$
5	$x \succ_i y \succ_i z$ and $x \succ_j z \succ_j y$	$\{(y,x,z),(z,x,y)\}$	$(xy \vee xz) \wedge (yz \vee zx) \wedge (yx \vee zy)$
6	$x \succ_i z \succ_i y$ and $x \approx_j y \succ_j z$	$\{(y,x,z),(z,x,y)\}$	$(xy \vee xz) \wedge (yz \vee zx) \wedge (yx \vee zy)$
7	$x \succ_i y \succ_i z$ and $y \succ_j x \succ_j z$	$\{(y,x,z),(x,y,z),(z,x,y),(z,y,x)\}$	$(xz \vee zy) \wedge (yz \vee zx)$
8	$y \succ_i x \succ_i z$ and $x \succ_j y \approx_j z$	$\{(y,x,z),(z,x,y)\}$	$(xy \vee xz) \wedge (yz \vee zx) \wedge (yx \vee zy)$
9	$x \succ_i y \succ_i z$ and $y \succ_j z \succ_j x$	$\{(x,y,z),(z,y,x)\}$	$(xy \vee zx) \wedge (xz \vee zy) \wedge (yz \vee yx)$
10	$z \succ_i x \succ_i y$ and $x \approx_j y \succ_j z$	$\{(y,x,z),(z,x,y)\}$	$(xy \vee xz) \wedge (yz \vee zx) \wedge (yx \vee zy)$
11	$y \succ_i z \succ_i x$ and $x \succ_j y \approx_j z$	$\{(y,z,x),(x,z,y)\}$	$(xy \vee yz) \wedge (xz \vee yx) \wedge (zx \vee zy)$
12	$x \succ_i y \succ_i z$ and $z \succ_j y \succ_j x$	$\{(x,y,z),(z,y,x)\}$	$(xy \vee zx) \wedge (xz \vee zy) \wedge (yz \vee yx)$
13	$x \succ_i y \approx_i z$ and $x \approx_j y \succ_j z$	$\{(y,x,z),(z,x,y)\}$	$(xy \vee xz) \wedge (yz \vee zx) \wedge (yx \vee zy)$
14	$x \succ_i y \approx_i z$ and $y \succ_j x \approx_j z$	$\{(x,z,y),(y,z,x)\}$	$(xy \vee yz) \wedge (xz \vee yx) \wedge (zx \vee zy)$
15	$z \succ_i x \approx_i y$ and $x \approx_j y \succ_j z$	$\{(y,x,z),(x,y,z),(z,x,y),(z,y,x)\}$	$(xz \vee zy) \wedge (yz \vee zx)$
16	$x \approx_i y \succ_i z$ and $x \approx_j y \succ_j z$	$\{(y,x,z),(x,y,z),(z,x,y),(z,y,x)\}$	$(xz \vee zy) \wedge (yz \vee zx)$
17	$x \approx_i y \succ_i z$ and $x \approx_j y \approx_j z$	$\{(y,x,z),(x,y,z),(z,x,y),(z,y,x)\}$	$(xz \vee zy) \wedge (yz \vee zx)$
18	$x \approx_i y \succ_i z$ and $x \approx_j z \succ_j y$	$\{(y,x,z),(z,x,y)\}$	$(xy \vee xz) \wedge (yz \vee zx) \wedge (yx \vee zy)$
19	$x \succ_i y \approx_i z$ and $x \succ_j y \approx_j z$	all permutations of $\{x,y,z\}$	n/a
20	$x \approx_i y \approx_i z$ and $x \succ_j y \approx_j z$	all permutations of $\{x,y,z\}$	n/a
21	$x \approx_i y \approx_i z$ and $x \approx_j y \approx_j z$	all permutations of $\{x,y,z\}$	n/a

Table 1: All possible combinations of pairs of agents from Q^T, with corresponding sets of legal orderings (third column). The last column shows a 2CNF representation of each ordering formula (note that we write, e.g., yx instead of $\neg xy$). Note that for rules 19–21 all permutations are allowed, but we do need to ensure that literals xy, xz, and yz indeed encode a permutation (that is, only six out of their eight possible truth assignments are legal; this requirement cannot be encoded using a 2CNF formula and we deal with it differently).

With Table 1 available, we can compute the set Q^T by looking up appropriate values. We illustrate this process with the example below.

Example 3. *Let the set of alternatives be $A' = \{a, b, c, d\}$ and let the preference profile $\succcurlyeq'$ be as follows ($N' = \{1, 2\}$):*

$$a \approx'_1 b \approx'_1 c \succ'_1 d, \quad c \approx'_2 d \succ'_2 a \succ'_2 b.$$

We see that $T = \{a, b, c, d\} = A'$, as each of the alternatives is top for some agent. We have four possible triples of alternatives and three possible pairs of agents (note that we can make a pair that consists of two copies of the same agent). Let us start with triple $\{a, b, c\}$ and agents 1 and 2. We get the following relation between alternatives from this triple: $a \approx'_1 b \approx'_1 c$ and $c \succ'_2 a \succ'_2 b$, which matches rule no. 4 from Table 1 (where $x \leftarrow c$, $y \leftarrow a$ and $z \leftarrow b$) and generates the following set of legal orderings:

$$\{(a, c, b), (c, a, b), (b, c, a), (b, a, c)\}.$$

Similarly, if we now take triple $\{a, b, d\}$ and agents 1 and 2, the relation looks as follows: $a \approx'_1 b \succ'_1 d$ and $d \succ'_2 a \succ'_2 b$, which matches rule no. 10 (with $x \leftarrow a$, $y \leftarrow b$ and $z \leftarrow d$).

Therefore it generates the set of legal orderings $\{(b,a,d),(d,a,b)\}$. *If we follow similar steps for triples* $\{a,c,d\}$ *and* $\{b,c,d\}$, *we will match rule no. 14 for the former and rule no. 6 for the latter, generating the two corresponding sets of legal orderings* ($\{(a,c,d),(d,c,a)\}$ *and* $\{(b,c,d),(d,c,b)\}$). *On top of that, we have to consider pairs that are made of two copies of the same agent (that is* 1 *and* 1; 2 *and* 2). *As a result, family* Q *will have* 12 *elements and will be:*

$$\begin{aligned}
Q = \{&\{(a,c,b),(c,a,b),(b,c,a),\underline{(b,a,c)}\},\\
&\{\underline{(b,a,d)},(d,a,b)\},\\
&\{\underline{(a,c,d)},(d,c,a)\},\\
&\{\underline{(b,c,d)},(d,c,b)\},\\
&\{\underline{(b,a,c)},(b,c,a),(c,a,b),(a,c,b),(c,b,a),(a,b,c)\},\\
&\{\underline{(b,a,d)},(a,b,d),(d,a,b),(d,b,a)\},\\
&\{(c,a,d),\underline{(a,c,d)},(d,a,c),(d,c,a)\},\\
&\{(c,b,d),\underline{(b,c,d)},(d,b,c),(d,b,a)\},\\
&\{(c,b,a),(c,a,b),(a,b,c),\underline{(b,a,c)}\},\\
&\{(d,b,a),(d,a,b),(d,b,c),\underline{(b,a,d)}\},\\
&\{(d,c,a),(d,a,c),\underline{(a,c,d)},(c,a,d),(a,d,c),(c,d,a)\},\\
&\{\underline{(b,c,d)},(c,b,d),(d,c,b),(d,b,c)\}\}.
\end{aligned}$$

Taking order $b >' a >' c >' d$, *we see that for each set* $X \in Q$ *there is at least one sequence* $\sigma = (\sigma_1, \sigma_2, \sigma_3) \in X$ *such that* $\sigma_1 >' \sigma_2 >' \sigma_3$ *(corresponding items for each set are underlined on the listing above). By Lemma 1, we conclude that* $\succcurlyeq'$ *is top-monotonic and* $>'$ *is its top-monotonic order.*

Set Q consists of $|N|^2 \times \binom{|A|}{3}$ elements, where each element is of the form of one of the sets from the third column of Table 1. We want to find a linear order $>$ over the set of alternatives such that for each $X \in Q$ we can find at least one sequence $\sigma = (\sigma_1, \sigma_2, \sigma_3) \in X$ with $\sigma_1 > \sigma_2 > \sigma_3$. It turns out that we can express this problem as an instance of the SAT-2CNF problem.

To illustrate this approach, let us consider rule no. 5 from Table 1, with output $\{(y,x,z), (z,x,y)\}$. If the desired order $>$ exists, it has to satisfy condition:

$$(y > x > z) \vee (z > x > y). \tag{2}$$

We can create a similar formula for each set of legal orderings from Q, and then expect the order $>$ (if it exists) to satisfy the conjunction of these formulas.

The crucial observation is that Eq. (2) (as well as formulas corresponding to all the other rules from Table 1) can be expressed in conjunctive normal form with two literals per clause (2CNF). To this end, we take our logical variables to be xy, xz, and yz and we interpret them as representing appropriate relations under order $>$ (e.g., xy is true if $x > y$ holds; we also write, e.g., yx as an abbreviation for $\neg xy$). Then, Eq. (2) can be equivalently expressed as:

$$(yx \wedge xz \wedge yz) \vee (zx \wedge xy \wedge zy)$$

or, equivalently, as:

$$(\neg xy \wedge xz \wedge yz) \vee (\neg xz \wedge xy \wedge \neg yz).$$

This formula, on the other hand, is true if and only if the following one is (see comments below):

$$(xz \vee xy) \wedge (yz \vee \neg xz) \wedge (\neg xy \vee \neg yz). \tag{3}$$

To check that the two formulas above are indeed equivalent, one may try all possible truth assignments for our variables (as there are three variables only, we need to consider eight possible assignments). For example, if we take the following one:

$$xy \leftarrow True, \quad xz \leftarrow True, \quad \text{and} \quad yz \leftarrow False$$

then both formulas evaluate to $False$, whereas if we take:

$$xy \leftarrow False, \quad xz \leftarrow True, \quad \text{and} \quad yz \leftarrow True$$

then they both evaluate to $True$. Formula (3) is in the 2CNF form and, in fact, we can represent each of the possible sets of legal orderings using 2CNF formulas, as presented in Table 1 (the only exception is that rules 19, 20, and 21 do not impose any restrictions on the orderings and, thus, do not generate formulas at all; yet, we do need to ensure that relevant literals encode permutations, but we show how to deal with this issue later).[6]

To summarize, we proceed as follows. For each three alternatives x, y, and z and each two agents i and j, we look-up the SAT-2CNF formula in Table 1 that corresponds to this setting (for each pair of alternatives p and q, we arbitrarily choose which of the literals pq or qp that arises is represented as a variable and which is represented as this variable's negation; note that if some literal does not arise ever in any of the rules, we do not create its variable). We form the *global ordering formula* by taking the conjunction of all these formulas. We now show that a top-monotonic order exists if and only if the global ordering formula is satisfiable.

Observation 1. *A top-monotonic order for $\succcurlyeq$ exists if and only if there is an assignment for the variables that satisfies the global ordering formula.*

To prove this result, we first show that if $\succcurlyeq$ has a top-monotonic order then the global ordering formula is satisfiable. Let $>'$ be this top-monotonic order. Now consider a variable assignment for the global ordering formula to be such that for each two candidates p and q, we set the literal $pq = 1$ if $p >' q$ (note that it may, in fact, mean setting variable qp to $False$, depending which one of pq and qp is used as a variable in the global ordering formula and which one is represented as its negation). It is easy to see that such a variable assignment is a valid solution for the global ordering formula.

6. The reader may wonder how we derived Formula (3) from Formula (2), or how we deduced that it can be translated to an equivalent 2CNF form. One way to look at this is by noting that the clauses in 2CNF formulas can be seen as implications (using the fact that for literals p and q we have that $p \vee q$ is equivalent to $\neg p \implies q$). The formula $(xz \vee xy) \wedge (yz \vee \neg xz)$, which is a part of our example formula, can be expressed as $(\neg xy \implies xz) \wedge (xz \implies yz)$. This, in turn, we interpret as $(y > x \implies x > z) \wedge (x > z \implies y > z)$. As a consequence, a satisfying truth assignment that encodes $y > x$ must, in fact, encode $y > x > z$. All the other cases can be analyzed similarly.

Now it remains to show that if there is a satisfying assignment for the variables of the global ordering formula, then the profile is top-monotonic. Let us assume that the global ordering formula has a satisfying assignment and let relation $>'$ be defined for each pair of alternatives $p, q \in A$ as follows: We set $p >' q$ exactly if the literal pq evaluates to $True$. By definition, relation $>'$ satisfies the global ordering formula and we only need to show that it, indeed, is an order over A. We show that this is the case by considering the three requirements that a strict order must satisfy:

(a) Relation $>'$ is *irreflexive* because we do not have literals of the form xx in our global ordering formula.

(b) Relation $>'$ is *asymmetric* because for each $x, y \in A$, if $x >' y$ then it is not the case that $y >' x$ because for each $x, y \in A$, $xy \equiv \neg yx$ and, so, it cannot be that both xy and yx are set to 1 at the same time.

(c) Relation $>'$ is *transitive*, that is, for every triple $x, y, z \in A$ if $x >' y$ and $y >' z$ then $x >' z$. This follows from the fact that for every triple $x, y, z \in A$ we need to satisfy a formula that corresponds to a set of legal orderings. It is clear that any element from the set of legal orderings has to satisfy the transitivity condition (as such an element represents an order of alternatives). The only exception is when a triple $x, y, z \in A$ matches either of the rules 19, 20, or 21 from Table 1 for every possible pair of agents $i, j \in N$. However, in such a case it is easy to see that it must be the case that $y \approx_k z$ for every $k \in N$. Therefore we can ignore this case as alternatives y and z are indistinguishable form each other for all the agents. We pick one of them to use in the algorithm (and remove the other one from the profile); if it turns out that a top-monotonic order exists, then we place these candidates next to each other in this order (the algorithm computes the position of one of them in the top-monotonic order, and the other one can be put on either of its sides).

So far, we have not argued that $>'$ is a total order and, indeed, it may be partial. We let $>^*$ be a linear extension of $>'$; we know that such an extension exists due to the order-extension principle. Since $>^*$ satisfies the global ordering formula (as it is an extension of $>'$) and it is a linear order, $>^*$ is a top-monotonic order for $\succcurlyeq$.

Finally, we note that, since the global ordering formula is in conjunctive normal form with at most two variables per clause, there is a simple polynomial-time algorithm that checks if it is satisfiable and, if so, produces a satisfying assignment. Further, the formula itself is of length polynomially bounded in the number of candidates and agents (we need $O(N^2 \cdot |A|^3)$ subformulas from Table 1, with at most $O(|A|^2)$ variables). $\square$

Example 4. *Let us consider the same setting as in Example 3, that is, let $A' = \{a, b, c, d\}$ be a set of alternatives and let $\succcurlyeq'$ be a preference profile defined as follows: $N' = \{(a \approx'_1 b \approx'_1 c \succ'_1 d), (c \approx'_2 d \succ'_2 a \succ'_2 b)\}$. Based on the family Q obtained in Example 3, we compute the conjunction of 2-CNF ordering formulas which looks as follows (we removed duplicate clauses):*

$$(ab \vee bc) \wedge (ba \vee cb) \wedge (ac \vee da) \wedge (ad \vee dc) \wedge (cd \vee ca) \wedge (ab \vee ad) \wedge$$
$$(bd \vee da) \wedge (ba \vee db) \wedge (bc \vee db) \wedge (bd \vee dc) \wedge (cd \vee cb) \wedge (ad \vee db) \wedge$$

$$(cd \vee da) \wedge (cd \vee db) \wedge (ab \vee bc) \wedge (ab \vee bd) \wedge (cb \vee bd)$$

Now we create an instance of SAT-2CNF problem with six variables, ab, ac, ad, bc, bd, and cd:

$$(ab \vee \underline{bc}) \wedge (\underline{\neg ab} \vee \neg bc) \wedge (\underline{ac} \vee \neg ad) \wedge (\underline{ad} \vee \neg cd) \wedge (\underline{cd} \vee \neg ac) \wedge (ab \vee \underline{ad}) \wedge$$
$$(\underline{bd} \vee \neg ad) \wedge (\underline{\neg ab} \vee \neg bd) \wedge (\underline{bc} \vee \neg bd) \wedge (\underline{bd} \vee \neg cd) \wedge (\underline{cd} \vee \neg bc) \wedge (\underline{ad} \vee \neg bd) \wedge$$
$$(\underline{cd} \vee \neg ad) \wedge (\underline{cd} \vee \neg bd) \wedge (ab \vee \underline{bc}) \wedge (ab \vee \underline{bd}) \wedge (\neg bc \vee \underline{bd})$$

We note that the above SAT-2CNF instance is satisfiable and one of the possible assignments is $ab \leftarrow False$, $ac \leftarrow True$, $ad \leftarrow True$, $bc \leftarrow True$, $bd \leftarrow True$, $cd \leftarrow True$. *Literals that are true according to this assignment are underlined. We see that each clause has at least one literal that is true. Based on the assignment we now get a strict partial order* $>'$ *defined as follows:* $>' = \{(b >' a), (a >' c), (a >' d), (b >' c), (b >' d), (c >' d)\}$, *which happens to also be a linear order over* A'. *Thus the computed order is* $b >' a >' c >' d$. *As* $>'$ *satisfies our global ordering formula, we conclude that it is a top-monotonic order for* $\succcurlyeq'$.

3.3 Main Proof

Our main result holds without the assumption that the profile is minimally rich. The full proof is more complicated due to the fact that there are more rules to be considered, with more cases where it is not clear if the rules indeed lead to a strict linear order. Fortunately, all these obstacles can be dealt with using arguments that, in principle, are similar to those presented already. The proof follows exactly the same path as that of Theorem 2, but takes into account that the set Q^{NT} may be non-empty. Thus, in addition to creating 2CNF formulas out of the sets of legal orderings from Q^T (see Table 1), we also do the same with the sets from Q^{NT} (using Table 2). As in the proof of Theorem 2, we show that the global formula has a satisfiable assignment if and only if a top-monotonic order exists.

Theorem 3. *Let A be a set of alternatives, N be a set of agents, and $\succcurlyeq$ be a preference profile over A. The problem of determining whether a top-monotonic order of $\succcurlyeq$ exists (and computing it) is polynomial-time solvable.*

Proof. We extend the proof of Theorem 2 by considering that Q^{NT} may be non-empty (if $Q^{NT} = \emptyset$ we can use Theorem 2 directly). This will be reflected in the number of additional combinations of alternatives that we have to consider. This will lead to an extended list of rules for the sets of legal orderings that, as we will show later on, can also be represented in the 2CNF form. This will allow us to use the same argument as in the proof of Theorem 2 to reduce the problem of determining a top-monotonic order to the SAT-2CNF problem.

Let us consider some element X from a non-empty set Q^{NT}. By the definition of Q^{NT}, X is a set of legal orderings for a pair of agents $i, j \in N$ and a triple of distinct alternatives x, y, ω, such that $x \in t_i(A)$, $y \in t_j(A)$, and $\omega \in A \setminus T$. There are exactly nine different possible combinations of preferences for the pair of agents i, j over the alternatives x, y, ω. We list all these combinations along with their corresponding sets of legal orderings in Table 2. Note that there are much fewer combinations than in Table 1 and this is because set Q^{NT} has a more strict definition than Q^T. Specifically, it is required that x is among

	Comb. of agents $i, j \in N$	Set of legal orderings	2CNF ordering formula
1	$x \succ_i \omega \succ_i y$ and $y \approx_j x \succ_j \omega$	$\{(\omega, x, y), (y, x, \omega), (x, \omega, y), (y, \omega, x)\}$	$(xy \vee y\omega) \wedge (\omega y \vee yx)$
2	$x \succ_i y \approx_i \omega$ and $y \succ_j x \succ_j \omega$	$\{(\omega, x, y), (y, \omega, x), (x, \omega, y), (y, x, \omega)\}$	$(xy \vee y\omega) \wedge (\omega y \vee yx)$
3	$x \succ_i y \succ_i \omega$ and $y \succ_j \omega \succ_j x$	$\{(y, \omega, x), (\omega, y, x), (x, \omega, y), (x, y, \omega)\}$	$(yx \vee x\omega) \wedge (\omega x \vee xy)$
4	$x \succ_i y \approx_i \omega$ and $y \succ_j \omega \succ_j x$	$\{(x, \omega, y), (y, \omega, x)\}$	$(xy \vee y\omega) \wedge (x\omega \vee yx) \wedge (\omega x \vee \omega y)$
5	$x \succ_i y \approx_i \omega$ and $y \approx_j x \succ_j \omega$	$\{(\omega, x, y), (y, x, \omega), (x, \omega, y), (y, \omega, x)\}$	$(xy \vee y\omega) \wedge (\omega y \vee yx)$
6	$x \succ_i y \approx_i \omega$ and $y \succ_j x \approx_j \omega$	$\{(y, \omega, x), (x, \omega, y)\}$	$(xy \vee y\omega) \wedge (x\omega \vee yx) \wedge (\omega x \vee \omega y)$
7	$x \succ_i y \succ_i \omega$ and $y \succ_j x \succ_j \omega$	all permutations of $\{x, y, \omega\}$	n/a
8	$x \succ_i y \succ_i \omega$ and $y \approx_j x \succ_j \omega$	all permutations of $\{x, y, \omega\}$	n/a
9	$x \approx_i y \succ_i \omega$ and $y \approx_j x \succ_j \omega$	all permutations of $\{x, y, \omega\}$	n/a

Table 2: All possible settings of preferences for pairs of agents $i, j \in N$ over the set of alternatives x, y, ω, where $x \in t_i(A)$, $y \in t_j(A)$ and $\omega \in A \setminus T$, with corresponding sets of legal orderings (third column). The last column shows 2CNF representations of each ordering formula (note that we write, e.g., yx instead of $\neg xy$). Rules 7–9 have the same interpretation as rules 19–21 in Table 1.

the most preferred alternatives for agent i, y is among the most preferred alternatives for agent j, and ω can never be the most preferred alternative.

Similarly to how we proceeded in the proof of Theorem 2, we can represent each of the sets of legal orderings from Table 2 as a 2CNF ordering formula. This time rules 7–9 do not introduce any constraints on the *top-monotonic* order.

We see now that we can make an instance I of a SAT-2CNF problem by taking the conjunction of the 2CNF formulas for the matching sets of legal orderings for all the elements from both Q^T and Q^{NT} by using the rules from Table 1 and 2 correspondingly, and by following the methodology from the proof of Theorem 2. We make similar claim as in that proof that a top-monotonic order exists for $\succcurlyeq$ if and only if I has a solution, and, if so, that the top-monotonic order is a linear extension of the order $>'$ induced by the assignment of the variables for the solution of I (in the same way as in the proof of Theorem 2). We can use almost all the same arguments as in the proof of Theorem 2 to prove that our reduction to SAT-2CNF is correct. The only exception regards showing that the transitivity property is fulfilled for the relation $>'$. We see that the transitivity property is fulfilled for all the triples $x, y, z \in T$ (by the argument from the proof of Theorem 2). We also see that it is fulfilled for all the triples x, y, ω, such that there are agents i and j such that $x \in t_i(A)$, $y \in t_j(A)$, $\omega \in A \setminus T$ and the preferences of these agents map to one of the rules 1–6 from Table 2 (the rules are "enforcing" transitivity, in the same way as in Theorem 2). The only situation that remains to be handled occurs if there exist $x', y' \in T$, $\omega' \in A \setminus T$, such that for each pair of agents $i, j \in N$, such that $x' \in t_i(A)$ and $y' \in t_j(A)$, the preferences of agents i and j over alternatives x', y', ω' map to the rules 7–9 from Table 2. In this case there is no 2CNF formula we can add to instance I that would "enforce" the transitivity between x', y' and ω', yet I may contain variables that correspond to the relations between each pair of these alternatives. We address this issue in the following lemma.

Lemma 4. *Let A be a set of alternatives, N be a set of agents, and $\succcurlyeq$ be a preference profile over A. Let T be a subset of A that contains exactly those alternatives that are top in A of some agent from N. If there exists a triple of alternatives x, y, ω, where $x, y \in T$*

and $\omega \in A \setminus T$, such that for every possible pair of candidates $i, j \in N$, with $x \in t_i(A)$ and $y \in t_j(A)$, the agents i and j's preferences over alternatives x, y, ω always match one of the rules 7–9 from Table 2, then SAT-2CNF instance I obtained for the set of agents N and the set of alternatives A will either not contain (some of) the variables that correspond to the relations between alternatives (x, y), (x, ω) and (y, ω) or each order $>$ that is obtained from each solution of I will fulfill transitivity property for the alternatives x, y, ω.

Proof. It is sufficient to show that if instance I is satisfiable and contains variables that correspond to at least one of (x, y), (x, ω) or (y, ω), then for all satisfying truth assignments for I the transitivity property is fulfilled for x, y and ω.[7] To the contrary, let us assume that instance I is satisfiable and for some solution of I the transitivity is not fulfilled for x, y, ω. For that to be true, I has to contain all three literals that map to (x, y), (x, ω) and (y, ω) as otherwise it would be impossible to encode the loss of transitivity. Without loss of generality, let us assume that the variables are xy, $x\omega$ and $y\omega$. Now, for the transitivity not to be satisfied they can have one of the two following assignments in the solution of I:

$$\begin{gathered} xy \leftarrow True, \quad x\omega \leftarrow False, \quad y\omega \leftarrow True, \\ \text{or} \\ xy \leftarrow False, \quad x\omega \leftarrow True, \quad y\omega \leftarrow False. \end{gathered} \tag{4}$$

We have assumed in the lemma statement that for all possible pairs of agents when considering triple $\{x, y, \omega\}$, we always get rules 7–9 from Table 2, which do not output any clauses. Yet, we need literals xy, $x\omega$ and $y\omega$ to be a part of the 2CNF formula that we built. Therefore there have to be additional alternatives in our election that, when considered jointly with x, y and ω, match rules that output these literals. So, let us assume that there is an alternative a such that when considering $\{a, x, \omega\}$, the triple matches one of the rules 1–6 from Table 2 and, therefore, generates clauses for our formula that include literal $x\omega$. Let us also assume that there exists an alternative b such that when considering triple $\{b, y, \omega\}$, we match rules that generate literal $y\omega$. It might be the case that $a = b$ but this is irrelevant for the rest of the proof, so we will not consider it as a separate case. We note that triples $\{a, x, \omega\}$ and $\{b, y, \omega\}$ can only match rules from Table 2 (because ω is never top in A), and hence both a and b must be in T. Finally, we also need literal xy to be generated. We will show later that for that to happen we do not need any additional alternatives in our election, and that either for $\{x, y, a\}$ or $\{x, y, b\}$ we will always get a rule that uses xy (note that a, b, x, and y are in T, so for the rules generated by aforementioned triples we use Table 1).

Based on the above discussion, we claim that if for a set of alternatives A that includes x, y, ω, a and b, set of agents N, and preference profile $\succcurlyeq$, there exists an instance I that includes literals xy, $x\omega$, and $y\omega$ and that can be satisfied with one of the assignments from Eq. 4, then we can find $N' \subseteq N$, $\|N'\| \leq 4$, such that instance I' computed for agents N', alternatives $A' = \{x, y, \omega, a, b\}$, and preference profile $\succcurlyeq$ is satisfiable, the set of literals in I' is a subset of literals from I, and the matching assignment for the solution of I is a solution for I'. In other words, we are going to show that if our lemma were false, then there would be an election with four agents and five candidates that would form a counterexample.

7. Note that the existence of x, y and ω in the set of alternatives alone does not imply the listed variables will be a part of I because we only add variables when needed by at least one of the rules we generate

Let us first consider a pair of agents i, j from N that, when considered, would output a rule that includes literal $x\omega$ (if there are many such pairs, we pick one arbitrarily). It means that both agents i and j have at least one of x and a as top in A, and that ω is not top in A of neither i nor j (in fact, it is not top of any agent from A). Formally, we have:

$$t_i(A) \cap \{x, a\} \neq \emptyset, \qquad t_j(A) \cap \{x, a\} \neq \emptyset, \qquad \text{and} \qquad \omega \notin t_i(A) \cup t_j(A).$$

We note that the same holds true for A'. This follows from the fact that if x were top in A of some agent, it would also be top in all subsets of A that include x. Also ω cannot be top in A' of neither i nor j. This is true because if x is top in A of i, then since ω is not top in A of i, agent i has to prefer x over ω, and therefore ω is not top in A' for i too. Otherwise, if x is not top in A of i, then a is for sure and the same reasoning can be used to show that ω is not top in A' of i. Finally, we can show the same for j, that is, that ω is not top in A' of j.

We select the second pair k, l of agents from N that are responsible for generating literal $y\omega$ in I in a similar way. We now see that the rules generated for agents $N' = \{i, j, k, l\}$ and alternatives $A' = \{x, y, \omega, a, b\}$ are the same under both I' and I. This is true because the preferences of the agents in N' with respect to the alternatives from A' remain unchanged in I'. Therefore, given our assumption that there is a solution of I where transitivity is not fulfilled for x, y, ω we see there exists a matching solution for I' with the same characteristic.

As a result of the above, we can see that assuming I exists we can find an instance I' that only has five alternatives and no more than four agents for which a satisfying truth assignment breaks transitivity. If we could show that no such instance I' exists, then we would prove our lemma by showing our assumption on the existence of I is not true.

So far we have not been able to find an easy-to-follow proof for showing that a counterexample for our lemma cannot be found in an election with up to four agents and five candidates. But since the number of possible problem instances is bounded by an acceptably small constant, we were able to perform a computer-assisted proof by exhaustive search where we tried all possible instances I'. The computer program we used works as follows:

- ▷ For a set of alternatives $A = \{x, y, \omega, a, b\}$, each possible set of agents N of size no bigger than four, and each possible preference profile $\succcurlyeq$ such that ω is not top on A for any agent from N and the profile leads to the literals $x\omega$ and $y\omega$ being generated, do:
  - ▷ Verify that literal xy is generated for $\succcurlyeq$.[8]
  - ▷ If $\succcurlyeq$ is *top-monotonic* (we use Definition 3 and test every possible order of alternatives) then check if instance I can be satisfied for one of the initial assignments of literals from Eq. 4; if it can, then we have found a profile that fails our criteria.

Below we describe the methodology that our program follows for scanning through all the possible combinations in the steps described above. Along the way we also provide an upper bound on the number of cases the program needs to consider.

For the initial step, it suffices to consider orderings that do not satisfy transitivity for x, y and ω, that is, orderings which for pairs (x, y), (x, ω) and (y, ω) match one of the options

8. For every profile where literals $x\omega$ and $y\omega$ were generated, literal xy was generated as well.

from Eq. (4). This brings the number of orderings that we need to consider from $2^{10} = 1024$ (for each of the 10 pairs of alternatives we need to choose in which of two ways it is ordered) down to $2 \cdot 2^7 = 256$ (we have two versions for arranging the three crucial pairs and we need to consider all possible arrangements for the remaining 7 pairs).

To get a sense for how many possible preference profiles we need to consider, let us start with an upper bound on the number of possible votes over five alternatives that we need to consider. For that we take any possible arrangement of five alternatives (in one of $5!$ ways) and in four places in between agents we place $\succ$ or $\approx$ symbol in 2^4 ways. This yields $5!\, 2^4 = 1920$ possible combinations. This number is pretty large, given that we need to consider four agents. This would lead to $1920^4 \approx 10^{13}$ possible profiles.

To reduce this number, we take advantage of the fact that we do not want to consider votes that place ω among top choices. Furthermore, we note that we only want to consider profiles in which no rules are being generated for triple (x, y, ω), so for each pair of agents we always expect that triple to fall under rules 7–9 from Table 2. For that to happen, if x or y is among top choices in a vote we also require that $x \succ \omega$ and $y \succ \omega$. This follows from the fact that a unique feature of rules 7–9 is that for both agents considered for generating the rule we always have $x \succ \omega$ and $y \succ \omega$.

To illustrate what votes are worth considering for our cases, let us look at the following examples. Vote $x \approx b \succ y \approx a \succ \omega$ is fine, as even though x is top of that agent, we also have $y \succ \omega$. Vote $a \succ \omega \approx b \succ x \succ y$ is also fine because neither x nor y is top of that agent. On the other hand, vote $y \approx a \succ x \approx b \approx \omega$ is not fine, because y is top while $x \approx \omega$.

To count the number of possible votes given all the constraints described above, we list four possible shapes of a valid vote:

Shape A: $\underline{x\quad} \succ \underline{y\quad} \succ \underline{\omega\quad}$ – each group (marked with an underline) represents a part of a vote where remaining alternatives can be added. Alternatives within a group can be reordered and separated with either $\approx$ or $\succ$. The only exception is the first group, where we can only use $\approx$ and hence the order of alternatives does not matter.

Shape B: $\underline{y\quad} \succ \underline{x\quad} \succ \underline{\omega\quad}$ – a symmetric case to the previous one, where x and y are swapped.

Shape C: $\underline{x \approx y\quad} \succ \underline{\omega\quad}$ – similarly here groups can be extended by adding remaining alternatives (a and b); in the first group only equality ($\approx$) can be used between the alternatives.

Shape D: $\underline{\quad} \succ \underline{x\ ?\ y\ ?\ \omega\quad}$ – in this case we require that the first group contains at least one alternative. We use ? in the second group to indicate the alternatives there can be reordered and either $\succ$ or $\approx$ can be used in place of ?.

We want to count the number of possible votes for each of the cases defined above. But first let us consider some simplified cases that will help with further calculations. If we have a group that contains two alternatives (say a and b), and in the group we can use either $\approx$ or $\succ$ to separate the alternatives, then there are three possible ways to order the alternatives in such a group, $a \succ b$, $b \succ a$ and $a \approx b$. Now let us consider a group of three alternatives. In such a case, there are 6 ways in which only $\succ$ is used, 6 ways with one $\succ$ and one $\approx$ (we have 3 unique ways when $\approx$ is between the first pair and 3 ways when it is between the second pair), and a single way with two $\approx$ symbols. In total, a group of three alternatives yields 13 unique ordering combinations. In a similar way, we can calculate the result for a group of four alternatives, which has 85 unique orderings.

Using the above we can count the number of possible orderings for each shape of a valid vote presented above:

Shape A: We first consider that we choose two different groups to add a and b. This can be done in 6 different ways. In that case, alternatives in these two groups (of two elements each) can be ordered in 3 unique ways (see above). This gives $6 \cdot 3 \cdot 3 = 54$ combinations in total. Another option is to select one group in 3 possible ways and add both a and b to it. Groups will have three alternatives so that gives $3 \cdot 13 = 39$ possible orderings. We sum up both cases to get $54 + 39 = 93$ possible combinations.

Shape B: This case is symmetric to Case 1 and similarly yields 93 possible combinations.

Shape C: In this case we can put both a and b in the first group in only one way (we need to use equality). If we decided to place both a and b in the second group we can do that in 13 ways (a group of three alternatives). Finally, if we put one alternative in the first and one in the second group, we get $2 \cdot 3 = 6$ ways. This sums up to $1 + 13 + 6 = 20$ possible votes.

Shape D: Similarly, for the last shape we consider placing both a and b in the first group. Since there are three alternatives in the second group, we can have 13 ordering combinations in that case. As we always need to place one alternative in the first group the only other way to create a valid ordering is by placing one alternative in the first group and the other one in the second. This can be done in 2 ways and the group will contain four alternatives so it can be ordered in 85 ways. In total we get $13 + 2 \cdot 85 = 183$ possible votes.

Now, let us get an upper bound for the number of possible profiles of four agents we want to consider. Given the above, we have $93 + 93 + 20 + 183 = 389$ different votes that we need to consider. Among these, we only have $93+20 = 113$ possible votes that rank x on top. We want to enforce that x and y are both top on A for some agents from N, so we can pick first agent's vote from this selection of 113 votes, and then the remaining ones from the full list of 389 possibilities. Note that, while following this methodology, we still may get combinations in which y is not on top for any of the agents. If we eliminated this possibility too, the upper bound would go down even further, but it would complicate the way of generating profiles. Instead, we filter out profiles that do not match our requirements. With that taken into account, we have $113 \cdot 389^3 < 7 \cdot 10^9$ possible profiles we should consider, which puts it in a reasonable range for a middle-end desktop hardware to process within a matter of seconds. As mentioned earlier, the profiles we can generate this way may not meet all the constraints required by the algorithm. Also, the generated profile may not lead to all the required literals being generated. We can therefore rule out even more profiles at the early step and avoid further calculations.

Our program has verified all possible combinations of instances with five alternatives and four or fewer agents and have not encountered an instance that would satisfy the algorithm. $\square$

By Lemma 4, we see that the case in which the transitivity property cannot be fulfilled even though a solution for I exists is not possible. We therefore see that the transitivity property is satisfied for $>'$. Similarly to the proof of Theorem 2, we state now that I has a solution if and only if $\succcurlyeq$ is top-monotonic, and if the solution exists, then a linear extension of the order $>'$ induced by the solution is a top-monotonic order of $\succcurlyeq$. We also see that I can be computed and solved in polynomial-time with respect to the numbers of agents and alternatives. The only difference comparing to the proof of Theorem 2 is that in the rules lookup phase we also consider some triples made of candidates that are not top in A for any agent. The upper bound for the number of literals and variables in I are the same, $3 \cdot N^2 \cdot \|A\|^3$ and $\|A^3\|$ correspondingly, as each combination of agents and alternatives will match at most one rule (either from Table 1 or from Table 2). □

Below we summarize the main steps of our algorithm and establish $O(m^4n^2)$ as a simple upper bound on its running time. We note that the algorithm could be implemented more effectively using various tricks and more effective data structures—what we describe here is the most basic implementation. The algorithm consists of the following five steps:

1. We first compute the set T. This can be done in time $O(nm)$ as we simply need to iterate over all the preference orders and mark the alternatives that are ranked on top; there are n agents and each of them ranks m candidates.

2. We elliminate indistinguishable alternatives: For each pair of alternatives we check if there is some agent that strictly prefers one alternative from the pair over the other. If such an agent does not exist, then we remove one of these alternatives from the preference profile. A naive implementation of this step requires time $O(m^3n)$; for each pair of alternatives we look at the whole profile of size $O(nm)$.

3. We generate our SAT-2CNF formula. This is the most computationally intensive part of the algorithm and it requires time $O(m^4n^2)$. We consider all triples of candidates (such that either all three candidates belong to T or two of them belong to T and one does not) and all pairs of agents; there are $O(m^3n^2)$ combinations to consider. We extract the preferences regarding each combination of alternatives and agents in time $O(m)$ and we form respective clauses by looking up our tables (each combination leads to at most a few added clauses).

4. We solve the formed SAT-2CNF formula using a standard linear-time algorithm. Since the formula has at most $O(m^3n^2)$ clauses, solving it takes at most $O(m^3n^2)$ time.

5. In the final step we prepare the top-monotonic order based on the partial order implied by the truth assignment for the SAT-2CNF formula. This can be done in time $O(m^2)$ using the standard topological sorting algorithm. In this step we also reinsert the indistinguishable alternatives that were removed in the second step (we place them right next to the candidates that they are indistinguishable from).

As our algorithm is relatively slow, it is natural to ask if there is a faster one, or if, e.g., there is a natural formulation of the problem as an integer linear program, which can be solved more efficiently in practice.

3.4 SAT-2CNF Algorithm for Recognizing Single-Peaked Profiles

The definition of top-monotonicity is not very intuitive and, by necessity, the algorithm presented in the preceding sections is somewhat involved. One exercise that may help with understanding our core ideas is to adapt our methodology to a more intuitive domain. Below we use our approach to derive an algorithm for recognizing single-peaked profiles (analogous technique, focused on ordering voters, also works for recognizing single-crossing profiles). While more efficient algorithms for solving this problem exist (Bartholdi & Trick, 1986; Escoffier et al., 2008), our point is to illustrate the approach.

As per Definition 1, a preference profile $\succ$ is single-peaked if there exists a linear order $>$ over the set of alternatives such that for each three alternatives x, y, and z it holds that:

$$(x > y > z) \vee (z > y > x) \implies \forall_{i \in N}(x \succ_i y \implies y \succ_i z).$$

We follow the notation from this definition. For each triple of alternatives $S = \{x, y, z\}$ we define the set L_S of legal orderings to be the set of ordered sequences $\sigma = (\sigma_1, \sigma_2, \sigma_3)$, such that:

1. $\{\sigma_1, \sigma_2, \sigma_3\} = S$, and
2. for each $i \in N$ we have $\sigma_1 \succ_i \sigma_2 \implies \sigma_2 \succ_i \sigma_3$.

We see that the set of legal orderings for a given triple tells us what are the possible ways in which the alternatives from the triple can be ordered on the societal axis, assuming it exists, in order to fulfill the requirements of single-peakedness. Therefore, if for some triple of alternatives S the set of legal orderings L_S is empty, we know that the profile is not single-peaked as that would mean there is no way these alternatives can be placed on the axis without violating the definition's requirements. From now on we assume that for every triple S, L_S is non-empty.

We note that if for some set $S = \{\sigma_1, \sigma_2, \sigma_3\}$ a triple $(\sigma_1, \sigma_2, \sigma_3)$ belongs to L_S, then triple $(\sigma_3, \sigma_2, \sigma_1)$ also belongs to L_S. This follows because the condition for including triple $(\sigma_1, \sigma_2, \sigma_3)$ simply says that if we restrict the preference orders of all the agents to alternatives $\sigma_1, \sigma_2, \sigma_3$, then σ_2 is never ranked last (so, in terms of Fishburn, 1997, we have a *never last* restriction). The same condition is satisfied by triple $(\sigma_3, \sigma_2, \sigma_1)$.

Let us consider a set $S = \{x, y, z\}$ of three alternatives and agent $i \in N$. There are six possible permutations on how alternatives from S can be ordered according to preference $\succ_i$ of agent i. Let us first assume that $x \succ_i y \succ_i z$. In such a case, we see that neither (y, z, x) nor (x, z, y) is a part of L_S. This follows from the fact that the least favorable alternative cannot be placed in between the more favorable ones on the societal axis, so such a setup clearly violates the definition of single-peakedness. Thus, regardless of how alternatives from S are ordered according to $\succ_i$, we can always find at least one pair of sequences of alternatives from S that are not included in L_S. Therefore what we are left with are either two or four possible sequences in L_S for each given S (we already assumed that L_S is never empty and we argued that if a sequence belongs to L_S then so does its reverse).

Below we consider two possible sizes of the set L_S for a selected triple $S = \{x, y, z\}$:

1. If L_S contains exactly two elements, then—up to renaming the alternatives—it is of the form $L_S = \{(x, y, z), (z, y, x)\}$.

2. If L_S contains four elements, then—up to renaming the alternatives—it is of the form $L_S = \{(x, y, z),\ (z, y, x),\ (y, x, z),\ (z, x, y)\}$. Indeed, for each sequence in L_S its reverse must be included as well, and one can verify that—up to renaming of the candidates—this is the only possible form of L_S.

Following the methodology from the algorithm for recognizing top-monotonic profiles, we now want to express the sets of legal orderings in the form of 2CNF formulas. As in the algorithm for the case of top-monotonicity, the literals in the formula will correspond to the ordered pairs of alternatives. For example, if we take alternatives x and y, then we create two complementary literals xy and yx, where literal xy is a negation of the literal yx (we choose arbitrarily which one is represented as a variable and which one is this variable's negation). For the first variant of the set L_S (with two elements) we get a 2CNF formula $(xy \vee zx) \wedge (xz \vee zy) \wedge (yz \vee yx)$, while for the second variant (with four elements) we get $(xz \vee zy) \wedge (yz \vee zx)$. We can use Table 1 to lookup rules based on the shape of the set of legal orderings. The listed formulas are respectively from the rule no. 12 and rule no. 1.

To summarize the process, for each triple of alternatives $S = \{a, b, c\}$ we compute the set of legal orderings L_S that will have one of the following shapes:

1. If L_S has two elements, then we can find a mapping between S and the set $\{x, y, z\}$, such that when we map elements from L_S we will get $\{(x, y, z), (z, y, x)\}$. In this case we generate a 2CNF formula following rule no. 12 from Table 1, $(xy \vee zx) \wedge (xz \vee zy) \wedge (yz \vee yx)$, by applying an inverse mapping function to the alternatives from S.

2. If L_S has four elements, then we can find a mapping between S and the set $\{x, y, z\}$, such that after mapping all the elements from L_S we get a set $\{(x, y, z), (z, y, x), (y, x, z), (z, x, y)\}$. We generate a 2CNF formula based on rule no. 1 from Table 1, $(xz \vee zy) \wedge (yz \vee zx)$, and translate it back into the domain of alternatives from S using inverse mapping.

Finally, we form the global ordering formula by taking a conjunction of all the generated 2CNF formulas. We claim that if the global ordering formula is satisfiable, then the profile is single-peaked, and otherwise it is not.

To prove the above statement, we first assume that the global ordering formula is satisfiable. In this case, for each pair of alternatives x, y, we define a relation $>$ such that $x > y$ if literal xy evaluates to $True$ in the global ordering formula solution, and $y > x$ if literal xy evaluates to $False$. We see that $>$ is a strict order over the set of alternatives because:

(a) It is defined for all possible distinct pairs of alternatives.

(b) It is irreflexive as we do not define the relation between two identical alternatives.

(c) It is asymmetric, as otherwise for some two candidates x and y we would have literal xy which would evaluate to both $True$ and $False$.

(d) It is transitive, because transitivity is enforced by the order of triples of alternatives S in the set L_S.

We note that $>$ (assuming it exists) is a single-peaked axis for the preference profile. We see that for each set of three alternatives $S = \{x, y, z\}$, if $x > y > z$ or $z > y > x$

then both (x, y, z) and (z, y, x) are included in L_S. This follows from the fact that the rule generated based on the set L_S enforces the relation such that x, y, and z are always placed in one of the orders from L_S. This in turn means that for each agent $i \in N$ we have $x \succ_i y \implies y \succ_i z$, as we require this for both (x, y, z) and (z, y, x) to be included in L_S.

To show the right to left part, it is sufficient to note that if single-peaked order exists then the global ordering formula is satisfiable. We assume $>$ exists and is the single-peaked order of our profile. We now take the global ordering formula and we assign $True$ to each literal xy such that x and y are two distinct alternatives such that $x > y$. To the contrary, let us assume that the above assignment of literals does not satisfy the global ordering formula. It means that there is a clause in the formula that has two literals both evaluating to $False$. We know that every clause in the formula corresponds to a set of legal orderings for some triple of alternatives. We let $S = \{x, y, z\}$ be a triple of alternatives and L_S be a set of legal orderings the failing clause corresponds to. As L_S contains all orderings of alternatives x,y and z that can legally appear on the societal axis, we see that the fact that the clause is not satisfied means that $>$ does not order x, y and z in one of these possible ways. This contradicts the fact that $>$ is a single-peaked axis and therefore shows the right-to-left part of our statement.

Example 5. *Consider candidate set $\{a, b, c, d\}$ and profile $\succ$ of three preference orders:*

$$b \succ_1 c \succ_1 d \succ_1 a, \qquad c \succ_2 b \succ_2 d \succ_2 a, \qquad a \succ_3 b \succ_3 c \succ_3 d.$$

To run our algorithm we start by defining the sets of legal orderings for each set of three alternatives:

1. *For the set $S_1 = \{a, b, c\}$, we have $L_{S_1} = \{(a, b, c), (c, b, a)\}$. Clearly, (b, a, c) and its reverse cannot be a part of L_{S_1} because $b \succ_2 a$ holds while $a \not\succ_2 c$ which makes it not fulfill the definition as we would expect that $b \succ_2 a \implies a \succ_2 c$ is true. Similarly, sequence (a, c, b) and its reverse are not included as we have that $a \succ_3 c$ but also $c \not\succ_3 b$.*
2. *For the set $S_2 = \{a, b, d\}$, we have $L_{S_2} = \{(a, b, d), (d, b, a)\}$.*
3. *For the set $S_3 = \{a, c, d\}$, we have $L_{S_3} = \{(a, c, d), (d, c, a)\}$.*
4. *Finally, for the set $S_4 = \{b, c, d\}$, we have $L_{S_4} = \{(b, c, d), (d, c, b), (c, b, d), (d, b, c)\}$. Here we only eliminated sequence (c, d, b) and its reverse, which do not satisfy the single-peakedness condition for any $i \in \{1, 2, 3\}$.*

As we see, there are no empty sets of legal orderings and, so, we can move on to the next step.

For each of S_i, $1 \leq i \leq 4$, we output the 2CNF formula that corresponds to L_{S_i} and we take the conjunction of these formulas as the global ordering formula. We present this formula below (the following lines represent the clauses generated from L_{S_1}, L_{S_2}, L_{S_3}, and L_{S_4}, respectively):

$$\begin{aligned} &(ab \lor ca) \land (ac \lor cb) \land (bc \lor ba) \\ \land &(ab \lor da) \land (ad \lor db) \land (bd \lor ba) \end{aligned}$$

$$\wedge(ac \vee da) \wedge (ad \vee dc) \wedge (cd \vee ca)$$
$$\wedge(cd \vee db) \wedge (bd \vee dc).$$

For each pair of alternatives x and y from $\{a, b, c, d\}$ we choose one of xy and yx to be a variable and the other to be its negation, to obtain the following global ordering formula:

$$\begin{aligned}
&(\underline{ab} \vee \neg ac) \wedge (\underline{ac} \vee \neg bc) \wedge (\underline{bc} \vee \neg ab) \\
\wedge&(\underline{ab} \vee \neg ad) \wedge (\underline{ad} \vee \neg bd) \wedge (\underline{bd} \vee \neg ab) \\
\wedge&(\underline{ac} \vee \neg ad) \wedge (\underline{ad} \vee \neg cd) \wedge (\underline{cd} \vee \neg ac) \\
&\wedge(\underline{cd} \vee \neg bd) \wedge (\underline{bd} \vee \neg cd).
\end{aligned}$$

We seek a satisfying truth assignment for this formula. This can be done using one of the polynomial-time solvers for SAT-2CNF problem, but in our case it suffices to notice that every clause contains a non-negated variable and, so, we simply set all the variables to be true. In the formula above we underlined the literals that are set to be true.

The fact that the global ordering formula has a solution indicates that the profile is single-peaked. From the solution of the formula we also get single-peaked order $R_> = \{(a, b), (a, c), (a, d), (b, c), (b, d), (c, d)\}$ that yields societal axis $a > b > c > d$.

4. Conclusion

We have given the first polynomial-time algorithm for recognizing if a profile of (possibly weak) preference orders is top-monotonic. Top-monotonic preferences are in principle very attractive. For example, they subsume single-peaked and single-crossing ones, while ensuring that a (weak) Condorcet winner always exists. However, they are not easy to work with.

Our proof relies on a novel way of modeling restricted domain problems by reducing them to boolean satisfiability problems. This methodology makes the top-monotonic definition easier to reason about and far more approachable to consider with respect to computational social choice problems. We also show that our approach is more general and can be used, e.g., for the single-peaked domain (and possibly other restricted domains too).

We therefore hope that our work will enable further researchers to show positive algorithmic consequences of the top-monotonicity assumption. For example, it is natural to ask if the Chamberlin–Courant rule is polynomial-time solvable under top-monotonic preferences; it is under single-peaked (Betzler et al., 2013) and single-crossing ones (Skowron et al., 2015). It is also interesting to compare the notion of top-monotonicity to that of value-restricted profiles (Sen, 1966). We also hope that the method presented in this paper can be successfully applied to other problems from the restricted domains area.

Acknowledgments

An early version of this paper was presented at IJCAI-2017. The current version includes all the proofs and an additional section illustrating how our techniques can be used in the case of single-peaked preferences. Piotr Faliszewski was supported by AGH University grant 11.11.230.337 (statutory research). We are very grateful to the IJCAI-17 and JAIR reviewers for helpful comments.

References

Arrow, K. (1951). *Social Choice and Individual Values*. John Wiley and Sons. Revised editon, 1963.

Aziz, H. (2014). Testing top monotonicity. Tech. rep. arXiv:1403.7625v5 [cs.GT], arXiv.org.

Barberà, S., & Moreno, B. (2011). Top monotonicity: A common root for single peakedness, single crossing and the median voter result. *Games and Economic Behavior*, *73*(2), 345–359.

Bartholdi, III, J., & Trick, M. (1986). Stable matching with preferences derived from a psychological model. *Operations Research Letters*, *5*(4), 165–169.

Betzler, N., Slinko, A., & Uhlmann, J. (2013). On the computation of fully proportional representation. *Journal of Artificial Intelligence Research*, *47*(1), 475–519.

Black, D. (1958). *The Theory of Committees and Elections*. Cambridge University Press.

Booth, K., & Lueker, G. (1976). Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *Journal of Computer and System Sciences*, *13*(3), 335–379.

Brandt, F., Brill, M., Hemaspaandra, E., & Hemaspaandra, L. (2015). Bypassing combinatorial protections: Polynomial-time algorithms for single-peaked electorates. *Journal of Artificial Intelligence Research*, *53*, 439–496.

Bredereck, R., Chen, J., & Woeginger, G. (2013). A characterization of the single-crossing domain. *Social Choice and Welfare*, *41*(4), 989–998.

Clearwater, A., Puppe, C., & Slinko, A. (2015). Generalizing the single-crossing property on lines and trees to intermediate preferences on median graphs. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pp. 32–38.

Conitzer, V. (2009). Eliciting single-peaked preferences using comparison queries. *Journal of Artificial Intelligence Research*, *35*, 161–191.

Cornaz, D., Galand, L., & Spanjaard, O. (2012). Bounded single-peaked width and proportional representation. In *Proceedings of the 20th European Conference on Artificial Intelligence*, pp. 270–275.

Cornaz, D., Galand, L., & Spanjaard, O. (2013). Kemeny elections with bounded single-peaked or single-crossing width. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pp. 76–82.

Elkind, E., Faliszewski, P., Lackner, M., & Obraztsova, S. (2015). The complexity of recognizing incomplete single-crossing preferences. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pp. 865–871.

Elkind, E., Faliszewski, P., & Slinko, A. (2012). Clone structures in voters' preferences. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pp. 496–513.

Elkind, E., & Lackner, M. (2015). Structure in dichotomous preferences. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pp. 2019–2025.

Escoffier, B., Lang, J., & Öztürk, M. (2008). Single-peaked consistency and its complexity. In *Proceedings of the 18th European Conference on Artificial Intelligence*, pp. 366–370.

Faliszewski, P., Hemaspaandra, E., & Hemaspaandra, L. (2014). The complexity of manipulative attacks in nearly single-peaked electorates. *Artificial Intelligence*, *207*, 69–99.

Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L., & Rothe, J. (2011). The shield that never was: Societies with single-peaked preferences are more open to manipulation and control. *Information and Computation*, *209*(2), 89–107.

Fishburn, P. (1997). Acyclic sets of linear orders. *Social Choice and Welfare*, *14*(1), 113–124.

Gaertner, W. (2001). *Domain Conditions in Social Choice Theory*. Cambridge University Press.

Gibbard, A. (1973). Manipulation of voting schemes. *Econometrica*, *41*(4), 587–601.

Guttmann, W., & Maucher, M. (2006). Variations on an ordering theme with constraints. In *4th IFIP International Conference on Theoretical Computer Science–TCS 2006*, pp. 77–90. Springer.

Krom, M. (1967). The decision problem for a class of first-order formulas in which all disjunctions are binary. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, *13*, 15–20.

Lackner, M. (2014). Incomplete preferences in single-peaked electorates. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pp. 742–748.

Magiera, K., & Faliszewski, P. (2017). How hard is control in single-crossing elections?. *Autonomous Agents and Multiagent Systems*, *31*(3), 606–627.

Mirrlees, J. (1971). An exploration in the theory of optimal income taxation. *Review of Economic Studies*, *38*, 175–208.

Monjardet, B. (2009). Acyclic domains of linear orders: A survey. In Brams, S., Gehrlein, W., & Roberts, F. (Eds.), *The Mathematics of Preference, Choice and Order*, Studies in Choice and Welfare, pp. 139–160. Springer Berlin Heidelberg.

Peters, D. (2017). Recognising multidimensional Euclidean preferences. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pp. 642–648.

Peters, D. (2018). Single-peakedness and total unimodularity: New polynomial-time algorithms for multi-winner elections. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pp. 1169–1176.

Peters, D., & Lackner, M. (2017). Preferences single-peaked on a circle. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pp. 649–655.

Puppe, C., & Slinko, A. (2017). Condorcet domains, median graphs and the single-crossing property. *Economic Theory*, 1–34.

Roberts, K. (1977). Voting over income tax schedules. *Journal of Public Economics*, *8*(3), 329–340.

Saporiti, A., & Tohmé, F. (2006). Single-crossing, strategic voting and the median choice rule. *Social Choice and Welfare*, *26*(2), 363–383.

Satterthwaite, M. (1975). Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, *10*(2), 187–217.

Sen, A. (1966). A possibility theorem on majority decisions. *Econometrica*, *34*(2), 491–499.

Skowron, P., Yu, L., Faliszewski, P., & Elkind, E. (2015). The complexity of fully proportional representation for single-crossing electorates. *Theoretical Computer Science*, *569*, 43–57.