

# Recognizing Single-Peaked Preferences on an Arbitrary Graph: Complexity and Algorithms

Bruno Escoffier and Olivier Spanjaard and Magdaléna Tydrichová

## Abstract

This paper is devoted to a study of single-peakedness on arbitrary graphs. Given a collection of preferences (rankings of a set of alternatives), we aim at determining a connected graph  $G$  on which the preferences are single-peaked, in the sense that all the preferences are traversals of  $G$ . Note that a collection of preferences is always single-peaked on the complete graph. We propose an Integer Linear Programming formulation (ILP) of the problem of minimizing the number of edges in  $G$  or the maximum degree of a vertex in  $G$ . We prove that both problems are NP-hard in the general case. However, we show that if the optimal number of edges is  $m-1$  (where  $m$  is the number of candidates) then any optimal solution of the ILP is integer and thus the integrality constraints can be relaxed. This provides an alternative proof of the polynomial time complexity of recognizing single-peaked preferences on a tree. We prove the same result for the case of a path (an axis), providing here also an alternative proof of polynomiality of the recognition problem. Furthermore, we provide a polynomial time procedure to recognize single-peaked preferences on a pseudotree (a connected graph that contains at most one cycle). We also give some experimental results, both on real and synthetic datasets.

## 1 INTRODUCTION

Aggregating the preferences of multiple agents is a primary task in many applications of artificial intelligence, e.g., in preference learning [8, 9] or in recommender systems [2, 20]. The individual preferences of the agents are often represented as rankings on a set of alternatives, where the alternatives may be cultural products (books, songs, movies...), technological products, candidates for an election, etc. The aim of preference aggregation is then to produce an aggregate ranking from a collection of rankings (called preference profile).

The preferences are said to be *structured* if they share some common structure [12]. For example, in a political context, it is conventional to assume that each individual preference is decreasing as one moves away from the preferred candidate along a left-right axis on the candidates, axis on which individuals all agree. Such preferences are called *single-peaked* [5]. They have been the subject of much work in social choice theory. The most well-known result states that if preferences are single-peaked, then one escapes from Arrow’s impossibility theorem. We recall that Arrow’s theorem states that any unanimous aggregation function for which the pairwise comparison between two alternatives is independent of irrelevant alternatives is dictatorial. Furthermore, from the computational viewpoint, many NP-hard social choice problems (e.g., Kemeny rule and Young rule for rank aggregation [6], Chamberlin-Courant rule for proportional representation [4]) become polynomially solvable if the preferences are single-peaked.

Given the axiomatic and algorithmic consequences, the question of the computational complexity of recognizing single-peaked preferences is thus natural. Bartholdi and Trick [3] have proposed an  $O(nm^2)$  algorithm to compute a compact representation of *all* axes on which a collection of  $n$  preferences on  $m$  candidates are single-peaked, or state that none exists. This complexity can be decreased to  $O(nm)$  if one looks for only *one* possible axis [14].

Several classes of structured preferences have been proposed in the literature in order to generalize the single-peaked domain with respect to an axis, i.e., a path, to more general graphs. Given a set  $\mathcal{C} = \{1, \dots, m\}$  of candidates, a preference order  $\succ$  over  $\mathcal{C}$  is single-peaked on an undirected graph  $\mathcal{G} = (\mathcal{C}, \mathcal{E})$  if it is a *traversal* of  $\mathcal{G}$ , i.e., for each  $j \in \mathcal{C}$  the upper-contour set  $\{i \in \mathcal{C} : i \succ j\}$  is connected. A preference profile is then single-peaked on  $\mathcal{G}$  if every preference is single-peaked on

$\mathcal{G}$ . Demange studied single-peakedness on a tree [11]; Peters and Lackner studied single-peakedness on a circle [23].

Some good axiomatic properties remain valid when preferences are single-peaked on a tree: if the number of voters is odd, such profiles still admit a Condorcet winner (a candidate who is preferred over each other candidate by a majority of voters) [11], and returning this Condorcet winner is a strategyproof voting rule. On the contrary, every majority relation can be realized by a collection of preferences that are single-peaked on a circle [23], and consequently single-peaked preferences on a circle do not inherit the good axiomatic properties of single-peakedness on an axis regarding voting rules that are based on the majority relation.

The goal of this paper is to study the recognition problem for single-peaked preferences on arbitrary connected graphs. Although one cannot expect social choice theoretic guarantees from single-peakedness on an arbitrary graph (as it does not result in a domain restriction, in the sense that any preference profile is single-peaked on an arbitrary graph), the knowledge of such graphs indeed provides information based on preferences that gives some insights on the similarity of candidates/items, that could be used for instance in recommendation systems. For instance, assume that one discovers that the preferences over movies  $\{1, 2, 3, 4, 5\}$  are single-peaked w.r.t. an axis  $(1, 2, 3, 4, 5)$  over the movies. If one knows that an agent likes movies 3 and 5, then it is natural to recommend to watch movie 4. More generally, one can take advantage of single-peakedness on a sparse graph in order to make recommendations in the neighbourhood of liked items. Thereby, we focus here on determining a graph that minimizes (1) the number of edges of the graph or (2) the maximum degree of a vertex. This choice is motivated by the fact that these criteria are measures of sparsity of a graph (the sparsest the graph is, the more informative), but also because they generalize known cases such as paths, cycles and trees. Let us indeed emphasize that the mathematical programming approach we propose to identify a graph generalizes the best known instances of the single-peaked recognition problem and provides a uniform treatment of them, leading to simple polynomial time algorithms.

**Our contribution.** We propose here an Integer Linear Programming formulation (ILP) of problems (1) and (2), and we show that both of them are NP-hard. Nevertheless, if the optimal value for problem (1) is  $m - 1$  (where  $m$  is the number of candidates), we prove the integrality of the optimal basis solution of the linear program obtained by relaxing the integrality constraint in the ILP. This provides an alternative polynomial time method, based on a simple linear programming solver, to recognize single-peakedness on a tree, as a connected graph with  $m$  vertices and  $m - 1$  edges is a tree. By adding some constraints on the max degree of a vertex, we obtain the same result for the case of paths. As a last theoretical result, we prove that preferences single-peaked on a *pseudotree* (a connected graph that contains at most one cycle) can be recognized in polynomial time. We also provide some experimental results, both on real-world and synthetic datasets, where we measure the density of the graphs depending on the diversity of preferences of voters.

**Related work.** We briefly describe here some previous contributions that have addressed the concept of single-peakedness on arbitrary graphs, the optimization view of the recognition problem and the use of integer linear programming formulations for computational social choice problems related to structured preferences:

- Nehring and Puppe defined a general notion of single-peaked preferences based on abstract betweenness relations between candidates [19]. In their setting, it is possible to define single-peaked preferences on a graph  $G$  by considering the *graphic betweenness relation*: candidate  $j$  is between candidates  $i$  and  $k$  if and only if  $j$  lies on a shortest path between  $i$  and  $k$  in  $G$ . A preference profile is then single-peaked on  $G$  if for every preference  $\succ$ , if  $i^*$  is the most preferred candidate w.r.t.  $\succ$  and  $j$  is on a shortest path between  $i^*$  and  $k$  then  $j \succ k$ . This definition enables them to state general results regarding strategyproofness on restricted domains of preferences. Note that this definition of single-peakedness on a graph does not coincide with the one we use.
- Peters and Elkind showed how to compute in polynomial time a compact representation of *all*

trees with respect to which a given profile is single-peaked [22]. This structure allows them to find in polynomial time trees that have, e.g., the minimum degree, diameter, or number of internal nodes among all trees with respect to which a given profile is single-peaked. We provide here alternative proofs for some of these results, based on linear programming arguments.

- Peters recently proposed ILP formulations for proportional representation problems, and showed that the binary constraint matrix is totally unimodular if preferences are single-peaked, because the matrix has then the consecutive ones property [21]. We recall that the vertices of a polyhedron defined by a totally unimodular constraint matrix are all integer, thus solving the linear programming relaxation yields an optimal solution to the original ILP problem. We also rely on linear programming for proving the polynomial time complexity of some of the recognition problems we tackle here.

**Organization of the paper.** The two optimization variants of the recognition problem tackled in the paper are defined in Section 2. The two problems are proved NP-hard, and an ILP formulation is proposed. In Section 3, we consider the continuous relaxation of the ILP to show how to recognize single-peaked preferences on a path or a tree by linear programming, thus providing a unified view of known polynomiality results. Section 4 is devoted to a new tractable case of recognition problem, namely the recognition of single-peaked preferences on a pseudotree. Finally, the results of numerical tests are presented in Section 5, on both real and synthetic data, to give some hints about the kind of graphs that are returned by the ILP on real data, and to study how the density of the graph varies with the number of voters and the diversity of preferences.

## 2 ILP FORMULATION AND COMPLEXITY

### 2.1 Problem Definition

We start by recalling some basic terminology of social choice theory. Given a set  $\mathcal{C} = \{1, 2, \dots, m\}$  of candidates and a set  $\{v_1, \dots, v_n\}$  of voters, each voter  $v_i$  ranks all candidates from the most to the least preferred one. This ranking is called the *preference* of  $v_i$ . It is simply a permutation of  $\mathcal{C}$ , which can be formally described as an  $m$ -tuple  $R_i = (\pi_i(1), \dots, \pi_i(m))$ , where  $\pi_i(k)$  is the  $k$ -th most preferred candidate of voter  $v_i$ . The set  $\mathcal{P} = \{R_1, \dots, R_n\}$  of preferences of all voters is called the *profile*.

As emphasized in the introduction, several definitions of single-peakedness on an arbitrary graph can be found in the literature. In our study, we are using the following one [12]:

**Definition 2.1. Single-peakedness on arbitrary graph (SP)** Let  $\mathcal{C}$  be a set of  $m$  candidates and  $\mathcal{P}$  the profile of preferences of  $n$  voters. Let  $\mathcal{G} = (\mathcal{C}, E)$  be a connected undirected graph. We say that  $\mathcal{P}$  is *single-peaked on the graph  $\mathcal{G}$*  (SP) if every  $R_i \in \mathcal{P}$  is a traversal of  $\mathcal{G}$ , i.e., for each  $R_i \in \mathcal{P}$  and for each  $k \in \{1, \dots, m\}$ , the subgraph of  $\mathcal{G}$  induced by the vertices  $\{\pi_i(1), \dots, \pi_i(k)\}$  is connected.

This notion of single-peakedness coincides with the standard definition on an axis/cycle/tree when  $\mathcal{G}$  is a path/cycle/tree. In this article, when a profile  $\mathcal{P}$  is single-peaked w.r.t. a graph  $\mathcal{G}$ , for conciseness we will say that  $\mathcal{P}$  is *compatible* with  $\mathcal{G}$  (or that  $\mathcal{G}$  is compatible with  $\mathcal{P}$ ).

**Example 1.** Consider the profile with 4 voters and 5 candidates:

$$R_1 : (1, 2, 3, 4, 5)$$

$$R_2 : (1, 3, 4, 2, 5)$$

$$R_3 : (2, 5, 3, 4, 1)$$

$$R_4 : (3, 5, 4, 2, 1)$$

Note that, for  $R_1$ , the connectivity constraint applied to the first two candidates makes the edge  $\{1, 2\}$  necessary in the graph. The same occurs for  $\{1, 3\}$ ,  $\{2, 5\}$  and  $\{3, 5\}$ . Thus, any solution

contains the 4-cycle  $(1, 2, 5, 3, 1)$  (in particular the profile is not SP on a tree or on a cycle). One can easily check that adding edge  $\{3, 4\}$  makes a graph with 5 edges compatible with the profile, and this is the (unique) optimal solution if we want to minimize the number of edges.  $\triangle$

Obviously, any profile is SP on the complete graph. However, this case is not interesting because it does not give any information about the preference structure. That is why we are looking for a *minimal* graph on which the profile is SP. The notion of minimality needs to be made more precise. In our study, we focus essentially on minimizing the number of graph edges. Another criterion we consider is the minimization of the maximum degree of vertices. Put another way, given a preference profile  $\mathcal{P}$ , we want to determine a graph  $\mathcal{G}$  on which the profile is SP, so as to minimize either the number of edges of  $\mathcal{G}$ , or its (maximum) degree. We emphasize the fact that:

- minimizing the number of edges allows to detect when the profile is compatible with a tree (this occurs iff the minimum number of edges is  $m-1$ , since  $\mathcal{G}$  is necessarily connected);
- minimizing the degree of  $\mathcal{G}$  allows to detect when the profile is compatible with a cycle (this occurs iff there exists a graph  $\mathcal{G}$  with maximum degree 2);
- combining the objective allows to detect when the profile is compatible with an axis: this occurs iff there is a graph  $\mathcal{G}$  with maximum degree 2 and  $m-1$  edges.

So the tackled problems generalize the most well known (tractable) recognition problems of single-peakedness.

## 2.2 ILP Formulation

We now present an ILP formulation of the tackled problems. We are looking for a graph  $\mathcal{G}$  with  $m$  vertices. For each pair  $\{k, l\} \subseteq \{1, \dots, m\}$  of vertices, we define a binary variable  $x_{\{k, l\}}$  such that

$$x_{\{k, l\}} = \begin{cases} 1 & \text{if edge } \{k, l\} \text{ is present in graph } \mathcal{G}, \\ 0 & \text{otherwise.} \end{cases}$$

Hence, if we are minimizing the number of graph edges, the objective function  $f(x)$  takes the form

$$f(x) = \sum_{\{k, l\} \subseteq \{1, \dots, m\}} x_{\{k, l\}}.$$

If the objective is to minimize the maximum degree, then

$$f(x) = \max_{k \in \{1, \dots, m\}} \sum_{l=1, l \neq k}^m x_{\{k, l\}}.$$

In this latter case, the classical way of linearizing  $f(x)$  in a minimization setting is to introduce an auxiliary variable  $z$  as follows:

$$\begin{aligned} & \min z \\ & \text{s.t.} \quad \sum_{l=1, l \neq k}^m x_{\{k, l\}} \leq z \quad \forall k \in \{1, \dots, m\} \end{aligned}$$

Regardless of the objective function, the other constraints of the problem remain the same. Each  $R_i = (\pi_i(1), \dots, \pi_i(m))$  for  $i \in \{1, \dots, n\}$  has to be a graph traversal. In other words, for each  $k \in \{2, \dots, m\}$ ,  $\pi_i(k)$  is connected to at least one of the vertices  $\pi_i(1), \dots, \pi_i(k-1)$ . In terms of LP constraints, it is formulated as follows:

$$\sum_{j=1}^{k-1} x_{\{\pi_i(j), \pi_i(k)\}} \geq 1 \quad \forall i \in \{1, \dots, n\}, \forall k \in \{2, \dots, m\}.$$

To sum up, the ILP formulation of the tackled problems is:

$$\begin{aligned} & \min f(x) \\ \text{s.t. } & \begin{cases} \sum_{j=1}^{k-1} x_{\{\pi_i(j), \pi_i(k)\}} \geq 1 \quad \forall i \in \{1, \dots, n\}, \forall k \in \{2, \dots, m\} \\ x_{\{k, l\}} \in \{0, 1\} \quad \forall \{k, l\} \subseteq \{1, \dots, m\} \end{cases} \end{aligned}$$

**Example 2.** Consider profile  $\mathcal{P} = \{(1, 4, 2, 3), (3, 4, 2, 1)\}$ . The ILP formulation of the problem of determining a graph  $\mathcal{G}$  compatible with  $\mathcal{P}$  while minimizing the number of graph edges is:

$$\begin{aligned} & \min x_{\{1,2\}} + x_{\{1,3\}} + x_{\{1,4\}} + x_{\{2,3\}} + x_{\{2,4\}} + x_{\{3,4\}} \\ \text{s.t. } & \begin{cases} x_{\{1,4\}} \geq 1 \\ x_{\{1,2\}} + x_{\{2,4\}} \geq 1 \\ x_{\{1,3\}} + x_{\{3,4\}} + x_{\{2,3\}} \geq 1 \\ x_{\{3,4\}} \geq 1 \\ x_{\{2,3\}} + x_{\{2,4\}} \geq 1 \\ x_{\{1,3\}} + x_{\{1,4\}} + x_{\{1,2\}} \geq 1 \\ x_{\{k,l\}} \in \{0, 1\} \quad \forall \{k, l\} \subset \{1, 2, 3, 4\} \end{cases} \end{aligned}$$

△

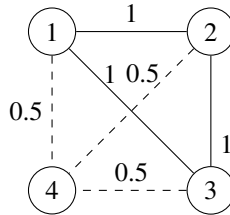
### 2.3 Minimizing the Number of Edges

In this section, we study the computational complexity of the problem of minimizing the number of edges of  $\mathcal{G}$ . As a first observation, note that we cannot use the continuous relaxation of this ILP ( $x_{\{k,l\}} \in [0, 1]$ ) to solve the problem. The following example indeed shows that the optimal solution (when minimizing the number of edges) of this relaxation is not necessarily integer:

**Example 3.** Consider the profile with 3 voters and 4 candidates:

$$\begin{aligned} R_1 &: (1, 2, 4, 3) \\ R_2 &: (2, 3, 4, 1) \\ R_3 &: (1, 3, 4, 2) \end{aligned}$$

From the two first options of each voter, we see immediately that the edges  $\{1, 2\}$ ,  $\{2, 3\}$  and  $\{1, 3\}$  are necessarily present in the graph. Then, we observe that vertex 4 needs to be connected to at least one of vertices 1 and 2, at least one of vertices 2 and 3 and finally at least one of vertices 1 and 3. Consequently any integer solution of the problem will be a graph with at least 5 edges. However, there exists a fractional solution of the continuous relaxation with value 4.5:



△

We now show that the problem is actually NP-hard.

**Theorem 2.1.** Given a preference profile  $\mathcal{P}$ , it is NP-hard to find a graph compatible with  $\mathcal{P}$  with a minimum number of edges.

*Proof.* We use a polynomial time reduction from the set cover problem, known to be NP-hard [16]. We recall its definition:

*Set cover problem:* Given a finite set  $\mathcal{U} = \{e_1, \dots, e_n\}$  of elements, a set  $\mathcal{S} = \{S_1, \dots, S_m\}$  of subsets of  $\mathcal{U}$  and  $k \in \mathbb{N}$ , the question is to determine if there exists a subset  $\mathcal{C} \subseteq \mathcal{S}$  of size  $k$  such that  $\cup_{S \in \mathcal{C}} S = \mathcal{U}$ .

From an instance of the set cover problem, we define a preference profile  $\mathcal{P}$  as follows:

- (i) Let  $\{S_1, \dots, S_m, z\}$  be a set of candidates.
- (ii) Let  $\{v_1, \dots, v_n\}$  be a set of voters. Let  $S_{i_1}, \dots, S_{i_l}$  be the subsets in  $\mathcal{S}$  containing element  $e_i \in \mathcal{U}$ , and  $S_{i_{l+1}}, \dots, S_{i_m}$  the other subsets in  $\mathcal{S}$ . Then, the preference of voter  $v_i$  is defined as

$$R_i = (S_{i_1}, \dots, S_{i_l}, z, S_{i_{l+1}}, \dots, S_{i_m}).$$

- (iii) We add  $\frac{m \cdot (m-1)}{2}$  voters  $v_{\{i,j\}}$ ,  $\{i,j\} \subseteq \{1, \dots, m\}$  such that

$$R_{\{i,j\}} = (S_i, S_j, \underbrace{S_1, \dots, S_m}_{\text{except } S_i, S_j}, z).$$

We prove that there exists a set cover of size  $k$  if and only if there exists a graph  $\mathcal{G}$  compatible with  $\mathcal{P}$  that has  $\frac{m \cdot (m-1)}{2} + k$  edges.

Let  $\mathcal{C}$  be a set cover solution of size  $k$ . We generate a graph  $\mathcal{G}$  compatible with  $\mathcal{P}$  in the following manner:

- a) For each  $\{i,j\} \in \{1, \dots, m\}^2, i \neq j$ , the edge  $\{S_i, S_j\}$  is in  $\mathcal{G}$  - this is necessary for the preferences of type (iii) above to be SP on  $\mathcal{G}$ .
- b) For each  $i \in \{1, \dots, m\}$ , the edge  $\{S_i, z\}$  is in  $\mathcal{G}$  if and only if  $S_i \in \mathcal{C}$ .

Hence, the subgraph formed by vertices  $\{S_1, \dots, S_m\}$  is a clique having  $\frac{m \cdot (m-1)}{2}$  edges, and there are exactly  $k$  more edges adjacent to  $z$  - in total,  $\mathcal{G}$  has  $\frac{m \cdot (m-1)}{2} + k$  edges. As  $k > 0$ , the graph is connected and all preferences of type (iii) are SP on  $\mathcal{G}$ . Let  $R_i$  be one of the preferences of type (ii). We need to prove that  $z$  is connected to at least one of the vertices  $S_{i_1}, \dots, S_{i_l}$ . As the sets  $S_{i_1}, \dots, S_{i_l}$  are the only sets of  $\mathcal{S}$  containing the element  $e_i$ , and as  $\mathcal{C}$  is a solution of the set cover instance, this is true due to b). So,  $\mathcal{G}$  is a graph compatible with  $\mathcal{P}$  that has  $\frac{m \cdot (m-1)}{2} + k$  edges.

To prove the other implication, let  $\mathcal{G}$  be a graph compatible with  $\mathcal{P}$  that has  $\frac{m \cdot (m-1)}{2} + k$  edges. As  $\mathcal{G}$  is compatible with  $\mathcal{P}$ , the subgraph induced by the set of vertices  $\{S_1, \dots, S_m\}$  must be a clique so that the preferences  $R_{\{i,j\}}$  of type (iii) are SP on  $\mathcal{G}$ . Hence, this subgraph contains  $\frac{m \cdot (m-1)}{2}$  edges, and so, there are exactly  $k$  edges adjacent to  $z$ . Let us define  $\mathcal{C}$  containing  $S_i$  iff  $S_i$  is adjacent to  $z$  in  $\mathcal{G}$ . As  $\mathcal{G}$  is compatible with  $\mathcal{P}$ , each preference  $R_i$  of type (ii) is SP on  $\mathcal{G}$ . It means that at least one of  $S_{i_1}, \dots, S_{i_l}$  is adjacent to  $z$ , so is in  $\mathcal{C}$ . As all these sets contains  $e_i$ , there is an element of  $\mathcal{C}$  that covers  $e_i$ . The subset  $\mathcal{C} \subseteq \mathcal{S}$  is thus a solution of size  $k$  of the set cover instance.  $\square$

## 2.4 Minimizing the Maximum Degree

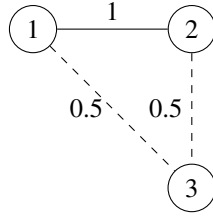
We now consider our second objective function, namely the maximum degree of a vertex in the graph (to be minimized). We come up with similar results.

First, as for the minimization of the number of edges, the ILP formulation we have proposed in Section 2.2 is not integer, as we can see in the following case:

**Example 4.** Consider a profile with 3 candidates and one voter with ranking  $(1, 2, 3)$ . The ILP formulation of the problem of determining a graph  $\mathcal{G}$  of minimum degree compatible with  $\mathcal{P}$ :

$$\begin{aligned} \min z \\ \text{s.t.} \quad & \begin{cases} x_{\{1,2\}} + x_{\{1,3\}} \leq z \\ x_{\{1,2\}} + x_{\{2,3\}} \leq z \\ x_{\{1,3\}} + x_{\{2,3\}} \leq z \\ x_{\{1,2\}} \geq 1 \\ x_{\{1,3\}} + x_{\{2,3\}} \geq 1 \\ x_{\{k,l\}} \in \{0, 1\} \quad \forall \{k, l\} \subset \{1, 2, 3\} \end{cases} \end{aligned}$$

The value of an optimal integer solution is  $z = 2$ , but there exists a fractional solution of the continuous relaxation with value 1.5:



△

Here again, we show that the problem of minimizing the degree of  $\mathcal{G}$  is NP-hard, by a similar reduction sketched below.

**Theorem 2.2.** Given a preference profile  $\mathcal{P}$ , it is NP-hard to find a graph compatible with  $\mathcal{P}$  with a minimum degree.

*Proof.* Let  $\mathcal{U} = \{e_1, \dots, e_n\}$ ,  $\mathcal{S} = \{S_1, \dots, S_m\}$ ,  $k \in \mathbb{N}$  be an instance of the set cover problem. Consider the profile  $\mathcal{P}$  defined as follows:

- (i) Let  $\{S_1, \dots, S_m, z, t_1, \dots, t_m\}$  be a set of candidates.
- (ii) Let  $\{v_1, \dots, v_n\}$  be a set of voters. Let  $S_{i_1}, \dots, S_{i_l}$  be the subsets in  $\mathcal{S}$  containing the element  $e_i \in \mathcal{U}$ , and  $S_{i_{l+1}}, \dots, S_{i_m}$  the other subsets in  $\mathcal{S}$ . Then, the preference of voter  $v_i$  is defined as

$$R_i = (S_{i_1}, \dots, S_{i_l}, z, S_{i_{l+1}}, \dots, S_{i_m}, t_1, \dots, t_m).$$

- (iii) We add  $\frac{m \cdot (m-1)}{2}$  voters  $v_{\{i,j\}}$ ,  $\{i, j\} \subseteq \{1, \dots, m\}$  such that

$$R_{\{i,j\}} = (S_i, S_j, \underbrace{S_1, \dots, S_m}_{\text{except } S_i, S_j}, z, t_1, \dots, t_m).$$

(iv) We add  $m$  voters  $v^{t_1}, \dots, v^{t_m}$  where the preference of  $v^{t_i}$  is defined as

$$R^{t_i} = (z, t_i, \underbrace{t_1, \dots, t_m}_{\text{except } t_i}, S_1, \dots, S_m),$$

and a voter  $w$  with preference  $(t_1, S_1, S_2, \dots, S_m, z, t_2, \dots, t_m)$ .

Note that in any graph  $\mathcal{G}$  compatible with the profile:

- a) the  $S_i$  form a clique (edge  $\{S_i, S_j\}$  is enforced by voter  $v_{\{i,j\}}$ ),
- b)  $z$  is adjacent to all  $t_i$  (due to voter  $v^{t_i}$ ),
- c)  $\{S_1, t_1\}$  is in the graph (due to  $w$ ).

We call these edges necessary edges. We claim that there exists a set cover of size (at most)  $k$  iff there is a graph compatible with the profile with degree at most  $m + k$ .

Suppose that there is a set cover  $\mathcal{C}$  of size at most  $k$ . Then, beyond the necessary edges mentioned above, we put an edge  $\{S_i, z\}$  iff  $S_i$  is in  $\mathcal{C}$ . Then the vertex with maximum degree is  $z$ , with degree  $m + k$ . The graph  $\mathcal{G}$  is compatible with each voter  $v_i$  because  $\mathcal{C}$  is a set cover. It is compatible with voter  $v_{\{i,j\}}$  thanks to the necessary edges (and  $k > 0$ , so  $z$  is connected as well). It is compatible with  $v^{t_i}$  and  $w$  thanks to the necessary edges.

Now suppose that there is a solution  $\mathcal{G}$  with degree at most  $m + k$ . In particular,  $z$  has degree at most  $m + k$ , hence is adjacent to at most  $k$  vertices  $S_i$ . The preference of voter  $v_i$  imposes that  $z$  is adjacent to some  $S_i$  which contains  $e_i$ . In other words, the set  $\mathcal{C}$  of these (at most)  $k$  sets  $S_i$  is a set cover of size at most  $k$ .  $\square$

### 3 RECOGNITION OF TREES AND PATHS

In this section, we focus on the tree and path recognition - given a profile  $\mathcal{P}$ , we are looking for a tree (or a path) on which the profile is SP.

Recognizing single-peaked preferences on a tree can be done using the procedure by Trick [24]. As an alternative proof of this result, we show in this section that the continuous relaxation of the ILP formulations given in Section 2.2 can be used to solve this recognition problem in polynomial time: in fact, all (optimal) extremal solutions are integral (Theorem 3.1). We show in Theorem 3.2 a similar result for the recognition of profiles SP on a path.

We start by recalling Trick's procedure [24], as we will use it in the proof of the results of the two theorems mentioned above.

**Recognition of profiles SP on a tree [24]** Let  $\mathcal{P}$  be a profile containing preference lists  $\{R_1, \dots, R_n\}$  of  $n$  voters over  $m$  candidates. The algorithm of Trick builds a tree in an iterative way. In each iteration, it identifies one candidate (i.e., graph vertex) which is necessarily a leaf, and it determines the set of vertices this leaf can be connected to. It deletes then this leaf from all preferences, and repeats this process on the modified profile with preferences over  $m-1$  candidates. The algorithm stops when only one candidate is remaining. Let us describe in more details the leaf recognition :

Let  $k$  be a candidate placed at the last position by at least one voter. Trick shows that, if preferences are SP on a tree, then  $k$  must necessarily be a leaf. More formally, for each  $i \in \{1, \dots, n\}$ , let us denote by  $A(k)_i$  the set of candidates ranked better than  $k$  by voter  $v_i$  if  $k$  is not ranked first by  $v_i$ ; if  $k$  is ranked first by  $v_i$ , then  $A(k)_i$  is the singleton containing the second most-preferred candidate of  $v_i$ . From  $A(k) = \bigcap_{i=1}^n A(k)_i$ , the following conclusions can be drawn:

- if  $A(k) = \emptyset$ , there does not exist a tree solution.



- Otherwise,  $A(k)$  is the set of vertices the leaf  $k$  can be connected to.

**Example 5.** Consider the profile  $\mathcal{P} = \{R_1, R_2, R_3\}$  defined by:

$$R_1 : (1, 2, 3, 4)$$

$$R_2 : (2, 1, 3, 4)$$

$$R_3 : (4, 1, 2, 3)$$

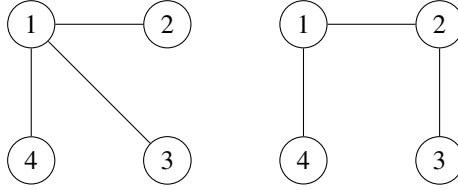
1. The candidate 4 is classed at the last position by at least one voter - we will determine the set  $A(4)$ :

$$A(4) = \{1, 2, 3\} \cap \{1, 2, 3\} \cap \{1\} = \{1\}$$

The candidate 4 is then deleted from the preference lists and we continue next iteration with the sub-profile  $R_1^1 = (1, 2, 3)$ ,  $R_2^1 = (2, 1, 3)$  and  $R_3^1 = (1, 2, 3)$ .

2. The candidate 3 is classed at the last position by at least one voter - we see that  $A(3) = \{1, 2\}$ . We continue with the sub-profile  $R_1^2 = (1, 2)$ ,  $R_2^2 = (2, 1)$ ,  $R_3^2 = (1, 2)$ .
3. We get  $A(2) = \{1\}$ , and the algorithm stops as we obtain a sub-profile involving only one candidate.

To sum up, we have obtained  $A(2) = \{1\}$  (first iteration),  $A(3) = \{1, 2\}$  (second iteration) and  $A(4) = \{1\}$  (third iteration). Consequently, in any tree compatible with  $\mathcal{P}$ , vertex 2 and vertex 4 have to be connected to vertex 1, and vertex 3 has to be connected to vertex 1 or 2. Hence, there exists two trees on which profile  $\mathcal{P}$  is SP, and these are:



△

**Using LP to recognize SP preferences on a tree or a path** Let us consider the following continuous relaxation LP-SP (linear program for single-peakedness) of the ILP introduced in Section 2.2:

$$\begin{aligned}
 \min \quad & \sum_{\{k,l\} \subseteq \{1,\dots,m\}} x_{\{k,l\}} \\
 \text{s.t.} \quad & \begin{cases} \sum_{j=1}^{k-1} x_{\{\pi_i(j), \pi_i(k)\}} \geq 1 \quad \forall i \in \{1, \dots, n\}, k \in \{2, \dots, m\} \\ x_{\{k,l\}} \in [0, 1] \quad \forall \{k,l\} \subseteq \{1, \dots, m\} \end{cases}
 \end{aligned}$$

We show in Theorem 3.1 that we can use LP-SP to solve in polynomial time the problem to determine, given a profile, whether there exists or not a tree compatible with it.

**Theorem 3.1.** If a profile  $\mathcal{P}$  is compatible with a tree, then any extremal optimal solution  $x$  of LP-SP is integral, i.e.,  $x_{\{k,l\}} \in \{0, 1\}$  for any  $\{k,l\} \subseteq \{1, \dots, m\}$ .

*Proof.* The proof is based on two properties of optimal solutions of LP-SP when the profile is compatible with a tree. These two properties allow to come up with a reformulation of the problem as a maximum flow problem, where there is a bijection between the solutions of LP-SP of value  $m - 1$  and the (optimal) flows of value  $m - 1$ . The result then comes from the fact that any extremal solution of the flow problem (with integral capacity) is integral [1].

The first property states that all constraints of LP-SP are tight in a solution of value  $m - 1$ .

**Property 1.** If the optimal value of LP-SP is  $m-1$ , then all constraints are tight in an optimal solution  $x^*$ :  $\sum_{j=1}^{k-1} x_{\{\pi_i(j), \pi_i(k)\}}^* = 1$ .

*Proof of Property 1.* Let  $v_i$  be a voter. There are  $m-1$  constraints associated with  $v_i$ , and each variable  $x_{\{k,l\}}$  appears in exactly one of these constraints. Since on the one hand the sum of all variables is  $m-1$  (objective function), and on the other hand the sum of variables in each of these  $m-1$  constraints is at least one, each constraint must be tight. This concludes the proof of Property 1.

Now, let us consider that the profile is SP with respect to a tree. The recognition procedure recalled above starts by identifying a candidate, say  $m$ , ranked last in at least one ranking and such that  $A(m) \neq \emptyset$ . This procedure is then applied recursively, till there is only one candidate. For simplicity, let us assume that the first removed (identified) candidate is  $m$ , the second  $m-1$ , and so on. To avoid confusion, we denote  $B(k)$  the set  $A(k)$  when considering the profile restricted to the first  $k$  candidates (when  $k$  is identified as a leaf, and then removed from the profile).

**Property 2.** If the profile is SP on a tree, then in an optimal solution of LP-SP, for any candidate  $k \geq 2$  we have  $\sum_{j \in B(k)} x_{\{j,k\}} = 1$ , and  $x_{\{j,k\}} = 0$  for any  $j \in \{1, \dots, k-1\} \setminus B(k)$ .

*Proof of Property 2.* Let us consider some candidate  $k \geq 2$ , some optimal solution  $X$  of LP-SP, and assume that the properties in the lemma are true for any  $k' > k$ . We will show that they are also true for  $k$ .

To do this, let us define LP-SP( $k$ ) as the linear program corresponding to the problem restricted to the candidates  $1, 2, \dots, k$ . We first need to show that the optimal solution  $X$  restricted to the first  $k$  candidates, let us call it  $X_k$ , is feasible and optimal for LP-SP( $k$ ). To do this, let us consider a constraint of LP-SP( $k$ ), let us say the constraint of connecting candidate  $j \leq k$  for some voter  $i$ . In the corresponding constraint in the initial program LP-SP, there are possibly some other variables  $x_{k',j}$ : there is such variable  $x_{k',j}$  for each candidate  $k' > k$  ranked before  $j$  by  $i$ . But then when  $k'$  was removed, if  $x_{k',j}$  appears in the constraint then  $j \notin B(k')$ , and then  $x_{k',j} = 0$ . So all ‘removed variables’ in the constraint was set to 0 in  $X$ , hence  $X_k$  is feasible for LP-SP( $k$ ). We can now easily see that it is optimal: each time a candidate  $k' > k$  has been removed,  $\sum_{j \in B(k')} x_{j,k} = 1$  so the total weights of (remaining) variables reduce by 1. Thus  $X_k$  is a feasible solution of LP-SP( $k$ ) of value  $k-1$ .

Now we can focus on  $X_k$  on LP-SP( $k$ ). Note that the profile is trivially SP on the first  $k$  candidates (as it is SP on the whole set of candidates). Candidate  $k$  is ranked in last position by some voter  $v_i$ , so we have  $\sum_{j=1}^{k-1} x_{\{j,k\}} \geq 1$  (constraint of connecting  $k$  for voter  $v_i$ ), and by Property 1 we have  $\sum_{j=1}^{k-1} x_{\{j,k\}} = 1$ . If all candidates  $1, 2, \dots, k-1$  are in  $B(k)$  then we are done. Otherwise, consider a candidate  $\ell \notin B(k)$ . Then  $\ell$  is ranked after  $k$  by some other voter  $v_j$  (and, for this voter  $v_j$ ,  $k$  and  $\ell$  are not the best two candidates). Then, if  $x_{\{\ell,k\}} > 0$  we get  $\sum_{j \notin \{\ell,k\}} x_{\{j,k\}} = 1 - x_{\{\ell,k\}} < 1$ , and the constraint associated to  $v_j$  for connecting  $k$  to its predecessors is violated. So  $x_{\{\ell,k\}} = 0$  for any  $\ell \notin B(k)$ , and consequently  $\sum_{j \in B(k)} x_{\{j,k\}} = 1$ . This concludes the proof of Property 2.

Now we reformulate the problem as a flow problem. From  $\mathcal{P}$ , we build a network (directed graph)  $R$  with:

- A source  $s$ , a destination  $t$ , and for each candidate  $k$  two vertices  $\ell_k$  and  $r_k$ .
- We have an arc from  $s$  to each  $\ell_k$  with capacity 1, and an arc from each  $r_k$  to  $t$  with capacity  $\infty$ .
- For each candidate  $k$ , we have an arc  $(\ell_k, r_j)$  for each  $j < k$ . The capacity of this arc is 1 if  $j \in B(k)$ , and 0 otherwise.

Let us denote by  $\phi$  a flow on this network, with  $\phi(e)$  the flow on edge  $e$ . Note that  $\ell_1$  has no outgoing edge, so the optimal flow is at most  $m-1$ .

We show that the correspondence  $x_{\{k,j\}} = \phi(k,j)$  (for each  $j < k$ ) is a bijection between solutions of value  $m-1$  of LP-SP and (optimal) flows of value  $m-1$  in  $R$ .

Let  $\phi$  be a flow of value  $m-1$ . As there is no flow through  $\ell_1$ , there is a flow of value 1 through each  $\ell_k, k > 1$ . Since arc  $(k,j)$  has capacity 0 if  $j \notin B(k)$ , by flow conservation we have  $\sum_{j \in B(k)} \phi(k,j) = 1$ , which means that  $\sum_{j \in B(k)} x_{\{k,j\}} = 1$ . Now consider a voter  $v_i$  where  $k$  is not ranked first. By the procedure of Trick, when  $k$  is identified as a leaf, all candidates in  $B(k)$  are ranked before  $k$ , and the corresponding constraint is satisfied. This is true for all candidates and voters, so  $x$  is a feasible solution of LP-SP, of value  $m-1$ .

Conversely, let  $x$  be a feasible solution of LP-SP of value  $m-1$ . From Property 2, we have  $\sum_{j \in B(k)} x_{\{j,k\}} = 1$  for each candidate  $k \geq 2$ . This immediately gives a flow of value  $m-1$ .

By integrality of extremal flows (any non integral optimal flow is a convex combination of integral flows), any extremal optimal solution of LP-SP is integral (when there exists a tree compatible with  $\mathcal{P}$ ).  $\square$

Let us now turn to the recognition of profiles SP on a path. A (connected) graph is a path iff it is a tree with degree at most 2. Hence, we consider the following ILP formulation where we minimize the number of edges and add constraints on the vertex degrees:

$$\begin{aligned} \min \quad & \sum_{\{k,l\} \subseteq \{1,\dots,m\}} x_{\{k,l\}} \\ \text{s.t.} \quad & \begin{cases} \sum_{j=1}^{k-1} x_{\{\pi_i(j), \pi_i(k)\}} \geq 1 \quad \forall i \in \{1, \dots, n\}, k \in \{2, \dots, m\} \\ \sum_{l=1, l \neq k}^m x_{\{k,l\}} \leq 2 \quad \forall k \in \{1, \dots, m\} \\ x_{\{k,l\}} \in \{0, 1\} \quad \forall \{k, l\} \subseteq \{1, \dots, m\} \end{cases} \end{aligned}$$

Clearly, a profile is compatible with a path iff the optimal value of the previous ILP is  $m-1$ . Let us call LP-SP2 the continuous relaxation. By using very similar arguments as above (same reformulation as a flow problem), one can prove the following result.

**Theorem 3.2.** If a profile  $\mathcal{P}$  is compatible with a path, then any extremal optimal solution of LP-SP2 is integral, i.e.,  $x_{\{k,l\}} \in \{0, 1\}$  for any  $\{k, l\} \subseteq \{1, \dots, m\}$ .

## 4 RECOGNITION OF PSEUDOTREES

So far, we have seen that our minimization problem is NP-hard in the general case, but polynomially solvable in the case where the optimal solution is a tree. As a natural extension, we consider the problem to recognize profiles that are single-peaked with respect to a graph with  $m-1+k$  edges, for some fixed  $k$ , thus allowing  $k$  more edges than in a tree. In this section, we consider the case  $k = 1$ . A graph on  $m$  vertices with  $m$  edges is called a pseudotree. We show that recognizing if there exists a pseudotree compatible with a given profile can be done in polynomial time. We leave as open question the parameterized complexity of the problem when  $k$  is the parameter: would the problem be in XP? Or even in FPT?

Let us now deal with the case of pseudotree. Hence, the set of solutions we want to recognize is the class of connected graphs having (at most)  $m$  edges. To solve the problem in polynomial time, we devise an algorithm that first identifies the leaves of the pseudotree and then the cycle on the remaining vertices. The second step (cycle recognition) is done using the polynomiality of recognizing single-peakedness on a cycle [23]. For the first step, we need to modify the procedure

of recalled in Section 3. This procedure was able to correctly identify leaves when the profile was compatible with a tree, but it fails to correctly identify leaves when the underlying structure is a pseudotree. With a slight modification though, we obtain in Proposition 4.1 a necessary and sufficient condition for a candidate to be a leaf in a pseudotree. This is the stepping stone leading to the polynomiality of detecting whether a given profile is compatible with a pseudotree, stated in Theorem 4.1.

**Example 6.** Let us consider the profile on 4 voters and 5 candidates given in Example 1, for which there is a (unique) pseudotree compatible with it.

The procedure to detect leaves when looking for a tree focuses on candidates ranked last in some  $R_i$ , candidates 1 and 5 here, and  $A(1) = A(5) = \emptyset$ . Note that the whole profile is not compatible with a cycle, so we need somehow to first detect 4 as a leaf, and then detect that the candidates 1, 2, 3, 5 are SP with respect to a cycle.  $\triangle$

The central property that allows to recognize profiles compatible with a pseudotree is given in the following proposition.

**Proposition 4.1.** Let  $\mathcal{P}$  be a preference profile, and suppose that a candidate  $i$  is such that  $A(i) \neq \emptyset$ . Then  $\mathcal{P}$  is compatible with a pseudotree if and only if it is compatible with a pseudotree where  $i$  is a leaf.

*Proof.* Let  $\mathcal{G}$  be a pseudotree compatible with  $\mathcal{P}$  where  $i$  is not a leaf. We transform  $\mathcal{G}$  into a pseudo-tree  $\mathcal{G}'$  compatible with  $\mathcal{P}$  where  $i$  is a leaf. Let  $j \in A(i)$ .

**Case 1:**  $\{i, j\} \in \mathcal{G}$ . Let us first consider an easy case, where  $\{i, j\} \in \mathcal{G}$ . Then we build  $\mathcal{G}'$  from  $\mathcal{G}$  by simply replacing each edge  $\{i, k\}$  (with  $k \neq j$ ) by the edge  $\{j, k\}$ . Since for each voter either  $j$  is ranked before  $i$ , or  $i$  is first and  $j$  second, then this modification creates a graph  $\mathcal{G}'$  compatible with all the preferences. Note that  $\mathcal{G}'$  has (at most) as many edges as  $\mathcal{G}$ , so it is a pseudotree (or a tree, and we can add any edge to create a pseudotree).

**Case 2:**  $\{i, j\} \notin \mathcal{G}$ . Let us now consider the case where  $\{i, j\} \notin \mathcal{G}$ . Note that then  $j$  is ranked before  $i$  in all preferences (otherwise  $i$  is first and  $j$  is second, and the edge  $\{i, j\}$  is forced to be in any compatible graph, a contradiction). Then we transform  $\mathcal{G}$  into a graph  $\mathcal{G}'$  which is a pseudotree containing the edge  $\{i, j\}$ , and then Case 1 applies to  $\mathcal{G}'$ . To do this, let us consider two subcases.

**Case 2a.** If, in  $\mathcal{G}$ , in all (simple) paths from  $j$  to  $i$  the predecessor of  $i$  is the same vertex  $u$ . Then we create  $\mathcal{G}'$  by replacing the edge  $\{u, i\}$  by the edge  $\{j, i\}$ . Consider a voter  $v$ . Since  $j$  is ranked before  $i$  by  $v$ , then  $u$  is ranked before  $i$  by  $v$  (the subgraph induced by  $i$  and the candidates ranked before him by  $v$  is connected and contains  $i$  and  $j$ , so it contains  $u$ ). Then the modification does not affect  $u$  (it is still connected to one of the candidates ranked before him), and  $i$  is now connected to  $j$ .

**Case 2b.** In the other case, in  $\mathcal{G}$  there are two simple paths from  $j$  to  $i$  such that the predecessor of  $i$  is  $u_1$  in the first one and  $u_2 \neq u_1$  in the second one (note that there cannot be more than 2 since  $\mathcal{G}$  is a pseudotree). We build  $\mathcal{G}'$  from  $\mathcal{G}$  by deleting the edges  $\{u_1, i\}$  and  $\{u_2, i\}$ , and adding edges  $\{i, j\}$  and  $\{u_1, u_2\}$ , see Figure 4.

Consider a voter  $v$ . Since  $v$  prefers  $j$  to  $i$  and the subgraph of  $\mathcal{G}$  induced by the candidates up to  $i$  in the ranking of  $v$  is connected, then  $u_1$  or  $u_2$  is ranked before  $i$  by  $v$ , say  $u_1$  (we assume wlog that  $u_1$  is preferred to  $u_2$  by  $v$ ). Then we see that  $\mathcal{G}'$  is compatible with the preference of  $v$ : indeed, when considering candidates one by one in the order of  $v$ , the only modification holds for  $u_2$ , which is now connected to  $u_1$  (ranked before him), and for  $i$ , which is now connected to  $j$  (ranked before him).  $\square$

Note that  $i$  can be connected to any vertex  $j \in A(i)$ .

Before giving the procedure that recognizes preferences compatible with a pseudotree, we need to establish another property regarding such preferences.

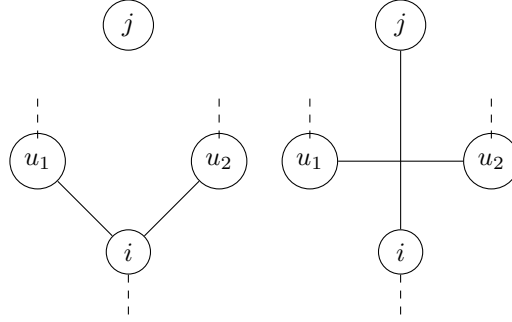


Figure 1: Edge swap

**Proposition 4.2.** If a preference profile  $\mathcal{P}$  is compatible with a pseudotree, then either there exists a candidate  $i$  such that  $A(i) \neq \emptyset$ , or  $\mathcal{P}$  is compatible with a cycle.

If  $\mathcal{P}$  is not compatible with a cycle, then there exists a leaf  $i$  in a pseudotree  $\mathcal{G}$  compatible with  $\mathcal{P}$ . Let  $j$  be the unique neighbor of  $i$  in  $\mathcal{G}$ . Suppose that  $j \notin A(i)$ . Let  $v$  be a voter who prefers  $i$  to  $j$ , and  $j$  is not second if  $i$  is first in the ranking of  $v$ . If  $i$  is not first, then the subgraph induced by the candidates up to  $i$  (in the ranking of  $v$ ) is not connected. If  $i$  is first (and  $j$  not second), the subgraph induced by the first two candidates (in the ranking of  $v$ ) is not connected. Contradiction.

Consider now the following procedure DETECT\_PSEUDOTREE that detects pseudotree-singlepeakedness:

1. Set  $E' = \emptyset$
2. While there are at least 4 candidates, and a candidate  $i$  such that  $A(i) \neq \emptyset$ :
  - (a) Add edge  $\{i, j\}$  to  $E'$  for some  $j \in A(i)$ .
  - (b) Remove  $i$  from the profile.
3. Detect if there is a cycle  $C$  which is compatible with the (remaining) profile:
  - If YES: output  $E' \cup C$
  - If NO: output NO.

**Theorem 4.1.** Given a preference profile  $\mathcal{P}$  on at least 3 candidates, DETECT\_PSEUDOTREE is a polytime procedure which outputs a pseudotree compatible with  $\mathcal{P}$  if some exists, and outputs NO otherwise.

*Proof.* DETECT\_PSEUDOTREE obviously runs in polynomial time. We proceed by induction on the number of candidates. If there are three candidates the procedure outputs a cycle on these 3 candidates. Now suppose that the result is true up to  $m - 1$  candidates, and consider a profile  $\mathcal{P}$  on  $m \geq 4$  candidates.

Suppose that  $\mathcal{P}$  is compatible with a pseudotree  $\mathcal{G}$ .

- If there exists a candidate  $i$  with  $A(i) = \emptyset$ , then by Proposition 4.1, there exists a pseudotree  $\mathcal{G}'$  compatible with  $\mathcal{P}$  where  $i$  is a leaf. Then the profile obtained from  $\mathcal{P}$  by removing  $i$  is compatible with a pseudotree  $(\mathcal{G}' \setminus i)$ , and adding the edge  $\{j, i\}$  as done by DETECT\_PSEUDOTREE gives a pseudotree compatible with  $\mathcal{P}$ .

- Otherwise, by Proposition 4.2,  $\mathcal{P}$  is compatible with a cycle, which is found by DETECT\_PSEUDOTREE (Step 3).

Suppose now that DETECT\_PSEUDOTREE does not output NO. If there were no candidate  $i$  with  $A(i) \neq \emptyset$ , then  $\mathcal{P}$  is compatible with the cycle  $C$ . Otherwise, let  $i$  be the candidate in the first iteration of the loop in Step 2 ( $A(i) \neq \emptyset$ ). Then, on the profile  $\mathcal{P}$  where  $i$  is removed, DETECT\_PSEUDOTREE outputs a pseudotree, compatible with this profile without  $i$  by induction. Since  $j \in A(i)$ , adding edge  $\{j, i\}$  makes the pseudotree compatible with  $\mathcal{P}$ .  $\square$

Note that the generalization of this polynomiality result to, say, connected graphs with  $(n-1+k)$  edges seems to require new techniques (even for fixed  $k$ , i.e. to show that the problem is in XP when parameterized by  $k$ ). Indeed, an enumeration of all subsets of  $k$  edges does not allow to reduce the problem to trees. Procedure DETECT\_PSEUDOTREE does not seem to generalize as well, as it specifically relies on the decomposition of the solution into one cycle and leaves.

## 5 EXPERIMENTAL STUDY

We carried out numerical experiments<sup>1</sup> on real and randomly generated instances of the problems tackled in the paper. In the case of real data, we compare the optimal solution of the ILP to that of its continuous relaxation. We also focus on the ability to detect structure in voters' preferences depending on the election context. To go further, we use randomly generated instances to study structural aspects of solutions; we notably study the graph density depending on the number of voters and on the dispersion of their opinions.

### 5.1 Numerical tests on real data

We used PrefLib data sets available on [www.preflib.org](http://www.preflib.org) to perform our numerical tests [18]. While this database offers four different types of data, only the ED (Election Data) type is relevant for our study. Among the ED data sets, we used the complete strict order lists (which correspond to files with `.soc` extension).

At the time we carried out these experiments, 315 data files of this type were available in PrefLib, however, many of them were not adapted to our study for several reasons. The first one is that many elections dealt with only 3 or 4 candidates and a great number of voters, hence the obtained graph was, unsurprisingly, always complete. We also met the opposite problem when there were very few voters, typically 4, so there was no point in looking for some general structure. Thus, in practice, there were 25 real data files usable for our purposes, namely:

- 20 files from the ED-00006 data set, which contains figure skating rankings from various competitions during the 1998 season including the World Juniors, World Championships, and the Olympics.
- 2 files from the ED-00009 data set, which contains the results of surveying students at AGH University of Science and Technology (Krakow, Poland) about their course preferences.
- 1 file from the ED-00012 data set, where individuals ranked T-shirt designs.
- 1 file from the ED-00014 data set, which contains preferences about various kinds of sushi (surveys conducted by Toshihiro Kamishima).

---

<sup>1</sup>All tests were performed on a Intel Core i7-1065G7 CPU with 8 GB of RAM under the Windows operating system. We used the IBM Cplex solver for the solution of ILPs.

- 1 file from the ED-00032 data set, which contains the results of surveying students in the Faculty of Informatics, Instituto Superior Politécnico José Antonio Echeverría (Cujae, Havana, Cuba), about the most important criteria affecting their performances as students.

We now present the results obtained for these 25 instances. The tackled optimization problem was to determine a graph with a *minimal number of edges*. For all instances considered here, an optimal graph has been obtained in about 40 milliseconds for the ILP formulation and 20 milliseconds for its LP relaxation. In fact, the linear programming formulation always returned an integer solution. Table 2 summarizes the obtained results.

**ED-00006 data set.** The number of candidates (skaters) varies from 14 to 30, and the number of voters (judges) from 7 to 9. For the 20 instances considered, a tree has been obtained 14 times, a pseudotree 5 times, and a solution with 23 edges for 19 candidates (vertices) has been obtained once. The possible interpretation of these results is that, even though the rankings are based on subjective opinions of the judges, there is something like a “true ranking” behind as some skaters are objectively better than other ones. Thus, the rankings given by the judges can be viewed as biased observations of the true ranking, so that they are quite close. We note also that even when the solution was not a tree, the LP continuous relaxation gave an integer solution (identical to the one of the ILP). We also precise that we also checked compatibility with an axis, and no profile was single-peaked with respect to an axis.

**ED-00009 data set.** Each student provided a rank ordering over all the courses with no missing elements. There were 9 courses to choose from in 2003 and 7 in 2004, and about 150 students. For both years, the optimal solution was a tree, more specifically a star. This is easily explained from the fact that, in both years, there was one course which was the most preferred for every student.

**ED-00012 and ED-00014 data sets.** The optimal solution for the preferences over the T-shirt designs had 25 edges, which is quite a lot regarding the number of candidates (11) and voters (30). However, it is consistent with the intuition that there is probably no structure behind T-shirt designs. The same can be said for the preferences over the kinds of sushi, where 5000 voters were asked for their preferences about 10 kinds of sushi (the optimal solution is a complete graph in this case).

**ED-00032 data set** In the single instance with no tie nor missing element, there were 15 students that ranked the 6 criteria affecting their performances. In both the ILP and LP formulations, a solution with 8 edges has been returned.

File	#candidates	#voters	#edges
ED-00006-00000003.soc	14	9	13 (tree)
ED-00006-00000004.soc	14	9	13 (tree)
ED-00006-00000007.soc	23	9	22 (tree)
ED-00006-00000008.soc	23	9	22 (tree)
ED-00006-00000011.soc	20	9	20 (pseudotree)
ED-00006-00000012.soc	20	9	20 (pseudotree)
ED-00006-00000018.soc	24	9	23 (tree)
ED-00006-00000021.soc	18	7	17 (tree)
ED-00006-00000022.soc	18	7	17 (tree)
ED-00006-00000028.soc	24	9	23 (tree)
ED-00006-00000029.soc	19	9	23
ED-00006-00000032.soc	23	9	23 (pseudotree)
ED-00006-00000033.soc	23	9	22 (tree)
ED-00006-00000034.soc	23	9	22 (tree)
ED-00006-00000035.soc	18	9	17 (tree)
ED-00006-00000036.soc	18	9	17 (tree)
ED-00006-00000037.soc	19	9	18 (tree)
ED-00006-00000044.soc	20	9	19 (tree)
ED-00006-00000046.soc	30	9	30 (pseudotree)
ED-00006-00000048.soc	24	9	23 (tree)
ED-00009-00000001.soc	9	146	8 (tree)
ED-00009-00000002.soc	7	153	6 (tree)
ED-00012-00000001.soc	11	30	25
ED-00014-00000001.soc	10	5000	45 (clique)
ED-00032-00000002.soc	6	15	7

Figure 2: Minimal number of edges on real data sets from PrefLib. The first column gives the file name, the second and third the number of candidates and voters. The fourth column gives the minimal number of edges; specific structures (axis, tree, pseudotree or clique) are indicated in parentheses when they appear.

## 5.2 Experimental study on randomly generated data

The experimental study on real data revealed some interesting information. Nevertheless, it is limited by the small amount of data available. It seems indeed hard to obtain real election data with complete strict order rankings. To overcome this difficulty, other experiments can be considered. One of them is to adapt the approach to partial order rankings that can be met more often in practice. However, in this paper, we preferred to generate random data in order to study the structure of solutions and the relation between the solutions returned by ILP and LP formulations.

As mentioned in the previous section, it seems that in some contexts we can assume that the voter’s preferences are biased observations of a “true” ranking. This idea can be modeled using the Mallows distribution on rankings. In this model, the “true” ranking is called *central permutation* and its probability is the highest one. The probability of other permutations decreases with the Kendall tau distance from the central permutation. Formally, let  $R_0$  be the central permutation. The probability of a permutation  $R$  is:

$$P(R) = \frac{\exp(-\theta d(R, R_0))}{\psi(\theta)}$$



where  $d(.,.)$  is the Kendall tau distance,  $\theta$  is a dispersion parameter modeling the opinion heterogeneity, and  $\psi(\theta)$  is a normalisation constant. The parameter  $\theta$  is a real number greater than or equal to 0. If  $\theta = 0$ , the uniform distribution is obtained. The greater the value of  $\theta$ , the more the voters agree on the central permutation.

We used the PerMallows R package<sup>2</sup> for generating the random data according to the Mallows model. The number of candidates was set to  $m = 20$ , the value of  $\theta$  varied from 0 to 1 by step of 0.1. The number of voters  $n$  varied from 20 to 100 by step of 10. For each pair  $(\theta_0, n_0)$  of parameter values, the results are averaged over 1000 randomly drawn preference profiles. The curves in Figure 3 shows the evolution of the graph density according to these parameter values.

In the best case, the obtained solution is a tree, hence, the density is  $(m-1)/\frac{m(m-1)}{2} = \frac{2}{m}$ . As we set  $m = 20$ , this corresponds to a density of 0.1. The function representing the graph density seems indeed to converge to the constant function of value 0.1 while the value of  $\theta$  increases and the preferences in the profile become similar (the curves get closer and closer to the x-axis). Put another way, the density captures the similarity of voters' preferences, as clearly the higher  $\theta$  the lower the curve. On the contrary, the graph density becomes of course higher when the number  $n$  of voters increases. Nevertheless, note that, even for 100 voters, the graph is still quite far from being complete. Besides, the slope of the curve decreases with  $n$ . During our experiments, we plotted functions  $1 - \log(\text{density})$  and obtained a set of (approximative) straight lines, thus indicating that the convergence towards density 1 (complete graphs) is of the form  $1 - e^{-\lambda_\theta n}$ , where  $\lambda_\theta > 0$  is a parameter decreasing with  $\theta$ .

We now give some theoretical arguments that support this observation. Let us recall that if a voter ranks  $j$  first and  $k$  second (or the opposite), then edge  $\{j, k\}$  must be present in the graph and is called *necessary* edge. Assuming that the preferences in the profile are generated with the Mallows model, let us now estimate the number of necessary edges in the graph for  $n$  voters and  $m$  candidates, which gives us an underestimation and hopefully good approximation of the total number of edges. Let  $\theta$  be the model parameter and  $R_0$  the central permutation. The probability that a preference induces the necessary edge  $\{j, k\}$  is

$$P(\{j, k\}) = \frac{1}{\psi(\theta)} \sum_{R \in \mathcal{R}_{\{j, k\}}} \exp(-\theta d(R, R_0)) \quad (1)$$

where  $\mathcal{R}_{\{j, k\}}$  is the set of permutations of  $\{1, \dots, m\}$  that ranks  $j$  and  $k$  in the first two positions. In a profile with  $n$  voters, the probability that no preference induces the necessary edge  $\{j, k\}$  is then written

$$(1 - P(\{j, k\}))^n.$$

Hence, by passing to the complement, the probability that  $\{j, k\}$  is a necessary edge is

$$P_n(\{j, k\}) = 1 - (1 - P(\{j, k\}))^n.$$

Finally, we obtain the expected value of the number of edges as

$$\sum_{\{j, k\} \subseteq \{1, \dots, m\}} P_n(\{j, k\}) = \binom{m}{2} - \sum_{\{j, k\} \subseteq \{1, \dots, m\}} (1 - P(\{j, k\}))^n. \quad (2)$$

For  $\theta = 0$ , as the distribution is uniform, we get that  $P(\{j, k\}) = 1/\binom{m}{2}$ . Then, we directly obtain that the average number of necessary edges is  $\binom{m}{2} (1 - e^{-\alpha n})$  with  $\alpha = -\ln(1 - 1/\binom{m}{2})$ , thus contributing for  $(1 - e^{-\alpha n})$  in the density, in accordance with the experiments. The curves in Figure 4 shows the evolution of the expected contribution of necessary edges in the graph density according to the values of  $\theta$  and  $n$ . The result extends to any value of  $\theta$  but requires a dedicated

<sup>2</sup><https://cran.r-project.org/web/packages/PerMallows/index.html>

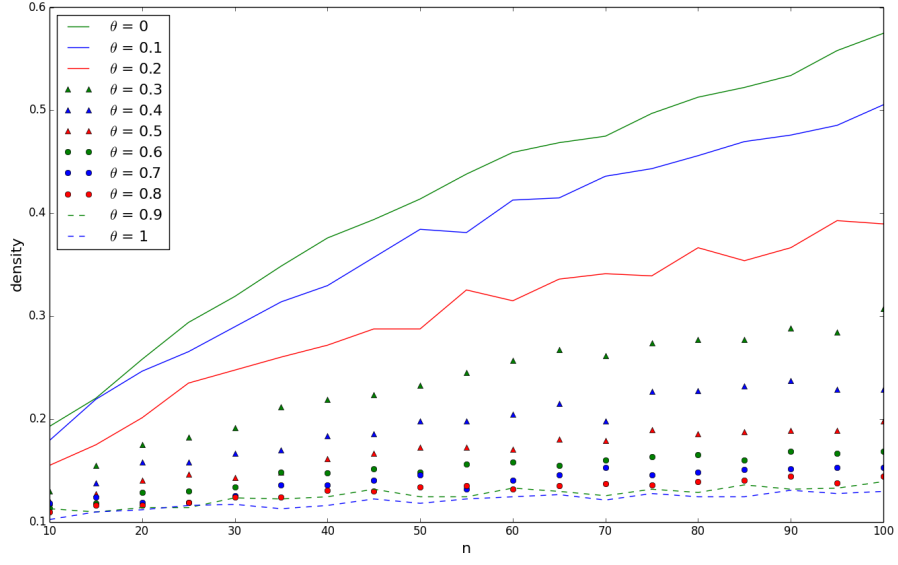


Figure 3: Density of the graph according to parameters  $\theta$  and  $n$  (with  $m=20$ ).

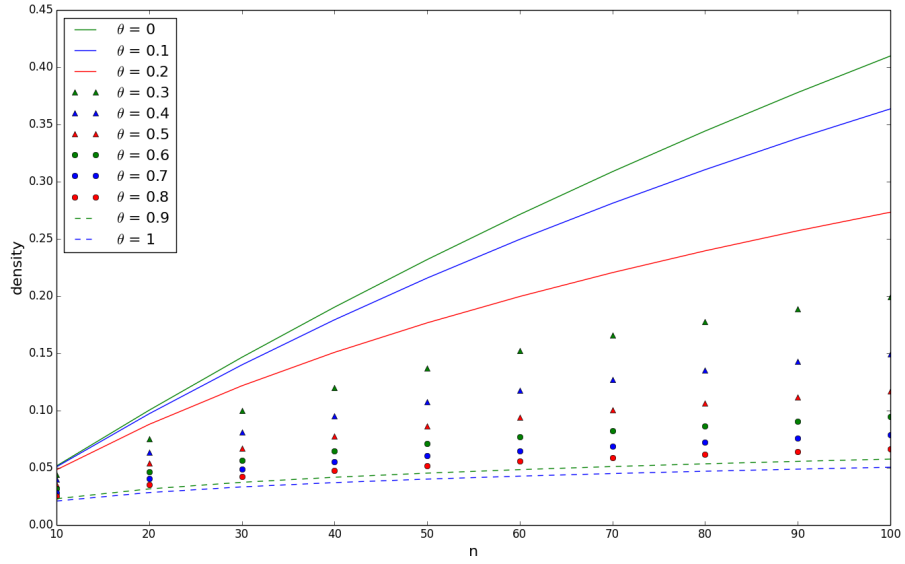


Figure 4: Expected number of necessary edges according to parameters  $\theta$  and  $n$  (with  $m=20$ ).

algorithm to compute efficiently  $P(\{j, k\})$  in Equation 2 (see Appendix A). As expected, we can see that the shapes of the curves coincide in Figures 3 and 4. Note, however, that the scale of the y-axis in Figure 4 slightly differs from the one in Figure 3 (the curves in Figure 4 indeed only account for *necessary* edges, thus the analytical values are smaller than the experimental ones).

By using the formula in Equation 2, we can have an idea of the evolution of the number of necessary edges in the graph for up to 1000 voters whose preferences follows the Mallows model. The obtained curves for various values of  $\theta$  are shown in Figure 5. For instance, if one assumes that all votes are equally likely (*impartial culture assumption*, corresponding to  $\theta = 0$ ), then the graph becomes complete for a thousand voters, while only around 45% of the edges are present if one sets  $\theta = 0.2$  (i.e., a lower preference heterogeneity).

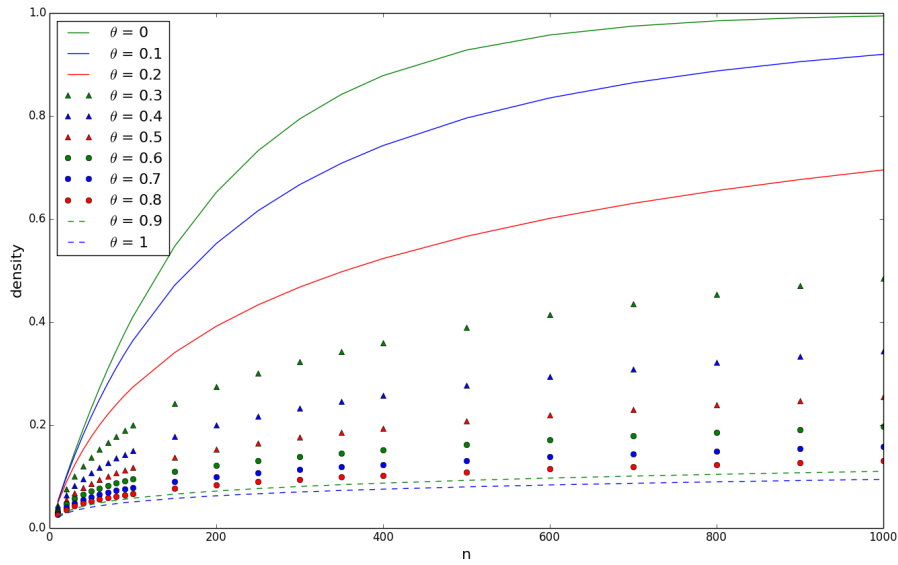


Figure 5: Expected number of necessary edges for up to  $n = 1000$  voters following the Mallows model of parameter  $\theta$  (with  $m = 20$ ).

## 6 CONCLUSION

In this paper, we have investigated single-peakedness on an arbitrary graph  $\mathcal{G}$ , a generalization of single-peakedness of an axis that offers the advantage that any preference profile is at worst single-peaked on a complete graph. The recognition problem is then turned into an optimization problem the objective function of which depends on the type of structure that is looked for. After establishing negative complexity results regarding the minimization of the number of edges or the maximum vertex degree in  $\mathcal{G}$ , we have derived positive complexity results from an ILP formulation of the problem, by proving that the extremal optimal solutions of the continuous relaxation are integral if the preferences are single-peaked on a tree or a path. Finally, we have identified a new class of graphs for which the complexity of the recognition problem is polynomial, that of pseudotrees.

While the generalization of single-peakedness to arbitrary graphs make it more plausible to learn some preference structures in applications, an interesting research direction for future works would be to formulate the recognition problem as the determination of a *maximum likelihood* graph (while possibly imposing that the graph is an axis or a tree) on which the preferences are single-peaked.

Put another way, one would relax the requirement of perfect compatibility of the graph with the observed preferences in order to facilitate structure learning on real preference data. As shown in the numerical experiments, for an high preference heterogeneity, the graph can indeed become very dense when the number of voters increases (as is often the case in real applications), thus making essential to consider a more flexible view of single-peakedness. Some interesting works in this direction have already been carried out (recognition of nearly structured preferences, e.g., [7, 10, 13, 15]), but a lot remains to be done to make the approach fully operational.

## References

- [1] Ravindra K Ahuja, Thomas L Magnanti, and James B Orlin. Network flows. 1988.
- [2] Reid Andersen, Christian Borgs, Jennifer Chayes, Uriel Feige, Abraham Flaxman, Adam Kalai, Vahab Mirrokni, and Moshe Tennenholtz. Trust-based recommendation systems: an axiomatic approach. In *Proceedings of the 17th international conference on World Wide Web*, pages 199–208. ACM, 2008.
- [3] John Bartholdi III and Michael A Trick. Stable matching with preferences derived from a psychological model. *Operations Research Letters*, 5(4):165–169, 1986.
- [4] Nadja Betzler, Arkadii Slinko, and Johannes Uhlmann. On the computation of fully proportional representation. *Journal of Artificial Intelligence Research*, 47:475–519, 2013.
- [5] Duncan Black. On the rationale of group decision-making. *The Journal of Political Economy*, 56(1), 1948.
- [6] Felix Brandt, Markus Brill, Edith Hemaspaandra, and Lane A Hemaspaandra. Bypassing combinatorial protections: Polynomial-time algorithms for single-peaked electorates. *Journal of Artificial Intelligence Research*, 53:439–496, 2015.
- [7] Robert Bredereck, Jiehua Chen, and Gerhard J Woeginger. Are there any nicely structured preference profiles nearby? *Mathematical Social Sciences*, 79:61–73, 2016.
- [8] Weiwei Cheng and Eyke Hüllermeier. A new instance-based label ranking approach using the mallows model. In Wen Yu, Haibo He, and Nian Zhang, editors, *Advances in Neural Networks – ISNN 2009*, pages 707–716. Springer, 2009.
- [9] Stéphan Cléménçon, Anna Korba, and Eric Sibony. Ranking median regression: Learning to order through local consensus. In *Algorithmic Learning Theory, ALT 2018, 7-9 April 2018, Lanzarote, Canary Islands, Spain*, pages 212–245, 2018.
- [10] Denis Cornaz, Lucie Galand, and Olivier Spanjaard. Bounded single-peaked width and proportional representation. In *ECAI*, volume 12, pages 270–275, 2012.
- [11] Gabrielle Demange. Single-peaked orders on a tree. *Mathematical Social Sciences*, 3(4):389–396, 1982.
- [12] Edith Elkind, Martin Lackner, and Dominik Peters. Structured preferences. In *Trends in Computational social choice*, chapter 10, pages 187–207. AI Access, 2017.
- [13] Gábor Erdélyi, Martin Lackner, and Andreas Pfandler. Computational aspects of nearly single-peaked electorates. *Journal of Artificial Intelligence Research*, 58:297–337, 2017.
- [14] Bruno Escoffier, Jérôme Lang, and Meltem Öztürk. Single-peaked consistency and its complexity. In *ECAI*, volume 8, pages 366–370, 2008.

- [15] Piotr Faliszewski, Edith Hemaspaandra, and Lane A Hemaspaandra. The complexity of manipulative attacks in nearly single-peaked electorates. *Artificial Intelligence*, 207:69–99, 2014.
- [16] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.
- [17] John G Kemeny. Mathematics without numbers. *Daedalus*, 88(4):577–591, 1959.
- [18] Nicholas Mattei and Toby Walsh. Preflib: A library of preference data [HTTP://PREFLIB.ORG](http://preflib.org). In *Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT 2013)*, Lecture Notes in Artificial Intelligence. Springer, 2013.
- [19] Klaus Nehring and Clemens Puppe. The structure of strategy-proof social choice—part i: General characterization and possibility results on median spaces. *Journal of Economic Theory*, 135(1):269–305, 2007.
- [20] David M Pennock, Eric Horvitz, C Lee Giles, et al. Social choice theory and recommender systems: Analysis of the axiomatic foundations of collaborative filtering. In *AAAI/IAAI*, pages 729–734, 2000.
- [21] Dominik Peters. Single-peakedness and total unimodularity: New polynomial-time algorithms for multi-winner elections. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, February 2-7, 2018, New Orleans, Louisiana, USA*, pages 1169–1176, 2018.
- [22] Dominik Peters and Edith Elkind. Preferences single-peaked on nice trees. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [23] Dominik Peters and Martin Lackner. Preferences single-peaked on a circle. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 649–655, 2017.
- [24] Michael A Trick. Recognizing single-peaked preferences on a tree. *Mathematical Social Sciences*, 17(3):329–334, 1989.

Bruno Escoffier  
 Sorbonne Université, CNRS, LIP6, F-75005 and Institut Universitaire de France  
 Paris, France  
 Email: [bruno.escoffier@lip6.fr](mailto:bruno.escoffier@lip6.fr)

Olivier Spanjaard  
 Sorbonne Université, CNRS, LIP6, F-75005  
 Paris, France  
 Email: [olivier.spanjaard@lip6.fr](mailto:olivier.spanjaard@lip6.fr)

Magdaléna Tydrichová  
 Sorbonne Université, CNRS, LIP6, F-75005  
 Paris, France  
 Email: [magdalena.tydrichova@lip6.fr](mailto:magdalena.tydrichova@lip6.fr)

## A Computation of the expected number of necessary edges

In practice, it is computationally cumbersome to enumerate all permutations to compute  $P(\{j, k\})$  according to Equation 1 (page 17). We present here another approach to compute efficiently this value. Note that the maximal distance between two permutations of length  $m$  is  $m(m-1)/2$ . For each value  $\delta \in \{1, \dots, m(m-1)/2\}$ , let  $N^m(\delta)$  denote the number of permutations  $R$  such

that  $d(R, R_0) = \delta$ , and  $N_{\{j,k\}}^m(\delta)$  the number of permutations  $R = (\pi(1), \dots, \pi(m))$  such that  $\{\pi(1), \pi(2)\} = \{j, k\}$  and  $d(R, R_0) = \delta$ . Then,  $P(\{j, k\})$  can be computed as

$$P(\{j, k\}) = \frac{\sum_{\delta=0}^{m(m-1)/2} N_{\{j,k\}}^m(\delta) \exp(-\theta\delta)}{\psi(\theta)}. \quad (3)$$

The value  $N_{\{j,k\}}^m(\delta)$  can be computed as follows:

- Firstly, we define the permutation  $R_1 = (\pi_1(1), \dots, \pi_1(m))$  such that  $\pi_1(1) = j, \pi_1(2) = k$  and for each pair  $i, l$  of candidates different from  $j$  and  $k$ , we have

$$\pi_1^{-1}(i) < \pi_1^{-1}(l) \text{ if and only if } \pi_0^{-1}(i) < \pi_0^{-1}(l).$$

Similarly, we define the permutation  $R_2 = (\pi_2(1), \dots, \pi_2(m))$  such that  $\pi_2(1) = k, \pi_2(2) = j$  and for each pair  $i, l$  of candidates different from  $j$  and  $k$ , we have

$$\pi_2^{-1}(i) < \pi_2^{-1}(l) \text{ if and only if } \pi_0^{-1}(i) < \pi_0^{-1}(l).$$

We denote by  $\delta_1$  (resp.  $\delta_2$ ) the Kendall tau distance between  $R_0$  and  $R_1$  (resp.  $R_2$ ).

- As  $R_1$  (resp.  $R_2$ ) is between<sup>3</sup>  $R_0$  and any permutation  $R$  such that  $\pi(1) = j$  and  $\pi(2) = k$  (resp.  $\pi(1) = k$  and  $\pi(2) = j$ ), we have  $d(R_0, R) = d(R_0, R_1) + d(R_1, R)$  (resp.  $d(R_0, R) = d(R_0, R_2) + d(R_2, R)$ ) because the Kendall tau distance satisfies the betweenness condition<sup>4</sup> [17]. Consequently, the number of permutations  $R$  inducing the necessary edge  $\{j, k\}$  and such that  $d(R_0, R) = \delta$  can be computed as

$$N_{\{j,k\}}^m(\delta) = N^{m-2}(\delta - \delta_1) + N^{m-2}(\delta - \delta_2)$$

because  $d(R_0, R_1) = \delta_1$  and  $d(R_0, R_2) = \delta_2$  by definition of  $\delta_1$  and  $\delta_2$ . Note that this equation is well defined because  $N^m(\delta)$  is fully characterized by  $m$  and  $\delta$ . The problem consists now in determining the values  $N^{m-2}(\delta - \delta_1)$  and  $N^{m-2}(\delta - \delta_2)$ . In this purpose, for any value  $m$  and distance  $\delta$ ,  $N^m(\delta)$  can be computed thanks to the following recursion principle:

- Regardless of the length  $m$  of the permutations considered,  $N^m(0) = 1$  as there is only one permutation at distance 0 of  $R_0$  - it is  $R_0$  itself.
- Let  $m > 1$  and  $\delta > 0$ . Let  $R$  be an arbitrary permutation of length  $m$  at distance  $\delta$  from  $R_0$ . The distance between  $R_0$  and  $R$  can be calculated as the number  $\delta'$  of swap operations needed to move  $\pi_0(1)$  to the first position in  $R$ , to which one adds the distance between the restrictions of  $R_0$  and  $R$  to their  $(m-1)$  last elements. We have  $0 \leq \delta' \leq m-1$ , and so  $N^m(\delta) = \sum_{\delta'=0}^{\min(\delta, m-1)} N^{m-1}(\delta - \delta')$ .

Overall, after a single preprocessing step in  $O(m^4)$ , each probability  $P(\{j, k\})$  can be computed in  $O(m^2)$ . The preprocessing step consists in determining  $N^i(\delta_i)$  for each  $i \in \{1, \dots, m\}$  and  $\delta_i \in \{1, \dots, i(i-1)/2\}$ . Hence, there are  $N(m) \in O(m^3)$  values to compute, and each of them is obtained in  $O(m)$ . Once these values are computed, Equation 3 allows us to compute  $P(\{j, k\})$  in  $O(m^2)$ . This is a significant improvement compared to the brute force implementation in  $O(m!)$  of Equation 1.

<sup>3</sup>A ranking  $R' = (\pi'(1), \dots, \pi'(m))$  is *between* rankings  $R = (\pi(1), \dots, \pi(m))$  and  $R'' = (\pi''(1), \dots, \pi''(m))$  if, for any pair  $i, j$  of candidates,  $\pi^{-1}(i) < \pi^{-1}(j)$  and  $\pi'^{-1}(i) < \pi'^{-1}(j)$  implies that  $\pi''^{-1}(i) < \pi''^{-1}(j)$ .

<sup>4</sup>A distance  $d$  satisfies the betweenness condition if for all  $R, R', R''$  such that  $R'$  is between  $R$  and  $R''$  we have  $d(R, R'') = d(R, R') + d(R', R'')$ .