A Note on Maximizing the Minimum Voter Satisfaction on Spanning Trees

Andreas Darmann\textsuperscript{a}, Christian Klamler\textsuperscript{b}, Ulrich Pferschy\textsuperscript{c}

\textsuperscript{a}University of Graz, Institute of Public Economics, Universitaetsstr. 15, 8010 Graz, Austria
\textsuperscript{b}University of Graz, Institute of Public Economics, Universitaetsstr. 15, 8010 Graz, Austria; University of Hamburg, Department of Economics, Van-Melle-Park 5, 20146 Hamburg, Germany
\textsuperscript{c}University of Graz, Department of Statistics and Operations Research, Universitaetsstr. 15, 8010 Graz, Austria

Abstract

A fair spanning tree of a graph maximizes the minimum satisfaction among individuals given their preferences over the edges of the graph. In this note we answer an open question about the computational complexity of determining fair spanning trees raised in Darmann et al. \cite{8}. It is shown that the maximin voter satisfaction problem under choose-\(t\)-election is \(\mathcal{NP}\)-complete for each fixed \(t \geq 2\).

1. Introduction

The topic of finding a minimum spanning tree is a classical problem in Combinatorial Optimization. Having determined a minimum spanning tree, the natural question may arise how to “fairly” assign the costs of such a tree to individuals. This issue has been widely studied since Claus and Keitman \cite{6}, e.g., by Bird \cite{2}, Bogomolnaia and Moulin \cite{3} and Dutta and Kar \cite{9}.

In this paper however, we focus on another aspect of fairness in connection with spanning trees. In our framework monetary costs are not taken into consideration. Instead, our approach is based on individuals’ preferences over the edges of a graph. Given such preferences, in Darmann, et al. \cite{7} the quality of methods that fairly, i.e., socially acceptably, construct a spanning tree is analyzed. In a companion paper of Darmann et al. \cite{8}, the focus is laid on the computational complexity of methods to fairly (in a maximin-sense) construct a spanning tree. In this paper we study the computational complexity of another method to fairly construct a spanning tree and therewith answer an open question stated in Darmann et al. \cite{8}.

Our framework covers many applications in which a network needs to be installed, e.g., when countries need to agree on transnational traffic systems or oil pipelines, or when homeowners have diverting opinions about the specific links in a sewage or water network that needs to be constructed.

In the Combinatorial Optimization literature maximin fairness can be found in terms of maximizing the minimum of concepts such as utility, costs, time, etc. However, in Social Choice Theory the maximin-approach is a well-known concept of formalizing fairness, originally discussed by Rawls \cite{12}; for extensive studies of social choice rules see e.g. Brams and Fishburn \cite{5}, Nurmi \cite{11} and Saari \cite{13}. In this paper, we follow the maximin-approach based on different social choice rules proposed in Darmann et al. \cite{8} and show that the maximin voter satisfaction problem (MMVS) under choose-\(t\) election is \(\mathcal{NP}\)-hard for each fixed \(t \geq 2\).

Email addresses: andreas.darmann@uni-graz.at (Andreas Darmann), christian.klamler@uni-graz.at (Christian Klamler), pferschy@uni-graz.at (Ulrich Pferschy)

Preprint submitted to Elsevier January 29, 2010
2. Preliminaries

2.1. Problem formulation

In this work \( G = (V, E) \) denotes an undirected graph, with a set \( V \) of vertices and a set \( E \) of edges. Let \( n \) be the number of vertices and and \( m \) be the number of edges of \( G \). We call a subset \( T \subseteq E \) spanning tree of \( G \) if the graph \((V, T)\) is connected and acyclic. Let \( \tau \) be the set of spanning trees of \( G \). A binary relation \( \succeq \) on a set \( A \) is called

- complete if \( \forall a, b \in A, a \neq b, a \succeq b \) or \( b \succeq a \).
- reflexive if \( \forall a \in A, a \succeq a \).
- transitive if \( \forall a, b, c \in A, (a \succeq b \) and \( b \succeq c) \Rightarrow a \succeq c \).

A relation is called weak order if it is complete, reflexive and transitive.

In our framework we are given a finite set \( I = \{1, 2, \ldots, k\} \) of voters. We assume that for every voter \( i \), \( 1 \leq i \leq k \), the voter’s preferences on the edges are given in terms of a weak order \( \succeq_i \) on \( E \). The \( k \)-tuple \( \pi = (\succeq_1, \succeq_2, \ldots, \succeq_k) \) is called voter preference profile.

The basic concept used in Darmann et al. \[8\] is the one of voters’ scoring functions, which can be understood as a generalization of the positional scoring procedures (for details concerning these procedures see Brams and Fishburn \[5\]).

**Definition 1** Let \( 1 \leq i \leq k \). We call a function \( v_i : E \to \mathbb{N}_0 \) voter \( i \)'s scoring function, if the following properties are satisfied:

1. for all \( e, f \in E \) \( e \succeq_i f \Leftrightarrow v_i(e) \geq v_i(f) \)
2. \( \max_{e \in E} \{v_i(e)\} \) is bounded by a polynomial in \( n \).

As in \[8\] we assume voters’ preferences on trees to be additively separable, i.e., there do not exist complementarities or synergies between the edges.

**Definition 2** For \( 1 \leq i \leq k \) let \( v_i \) be voter \( i \)'s scoring function. Voter \( i \)'s score of tree \( T \in \tau \) is \( v_i(T) := \sum_{e \in T} v_i(e) \).

Many classical voting procedures can be embedded in the framework of voters’ scoring functions (for details see Darmann et al. \[8\]). In this paper however, the focus is laid on choose-\( t \) election, which has a certain similarity to approval voting (see Brams and Fishburn \[4, 5\]). Given \( t \in \mathbb{N} \), in choose-\( t \) election for every voter \( i \) the set \( E \) is partitioned into a set \( S_i \subseteq E \), \( |S_i| = t \), of edges voter \( i \) approves of and a set \( E \setminus S_i \) of edges voter \( i \) disapproves of.

**Definition 3** Let \( 1 \leq i \leq k \) and \( t \in \mathbb{N} \). Let \( S_i \subseteq E \) with \( |S_i| = t \). In choose-\( t \) election voter \( i \)'s scoring function is the function \( a_i : E \to \mathbb{N}_0 \) with

\[
a_i(e) = \begin{cases} 
1 & \text{if } e \in S_i \\
0 & \text{if } e \in E \setminus S_i.
\end{cases}
\]

The function \( a_i \) is called voter \( i \)'s choose-\( t \) function. Voter \( i \)'s choose-\( t \) count of \( T \in \tau \) is defined by \( a_i(T) := \sum_{e \in T} a_i(e) \), and corresponds to the number of edges of \( T \) voter \( i \) approves of.

\(^1\)I.e., choose-\( t \) election corresponds to approval voting with a fixed number \( t \) of approved edges.
Now, we state the maximin voter satisfaction problem (MMVS). For the sake of formal correctness (since we deal with \(NP\)-completeness), we state the decision problem corresponding to MMVS as well.

**Definition 4 MMVS (optimization problem version)**

Let \( G = (V, E) \) be an undirected graph, let \( I \) be a set of voters and let \( \pi \) be a voter preference profile. For \( i \in I \) let \( v_i \) be voter \( i \)'s scoring function. The maximin voter satisfaction problem (MMVS) is the following problem:

\[
\max_{T \in \tau} \min_{i \in I} v_i(T)
\]

**Definition 5 MMVS (decision problem version)**

**GIVEN:** Undirected graph \( G = (V, E) \), set \( I \) of voters, voter preference profile \( \pi \), voter \( i \)'s scoring function \( v_i \) for all \( i \in I \), and non-negative integer \( K \).

**QUESTION:** Is there a spanning tree \( T \) of \( G \) such that \( v_i(T) \geq K \) for all \( i \in I \)?

In the optimization literature the problem appears in the completely different context of robust optimization. In this framework, Aissi, Bazgan and Vanderpooten [1] refer to an analogon of MMVS as *max-min spanning tree problem* while Kouvelis and Yu [10] use the terminology *absolute robust minimum spanning tree problem*. This paper's focus however is laid on the complexity of aggregating voters’ opinions with the help of a certain type of voting procedure.

### 2.2. Known results and our contribution

Kouvelis and Yu [10] showed that MMVS is strongly \(NP\)-complete for arbitrary scoring functions. However, the question of the computational complexity of MMVS under common voting procedures was not answered by Kouvelis and Yu [10]. It was shown by Darmann et al. [8] that MMVS is \(NP\)-complete under approval voting, Borda voting and vote-against-\(t\) election for \( t \geq 2 \), while it is polynomially solvable under vote-against-1 election and plurality voting.\(^2\) The computational complexity of MMVS under choose-\(t\) election, \( t \geq 2 \), has been open [8].

The contribution of this paper is to show that MMVS is \(NP\)-complete under choose-\(t\) election, \( t \geq 2 \), thus completing the analysis of the computational complexity of MMVS under the most common voting rules. In fact, our result settles the complexity status for any reasonable election process: As long as every voter is allowed to distinguish only one edge in a positive or negative sense the problem remains polynomially solvable. As soon as two or more edges receive an appraisal different from the remaining edges, the problem becomes strongly \(NP\)-complete.

### 3. MMVS under choose \(t\)-election

**Theorem 1** MMVS under choose-2 election is strongly \(NP\)-complete.

**Proof.** Let \( \Omega \) be an arbitrary instance of 3-SAT with a set \( X := \{x_1, ..., x_\ell\} \) of variables and a set \( C := \{C_1, ..., C_q\} \) of clauses over \( X \). From \( \Omega \) we create an instance \( J \) of MMVS by building a graph \( G \) (see Figure 1) and a voter preference profile \( \pi \) (see Table 1) as follows.

In order to create \( G = (V, E) \), for every variable \( x_j \) we introduce a triangle \((x_j, \bar{x}_j, g_j)\) and an edge connecting the vertex adjacent to \( x_j \) and \( \bar{x}_j \) to a dedicated vertex \( r \). Next, for each clause \( C_u \) we introduce a triangle \((d_{u,1}, d_{u,2}, d_{u,3})\) and, for \( 1 \leq u \leq q \), add an edge connecting \( r \) to the vertex

\(^2\)Note that plurality voting corresponds to choose-1 election.
adjacent to both $d_{u,1}$ and $d_{u,2}$. Finally, a triangle $(z_1, z_2, z_3)$ is added and the vertex adjacent to $z_1$ and $z_2$ is connected to $r$ by an edge.

For creating $\pi$, we first introduce $3\ell$ voters $\gamma^h_j$, $1 \leq h \leq 3$, $1 \leq j \leq \ell$, such that voter $\gamma^h_j$ assigns value 1 to the edges $g_j$ and $z_h$ and zero otherwise. Now let clause $C_u$ consist of e.g. the literals $x_{u,1}, \bar{x}_{u,2}$ and $x_{u,3}$. Then we add a voter $C^1_u$ which assigns value 1 to edges $d_{u,1}$ and $x_{u,1}$ and zero to all other edges. Further, we add voter $C^2_u$ assigning value 1 to the edges $\bar{x}_{u,2}$ and $d_{u,2}$ and zero to all other edges, and voter $C^3_u$ assigning value 1 to the edges $x_{u,3}$ and $d_{u,3}$ and zero to all other edges. Note that the size of the constructed instance $J$ is polynomial, since $|V| = 3(\ell + q) + 4$, $|E| = 4(\ell + q) + 1$ and the number of voters is $k = 3(\ell + q)$.

<table>
<thead>
<tr>
<th>$\gamma^1_1$</th>
<th>$\gamma^1_2$</th>
<th>$\gamma^1_{\ell}$</th>
<th>$\cdots$</th>
<th>$\gamma^1_\ell$</th>
<th>$C^1_u$</th>
<th>$C^2_u$</th>
<th>$C^3_u$</th>
</tr>
</thead>
<tbody>
<tr>
<td>edge $a_{x_1}$</td>
<td>edge $a_{x_2}$</td>
<td>edge $a_{x_3}$</td>
<td>$g_1$</td>
<td>$g_2$</td>
<td>$\cdots$</td>
<td>$g_\ell$</td>
<td>edge $a_{\bar{x}_{u,1}}$</td>
</tr>
<tr>
<td>$z_1$</td>
<td>$z_2$</td>
<td>$z_3$</td>
<td>$x_{u,1}$</td>
<td>$\bar{x}_{u,2}$</td>
<td>$x_{u,3}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Preference profile $\pi$ for the voters $\gamma^h_j$, $1 \leq j \leq \ell$ and $1 \leq h \leq 3$, and the voters $C^h_u$, $1 \leq h \leq 3$, derived from clause $C_u$ which is made up of the literals $\{x_{u,1}, \bar{x}_{u,2}, x_{u,3}\}$.

Figure 1: Graph $G$ for instance $J$ of MMVS.
We prove the theorem by showing the following equivalence:

\[ \Omega \text{ is satisfiable } \iff \exists J \text{ instance } J \text{ contains a spanning tree } T \text{ of } G \text{ such that } a_i(T) \geq 1 \text{ for all voters } i, 1 \leq i \leq k. \quad (1) \]

\[ \Rightarrow: \] If there is a satisfying truth assignment \( s \) for \( \Omega \), then we construct the spanning tree \( T \) as follows. At first we add to \( T \) all the edges emanating from \( r \). Next, we add all edges \( g_j, 1 \leq j \leq \ell \), and two arbitrary edges of the triangle \((z_1, z_2, z_3)\) to \( T \). For \( 1 \leq j \leq \ell \), if \( x_j \) is set “TRUE” under \( s \), then we add \( x_j \) to \( T \), otherwise we add \( \tilde{x}_j \) to \( T \). Now for each triangle \((d_{u,1}, d_{u,2}, d_{u,3})\), we add to \( T \) two of its edges such that the missing third edge \( d_{u,j'} \) corresponds to a literal \( x_{u,j'} \) (or \( \tilde{x}_{u,j'} \) respectively) which is already in \( T \). It is straightforward to see that \( T \) is a spanning tree of \( G \).

Concerning the scoring function of each voter, since all edges \( g_j \) are in \( T \) each voter \( \gamma^h \) has a score at least one for \( T \). Since \( s \) is a satisfying truth assignment, for each clause at least one of its literals is set “TRUE”. Thus, for every \( u \) there is at least one voter \( C^u \), \( 1 \leq j' \leq 3 \), such that \( x_{u,j'} \) (or \( \tilde{x}_{u,j'} \) respectively) is in \( T \). W.l.o.g. we assume that \( d_{u,j'} \) is not in \( T \).\(^3\) However, by construction for each of the two remaining voters the edge \( d_{u,j''}, j'' \neq j' \), is in \( T \). Therefore, \( a_i(T) \geq 1 \) holds for all voters \( i, 1 \leq i \leq k \).

\[ \Leftarrow: \] Let \( T \) be a spanning tree of \( G \) satisfying \( a_i(T) \geq 1 \) for all voters \( i, 1 \leq i \leq k \). Considering for every \( u, 1 \leq u \leq q \), the triangle \((d_{u,1}, d_{u,2}, d_{u,3})\) it follows from the spanning tree property that exactly two of its edges must be contained in \( T \). However, since all three voters \( C^u \), \( 1 \leq j' \leq 3 \), score at least one, there must be at least one edge \( x_{u,j'} \) (or \( \tilde{x}_{u,j'} \) resp.) contained in \( T \), i.e., at least one of the literals that make up \( C_u \) is in \( T \).

Further, from the voters \( \gamma^h \) it follows that the edges \( g_1, ..., g_t \) must be contained in \( T \), since only two edges of the triangle \((z_1, z_2, z_3)\) can be contained in \( T \). This however implies that for all \( j \) exactly one of \( x_j \) or \( \tilde{x}_j \) is in \( T \), since \( T \) is acyclic. Putting things together, we get that the assignment \( s \) defined by setting \( x_j \) “TRUE” if the edge \( x_j \) is in \( T \) and “FALSE” otherwise, is a satisfying truth assignment for \( T \). \( \square \)

**Theorem 2** MMVS under choose-\( t \) election is strongly \( \mathcal{NP} \)-complete for every fixed \( t \geq 2 \).

**Proof.** The case \( t = 2 \) is covered by Theorem 1. Let \( t \geq 3 \). Analogously to the proof of Theorem 1 we transform an arbitrary instance \( \Omega \) of 3-SAT with a set \( X := \{x_1, ..., x_t\} \) of variables and a set \( C := \{C_1, ..., C_q\} \) of clauses over \( X \) to an instance \( J_1 \) of MMVS under choose-\( t \) election. \( J_1 \) is defined by a graph \( G_1 \) and a voter preference profile \( \pi_1 \), where \( G_1 \) is created from the graph \( G \) used in the proof of Theorem 1 by concatenating a path of length \( t - 2 \) to vertex \( r \), i.e., we insert \( t - 2 \) vertices and edges \( e_1, e_2, ..., e_{t-2} \). Now we derive the preference profile \( \pi_1 \) from the profile \( \pi \) used in the proof of Theorem 1 by assigning value one to each of these new edges for every voter. Thus, in \( \pi_1 \) each voter has assigned value one to exactly \( t \) edges. Now since any spanning tree of \( G_1 \) necessarily contains all of the edges \( e_1, e_2, ..., e_{t-2} \), the following two decision problems (P1) and (P2) are equivalent:

\[(P1) \text{ GIVEN: Graph } G \text{ and preference profile } \pi. \]
\[ \text{QUESTION: Is there a spanning tree } T \text{ of } G \text{ such that } a_i(T) \geq 1 \text{ for all voters } i, 1 \leq i \leq k? \]

\[(P2) \text{ GIVEN: Graph } G \text{ and preference profile } \pi_1. \]
\[ \text{QUESTION: Is there a spanning tree } T_1 \text{ of } G_1 \text{ such that } a_i(T) \geq t - 1 \text{ for all voters } i, 1 \leq i \leq k? \]

Now the statement follows from (1) shown in the proof of Theorem 1. \( \square \)

\(^3\)By construction for each \( u \) there is always a \( j' \) such that, given \( x_{u,j'} \) (resp. \( \tilde{x}_{u,j'} \)) is in \( T \), \( d_{u,j'} \) is not in \( T \).
References


