Computing Spanning Trees in a Social Choice Context

Andreas Darmann, Christian Klamler and Ulrich Pferschy

Abstract

This paper combines social choice theory with discrete optimization. We assume that individuals have preferences over edges of a graph that need to be aggregated. The goal is to find a socially "best" spanning tree in the graph. As ranking all spanning trees is becoming infeasible even for small numbers of vertices and/or edges of a graph, our interest lies in finding algorithms that determine a socially "best" spanning tree in a simple manner. This problem is closely related to the minimum (or maximum) spanning tree problem in combinatorial optimization. Our main result shows that for the various underlying ranking rules on the set of spanning trees discussed in this paper the sets of "best" spanning trees coincide. Moreover, a greedy algorithm based on a transitive group ranking on the set of edges will always provide such a "best" spanning tree.

1 Introduction

In this paper we want to apply tools from social choice theory to topics from discrete optimization. Although these topics are historically separated, in recent years there have started attempts to combine these approaches. This is especially of interest whenever mathematical concepts (such as graphs) are used in problems where group decisions need to be made.

As an actual example one could think of a small village that has to install a water network or countries that need to agree on oil pipelines. Every homeowner in the village needs to be connected, however, there are many different ways to hook them up. A mathematical representation of such a situation could be done by a spanning tree on a graph that connects each pair of homeowners (i.e. vertices) in the village. However, different homeowners might have different preferences over the possible connections between the homeowners (i.e. edges of the graph). E.g. one homeowner might rather want to have it pass through his own garden than through a nice park, whereas another one might think the other way round.

Such aggregations of individual preferences are the major focus in social choice theory and many different aggregation rules do exist and have been studied and compared in the literature (see e.g. [8] or [10]). Our approach, however, will not analyse the aggregation of such individual preferences, but start off with a group ranking of the possible edges. This ranking of the edges does not necessarily allocate numerical values to the edges. As each spanning tree is a subset of the set of edges having a certain structure, we will - given the group ranking - try to rank the different spanning trees. This lies in the spirit of previous results on ranking sets of objects (for an overview see [2]). Such rankings could be based on Borda counts, on simple majority rule, be of lexicographic nature, etc; as for the edges, the ranking of the spanning trees does not have to be based on numerical values of the edges and does not need to assign a number to each tree. The goal for this type of problems is to find the spanning tree which is "best" w.r.t. such a relation on spanning trees. Ranking all spanning trees is, however, a difficult problem even for small number of vertices and/or edges given the quickly increasing number of feasible spanning trees. Our interest therefore lies in finding algorithms that determine a "best" spanning tree in a simple manner. This problem is closely related to the minimum (or maximum) spanning tree problem, a classical problem in discrete optimization. The minimum spanning tree problem has numerous applications in various fields and can be solved efficiently by greedy algorithms [1].

The main result in this paper shows that irrespective of the underlying ranking rules discussed in this paper the set of “best” spanning trees coincide. What’s more, using a greedy algorithm to determine a maximal spanning tree based on a transitive group ranking on the set of edges will always provide a "best" spanning tree.

2 Formal framework

2.1 Preliminaries

Let $G = (V, E)$ be an undirected graph where V denotes the set of nodes and E denotes the set of edges. Let $n := |V|$ and $m := |E|$. A subset $T \subseteq E$ is called spanning tree of G, if the subgraph (V, T) of G is acyclic and connected. Let τ denote the set of spanning trees of G. Let $I = \{1, ..., k\}$ denote a finite set of individuals. For every individual i, $1 \leq i \leq k$, the preference order P_i on E is assumed to be a linear order on E (i.e. P_i is assumed to be complete, transitive and asymmetric). A k-tuple $\pi = (P_1, P_2, ..., P_k)$ is called a preference profile and $\wp$ denotes the set of admissible preference profiles. A complete order $\succsim$ consists of an asymmetric part $\succ$ and a symmetric part $\sim$ respectively. Given a complete order $\succsim_\kappa$ on τ, we call a tree T *best tree with respect to* $\succsim_\kappa$ if there is no $B \in \tau$ with $B \succ_\kappa T$.

Remark. Let $T_1, T_2 \in \tau$. Having assigned a real number $w(e)$ to each $e \in E$, the minimum spanning tree problem is the problem of finding a best tree with respect to the relation $T_1 \succsim T_2 :\Longleftrightarrow \sum_{e \in T_1} w(e) \leq \sum_{e \in T_2} w(e)$. Analogously, a maximum spanning tree is a best tree with respect to the relation $T_1 \succsim T_2 :\Longleftrightarrow \sum_{e \in T_1} w(e) \geq \sum_{e \in T_2} w(e)$.

We first present basic complete orders on the set E of edges from which orders on the set τ of spanning trees of G are derived.

2.2 Basic orders on E

Definition 2.1 *Given P_i, individual i's Borda count of an edge $e \in E$ is given by $B_i(e) := |\{f \in E : eP_i f\}|$. The total Borda count of edge e is defined by $B(e) := \sum_{i \in I} B_i(e)$. For $e, f \in E$ we define $e \succsim_b f :\Longleftrightarrow B(e) \geq B(f)$.*

Definition 2.2 *Let $e, f \in E$. Then we define the Simple Majority-order on E by $e \succsim_{sm} f :\Longleftrightarrow |\{i \in I : eP_i f\}| \geq |\{i \in I : fP_i e\}|$.*

For all $i \in I$ we define the singleton set S_i^t representing individual i’s top choice by $S_i^t := \{e \in E | eP_i f \, \forall f \in E \setminus \{e\}\}$. Furthermore let, for all $i \in I$, the set E be partitioned into a set $S_i \subset E$ of edges individual i approves of and a set $E \setminus S_i$ individual i disapproves of.

Definition 2.3 *Let $e, f \in E$. The Approval count of e is defined by $A(e) := |\{i \in I : e \in S_i\}|$. The Approval-order $\succsim_a$ is then defined by $e \succsim_a f :\Longleftrightarrow A(e) \geq A(f)$.*

Definition 2.4 *Let $e, f \in E$. The Plurality count of e is $Pl(e) := |\{i \in I : e \in S_i^t\}|$. The Plurality-order $\succsim_{pl}$ is defined by $e \succsim_{pl} f :\Longleftrightarrow Pl(e) \geq Pl(f)$.*

The relations $\succsim_b$, $\succsim_a$ and $\succsim_{pl}$ are *weak orders* on E, i.e. these relations are complete and transitive. The relation $\succsim_{sm}$ is a complete order as well, but in general $\succsim_{sm}$ is not transitive and hence not a weak order. Thus, in $\succsim_{sm}$ preference cycles may occur. To overcome this inconvenience, one might be interested in procedures that transform a complete but not transitive order into a weak order. We introduce Copeland's procedure [7], other possibilities are e.g. Slater's procedure [11] or Black's procedure [5].

Definition 2.5 *Let $\succsim_n$ be a complete order on E and let $e, f \in E$. Let*

$$s(e,f) := \begin{cases} 1 & \text{if } e \succ_n f \\ 0 & \text{if } e \sim_n f \\ -1 & \text{if } e \prec_n f \end{cases}$$

be the score of e versus f. Let $z(e) := \sum_{g \in E} s(e,g)$. Then we define $e \succsim_{cl} f :\Longleftrightarrow z(e) \geq z(f)$ and call $\succsim_{cl}$ Copeland's order on E.

Remark. Note that $s(e,f) = -s(f,e)$ holds for all $e, f \in E$.

Obviously Copeland's order is a weak order on E. Thus setting $\succsim_n := \succsim_{sm}$ and determining the corresponding Copeland's order for example yields a weak order on E based on the Simple Majority-order.

3 Some complete orders on τ

We first present three weak orders on τ that are based on weak orders on E presented in Section 2.2. The Borda-order introduced in the following definition ranks $T_1 \in \tau$ not lower than $T_2 \in \tau$, if the sum of Borda counts of the edges contained in T_1 is at least as high as the sum of Borda counts of the edges of T_2.

Definition 3.1 *For $T \in \tau$ we define the Borda count of T by $B(T) := \sum_{e \in T} B(e)$. Then the Borda-order $\succeq_B$ on τ is defined by letting, for all $T_1, T_2 \in \tau$,*

$$T_1 \succeq_B T_2 :\Longleftrightarrow B(T_1) \geq B(T_2) \ .$$

Analogously, a best tree with respect to the Approval-order (Plurality-order) on τ is a tree maximizing the edge-sum of Approval counts (Plurality counts).

Definition 3.2 *For $T \in \tau$ the Approval count of T is defined by $A(T) := \sum_{e \in T} A(e)$. The Approval-order $\succeq_A$ on τ is defined by letting, for all $T_1, T_2 \in \tau$,*

$$T_1 \succeq_A T_2 :\Longleftrightarrow A(T_1) \geq A(T_2) \ .$$

Definition 3.3 *For $T \in \tau$ the Plurality count of T is defined by $Pl(T) := \sum_{e \in T} Pl(e)$. Then we define the Plurality-order $\succeq_P$ on τ by letting, for all $T_1, T_2 \in \tau$,*

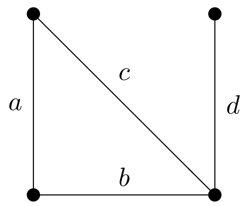
$$T_1 \succeq_P T_2 :\Longleftrightarrow Pl(T_1) \geq Pl(T_2) \ .$$

Having assigned Borda counts (Approval counts, Plurality counts) to the edges $e \in E$, a best tree with respect to the Borda-order (Approval-order, Plurality-order) is a tree with maximum Borda count (Approval count, Plurality count). This approach can be generalized as follows.

Definition 3.4 *Let τ be the set of spanning trees of G and let $\succsim$ be a weak order on E. A tree $M \in \tau$ is called max-spanning tree if and only if for every edge $f = \{i, j\}$, $f \notin M$,*

$$f \precsim e$$

holds for all $e \in M$ that are part of the unique simple path between i and j in M.



1	2	3
a	b	c
b	c	a
c	a	b
d	d	d

Figure 1: An undirected graph $G = (V, E)$ and a preference profile for three voters.

Remark. The above definition of a max-spanning tree is a generalization of the path optimality condition for the maximum spanning tree problem stated in [1]. Note that for Definition 3.4 $\succsim$ does not need to be based on numerical values; that is, there does not have to be a number assigned to each edge.

A tree with maximum Borda count (Approval count, Plurality count) and, a max-spanning tree in general, can be determined efficiently by applying greedy algorithms – such as Prim's or Kruskal's algorithm (for details see [1]) – for the maximum spanning tree problem. For example, a "generalized" version of Kruskal's algorithm to compute a max-spanning tree works as follows:
Arrange the edges $e \in E$ in non-increasing order according to $\succsim$ and iteratively add an edge to the solution set X (which is empty at the beginning) such that socially preferred edges are taken first. I.e. first add to X an edge that no other edge is socially preferred to, continue with the "next best" edge, etc. If adding an edge creates a cycle, the edge simply is ignored and we go on with the next edge.

As mentioned above, the Simple Majority-order $\succsim_{sm}$ on E however is not a weak order because in general it is not transitive, and thus preference cycles may occur. Hence the Simple Majority-order does not seem to immediately indicate an order on τ analogous to the Borda, Approval or Plurality case. Nevertheless the complete order $\succsim_{sm}$ on E can be used to compare two trees. The idea of the following concept for comparing two trees based on a given (not necessarily transitive) complete order $\succsim$ on E is to remove the edges the trees have in common and to pairwise compare the remaining edges according to $\succsim$.

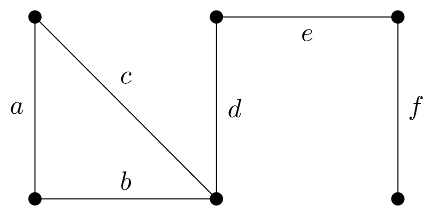
Definition 3.5 *Let $\succsim$ be a complete order on E and let $T_1, T_2 \in \tau$. Furthermore, let $\tilde{T}_1 := T_1 \setminus T_2$ and $\tilde{T}_2 := T_2 \setminus T_1$. Then we define*

$$T_1 \succsim_S T_2 \;:\Longleftrightarrow \sum_{a \in \tilde{T}_1} \sum_{b \in \tilde{T}_2} s(a, b) \geq 0 \ ,$$

where, for $a, b \in E$, $s(a, b)$ denotes the score of a versus b.

Remark. $\succsim_S$ is a complete order on τ. Furthermore it is worth mentioning that in Definition 3.5, as well as in Definition 3.4, $\succsim$ does not need to be of numerical nature, i.e. $\succsim$ does not have to allocate numbers to the edges.

Example 1 *Let $\succsim = \succsim_{sm}$. For the graph displayed in Figure 1 there exist three spanning trees: $T_1 := \{a, b, d\}$, $T_2 := \{b, c, d\}$ and $T_3 := \{a, c, d\}$. Given the preference profile in Figure 1 we get $a \succ_{sm} b$ because we have aP_1b and aP_3b, whereas only voter 2 prefers b to*



1	2	3
c	f	b
a	e	f
b	d	a
d	c	d
e	a	e
f	b	c

Figure 2: On the left side an undirected graph $G = (V, E)$ and on the right side a corresponding preference profile for three voters are displayed.

a. Analogously we get $b \succ_{sm} c$ and $c \succ_{sm} a$.
Because of $T_1 \setminus T_2 = \{a\}$ and $T_2 \setminus T_1 = \{c\}$ we get $T_2 \succ_S T_1$ due to $c \succ_{sm} a$. $T_1 \setminus T_3 = \{b\}$ and $T_3 \setminus T_1 = \{c\}$ yield $T_1 \succ_S T_3$ because of $b \succ_{sm} c$. Finally, we get $T_3 \succ_S T_2$ because $T_2 \setminus T_3 = \{b\}$ and $T_3 \setminus T_2 = \{a\}$ hold and $a \succ_{sm} b$ is satisfied. Thus we have $T_3 \succ_S T_2$, $T_2 \succ_S T_1$ and $T_1 \succ_S T_3$. Hence a best tree with respect to $\succsim_S$ does not exist in this example.

As the previous example shows, if $\succsim$ is not transitive (e.g. if $\succsim = \succsim_{sm}$) a best tree with respect to $\succsim_S$ in general does not exist because preference cycles may occur. Furthermore, as the following example shows, a tree T_g output by a greedy algorithm (a "generalized" version of Kruskal's algorithm) that is based on arranging edges e according to $|\{f \in E : e \succ f\}|$ in non-increasing order might be the *worst* tree according to $\succsim_S$, i.e. every other spanning tree T satisfies $T \succ_S T_g$.

Example 2 *Given the graph and the preference profile shown in Figure 2, the Simple Majority-order on $E = \{a, b, c, d, e, f\}$ is of the following form:*

$a \succ_{sm} b$	$b \succ_{sm} d$	$c \succ_{sm} a$	$d \succ_{sm} e$	$f \succ_{sm} c$
$a \succ_{sm} d$	$b \succ_{sm} e$	$c \succ_{sm} b$	$e \succ_{sm} c$	$f \succ_{sm} d$
$a \succ_{sm} e$	$b \succ_{sm} f$	$d \succ_{sm} c$	$f \succ_{sm} a$	$f \succ_{sm} e$

Thus, the edge f is superior to four edges according to the Simple Majority-order, the edges a and b are superior to three edges, c and d to two edges and e to one edge only. It is easy to see that a generalized version of Kruskal's algorithm that arranges the edges according to the Simple Majority-wins outputs the tree $T_g = \{f, a, b, d, e\}$. Altogether the graph contains three spanning trees, the two others being $T_1 = \{b, c, d, e, f\}$ and $T_2 = \{a, c, d, e, f\}$. We get $T_1 \setminus T_g = \{c\}$ and $T_g \setminus T_1 = \{a\}$, implying $T_1 \succ_S T_g$. Furthermore we have $T_2 \setminus T_g = \{c\}$ and $T_g \setminus T_2 = \{b\}$ and hence $T_2 \succ_S T_g$. I.e. according to $\succsim_S$ every other spanning tree of the graph is strictly preferred to T_g.

In the next section however we show that a best tree with respect to $\succsim_S$ always exists and can be computed efficiently if $\succsim$ is assumed to be a weak order on E. Thus it seems to be a reasonable approach to use Copeland's order (Slater's order, Black's order) to establish from $\succsim_{sm}$ a weak order on the set E; Copeland's order (Slater's order, Black's order) then might be used in order to determine $\succsim_S$.

4 Comparing trees

In the previous section we presented three weak orders on E that yield weak orders on τ in an intuitive way. Further we presented a complete order $\succsim_S$ on τ that consists of pairwise comparisons of edges based on a complete order $\succsim$ on E. If $\succsim$ is not transitive (e.g. relation $\succsim_{sm}$), preference cycles may occur and a best tree with regard to $\succsim_S$ might not exist. To overcome this difficulty, we presented methods to transform a complete order on E that is not transitive into a transitive order on E.

In this Section we will show that assuming $\succsim$ to be a weak order implies the existence of a best tree with respect to $\succsim_S$. Moreover, we will show the equivalence of four orders on τ that are based on a weak order $\succsim$ in the sense that a best tree with respect to one order is always a best tree with respect to each of the other orders. This implies that, for all four orders, a best tree with respect to the regarded order always exists and can be computed efficiently.

In what follows, $\succsim$ is assumed to be a given weak order on E. Note that $\succsim$ is not necessarily based on numerical values assigned to the edges.

4.1 Three more complete orders on τ

Given $T_1, T_2 \in \tau$, we use the notation $\tilde{T}_1 := T_1 \setminus T_2$, $\tilde{T}_2 := T_2 \setminus T_1$ and $r := |\tilde{T}_1|$ within Section 4.1 for convenience.

Based on the weak order $\succsim$ on E the three complete orders $\succsim_{lex}$, $\succsim_{mxn}$, and $\succsim_{ps}$ on the set τ of spanning trees of G are defined. The following maxmin-order on τ is derived from the maxmin-order on sets presented in [2]. According to the maxmin-order, a spanning tree T_1 is preferred to a spanning tree T_2 if either a "best" edge in $T_1 \setminus T_2$ is preferred to a socially most attractive edge in $T_2 \setminus T_1$ or, in case of indifference between these two edges, if the least accepted edge in $T_1 \setminus T_2$ is preferred to the socially worst edge in $T_2 \setminus T_1$.

Definition 4.1 *Let T_1, $T_2 \in \tau$. Then we define the maxmin-order $\succsim_{mxn}$ on τ by*

$$T_1 \succsim_{mxn} T_2 \;:\Longleftrightarrow\; \begin{array}{l} [\tilde{T}_1 = \emptyset \textit{ or} \\ \max \tilde{T}_1 \succ \max \tilde{T}_2 \textit{ or} \\ (\max \tilde{T}_1 \sim \max \tilde{T}_2 \textit{ and } \min \tilde{T}_1 \succsim \min \tilde{T}_2)] \end{array}$$

Analogously we define the leximax order on trees based on the leximax order on sets presented in [2].

Definition 4.2 *Let T_1, $T_2 \in \tau$. Further let $\tilde{T}_1 := \{e_1, e_2, ..., e_r\}$, $\tilde{T}_2 := \{f_1, f_2, ..., f_r\}$ such that $e_i \succsim e_{i+1}$ and $f_i \succsim f_{i+1}$ holds for $1 \le i \le r-1$. Then the leximax order $\succsim_{lex}$ on τ is defined by*

$$T_1 \succsim_{lex} T_2 \;:\Longleftrightarrow\; \begin{array}{l} [\tilde{T}_1 = \emptyset \textit{ or} \\ e_i \sim f_i \textit{ for all } 1 \le i \le r \textit{ or} \\ (\exists j \in \{1, ..., r\} \textit{ such that} \\ e_i \sim f_i \textit{ for all } i < j \textit{ and } e_j \succ f_j)] \end{array}$$

A third approach is to rank the edges of the disjoint union of the two trees according to $\succsim$. For the resulting ranking a positional scoring concept is used to compare the two regarded trees. This approach adapts the concept of the positional scoring procedures presented in [6].

Definition 4.3 *Let T_1, $T_2 \in \tau$. Further let $\tilde{T}_1 \cup \tilde{T}_2 := \{d_1, d_2, ...d_{2r}\}$ such that $d_i \succsim d_{i+1}$ holds for $1 \le i \le 2r-1$. Let $b : E \to \mathbb{R}$ be strictly increasing according to $\succsim$, that is, for $1 \le i \le 2r-1$, $b(d_i) = b(d_{i+1})$ if $d_i \sim d_{i+1}$ and $b(d_i) > b(d_{i+1})$ if $d_i \succ d_{i+1}$.*
Let $b(\tilde{T}_1) := \sum_{e \in \tilde{T}_1} b(e)$ and $b(\tilde{T}_2) := \sum_{f \in \tilde{T}_2} b(f)$.

Then we define

$$T_1 \succsim_{ps} T_2 \;:\Longleftrightarrow b(\tilde{T}_1) \ge b(\tilde{T}_2) \;.$$

Remark. The edges being ranked as in the above definition, and having assigned $b(d_j)$ scoring points to the edge in the j-th position as in Definition 4.3, a best tree with respect to $\succsim_{ps}$ is a spanning tree that maximizes the sum of scoring points.

4.2 Results

This Section is organized as follows.

Theorem 4.1 states that a max-spanning tree can be computed in polynomial time[1]. We afterwards show that a max-spanning tree (as defined in Definition 3.4) corresponds to a best tree with respect to $\succsim_S$ and to a best tree with respect to each of the orderings on τ defined in Section 4.1. Vice versa, a best tree with respect to one of these orderings always corresponds to a max-spanning tree. We summarize these results in Theorem 4.4. Together with Theorem 4.1, Theorem 4.4 implies that a best tree with respect to each of these orderings can be determined in polynomial time.

Theorem 4.1 *A max-spanning tree can be computed in $\mathcal{O}(m + n \log n)$ time.*

Proof. The Theorem immediately follows from the fact that the maximum spanning tree problem can be solved in $\mathcal{O}(m + n \log n)$ time [1]. □

Theorem 4.2 *A tree $M \in \tau$ is a max-spanning tree if and only if there is no tree $B \in \tau$ such that*

$$B \succ_{lex} M$$

holds.

Proof.
"$\Rightarrow$": Assume there exists $B \in \tau$ with $B \succ_{lex} M$. Let $B \setminus M := \{f_1, ..., f_r\}$ and $M \setminus B := \{e_1, ..., e_r\}$ for some $r \geq 1$ such that

$$f_i \succsim f_{i+1} \text{ and } e_i \succsim e_{i+1} \text{ for all } 1 \leq i \leq r-1 \ . \tag{1}$$

Obviously $B \succ_{lex} M$ implies $f_1 \succsim e_1$. We now show that $f_1 \sim e_1$ must hold.

- Assume that $f_1 \succ e_1$ holds. Because M is a tree adding f_1 to M yields a cycle K_1. Since B is a tree not all edges of K_1 can be contained in B and hence K_1 must contain an edge $\tilde{e} \in \{e_1, e_2, ..., e_r\}$. This means that there is an edge e in K_1 that satisfies $f_1 \succ \tilde{e}$ because both $f_1 \succ e_1$ and $e_1 \succsim \tilde{e}$ hold. This is a contradiction to the fact that M is a max-spanning tree.

Thus we have $f_1 \sim e_1$ and by our assumption there exists an index $1 < j \leq r$ such that

$$e_i \sim f_i \tag{2}$$

for all $i < j$ and

$$e_j \prec f_j \tag{3}$$

hold. Note that because of (1) and (2)

$$f_i \succsim e_{i+1} \tag{4}$$

[1]Note that, in case $P \neq NP$, this does not have to hold if the order $\succsim$ on E is not assumed to be given but needs to be determined from the individual preferences. For example, computing $\succsim$ using Kemeny's or Dodgson's rule is NP-hard [3]. Thus, the whole process of computing $\succsim$ according to Kemeny's rule and determining a max-spanning tree thereafter would be NP-hard.

holds for all $i < j-1$.
Adding f_{j-1} to M creates a cycle K. Because M is a max-spanning tree inequality

$$e \succsim f_{j-1} \tag{5}$$

holds for all $e \in K$. Now add K to B. Removing f_{j-1} from $B \cup K$ yields a connected graph $A := (B \cup K) \setminus \{f_{j-1}\}$ because f_{j-1} is part of K. Because B is a tree, for every cycle C contained in A an edge $e \in K$ (recall that $e \in M$ holds) must be part of C. Hence removing such edges e from A until we get an acyclic graph yields a tree A_1 which must be of the form $A_1 = B \setminus \{f_{j-1}\} \cup \{e\}$ for some $e \in M$. Note that due to (1) and (3) $f_{j-1} \succsim f_j \succ e_j$ holds. Hence (5) implies that

$$e \in \{e_1, e_2, ..., e_{j-1}\} \tag{6}$$

must hold. Now we show that $A_1 \succ_{lex} M$ must be satisfied:

- Case (i): $e = e_1$. Then $M \setminus A_1 = \{e_2, e_3, ..., e_{j-1}, e_j, ..., e_r\}$ and $A_1 \setminus M = \{f_1, f_2, ..., f_{j-2}, f_j, ..., f_r\}$. (4) states $f_i \succsim e_{i+1}$ for all $i < j-1$ and (3) states $f_j \succ e_j$ which implies $A_1 \succ_{lex} M$.
- Case (ii): $e = e_{j-1}$. Then we get $M \setminus A_1 = \{e_1, ..., e_{j-2}, e_j, ..., e_r\}$ and $A_1 \setminus M = (f_1, ..., f_{j-2}, f_j, ..., f_r)$. Obviously $A_1 \succ_{lex} M$ holds in this case since (2) and (3) hold.
- Case (iii): $e = e_k$ for some $1 < k < j-1$. In this case we have

  $$M \setminus A_1 = \{e_1, ..., e_{k-1}, e_{k+1}, ..., e_{j-1}, e_j, ..., e_r\}$$

  and $A_1 \setminus M = (f_1, ..., f_{k-1}, f_k, ..., f_{j-2}, f_j, ..., f_r)$. Again because of (2), (4) and (3) we get $A_1 \succ_{lex} M$.

In all three cases we have $A_1 \succ_{lex} M$ with $A_1 \setminus M = \{f_1, f_2, ..., f_{j-2}, f_j, ..., f_r\}$. I.e. given $B \succ_{lex} M$ we can create a tree $A_1 \succ_{lex} M$ such that $A_1 \setminus M$ equals $B \setminus M$ without edge f_{j-1}.
Repeating this procedure $j-2$ times yields a tree $A_{j-1} \succ_{lex} M$ with $A_{j-1} \setminus M = \{f_j, ..., f_r\}$. Note that $M \setminus A_{j-1} = \{e_j, ..., e_r\}$ must hold due to (6). Now recall that $f_j \succ e_j$ was stated in (3). Furthermore recall that at the beginning of this proof we showed that $f_1 \succ e_1$ does not hold. Therewith it is proven that there cannot exist a tree $C \succ_{lex} M$ with $C \setminus M = \{c_1, ..., c_p\}$, $M \setminus C = \{d_1, ..., d_p\}$ for some $p \geq 1$, where $c_i \succsim c_{i+1}$ and $d_i \succsim d_{i+1}$ hold for all $1 \leq i \leq p-1$, such that $c_1 \succ d_1$ holds. This contradicts to the existence of A_{j-1}.
"$\Leftarrow$": Assume M is not a max-spanning tree. This assumption implies the existence of an edge $e \in E \setminus M$ such that the unique cycle K in $M \cup \{e\}$ contains an edge f such that $f \prec e$ holds. Now consider the tree $T := M \cup \{e\} \setminus \{f\}$. We get $T \setminus M = \{e\}$ and $M \setminus T = \{f\}$. Thus we get $T \succ_{lex} M$ which is a contradiction. □

Remark. It should be mentioned that Bern and Eppstein [4] state without proof the observation that a minimum spanning tree lexicographically minimizes the vector of edge lengths. Assigning a real number $w(e)$ to each $e \in E$ and setting $e \succsim f :\Longleftrightarrow w(e) \leq w(f)$ for $e, f \in E$, where $\leq$ denotes the common less-than-or-equal relation on $\mathbb{R}$, this observation immediately follows from Theorem 4.2.

Theorem 4.3 *A tree $M \in \tau$ is a max-spanning tree if and only if there is no tree $B \in \tau$ such that $B \succ_S M$ holds.*

Proof.
"⇒": Assume the set $\beta := \{B \in \tau : B \succ_S M\}$ is not empty and let $E_B := B \cap M$ for all $B \in \beta$. Clearly, there exists a tree $B_1 \in \beta$ such that $|E_{B_1}| \geq |E_B|$ holds for all $B \in \beta$. Since every spanning tree contains exactly $n-1$ edges, $|E_{B_1}| = n-1$ means $B_1 = M$ in contradiction to $B_1 \in \beta$. Hence $0 \leq |E_{B_1}| \leq n-2$ holds. This implies that, for some $l \in \{1, 2, ..., n-1\}$ and some $e_i, f_i \in E$, $1 \leq i \leq l$, we have

$$\tilde{B}_1 := B_1 \setminus M = \{f_1, f_2, ..., f_l\} \text{ with } f_1 \succeq f_2 \succeq ... \succeq f_l \tag{7}$$

and

$$M_1 := M \setminus B_1 = \{e_1, e_2, ..., e_l\} \text{ with } e_1 \succeq e_2 \succeq ... \succeq e_l \; . \tag{8}$$

Note that $f_1 \preceq e_1$ must hold because of Theorem 4.2. Hence we get

$$f_i \preceq e_1 \text{ for all } 1 \leq i \leq l \; . \tag{9}$$

Adding e_1 to B_1 yields a cycle K_1. Obviously not all edges of K_1 can be contained in M and thus K_1 contains at least one edge f_j, $1 \leq j \leq l$. This means that $A := B_1 \cup \{e_1\} \setminus \{f_j\}$ is a spanning tree of G. Now we show that $A \succ_S M$:

- We have $\tilde{A} := A \setminus M = \{f_1, f_2, ..., f_{j-1}, f_{j+1}, ..., f_l\}$ and $\tilde{M} := M \setminus A = \{e_2, e_3, ..., e_l\}$ for some $1 \leq j \leq l$. Hence we get

$$\begin{aligned}\sum_{f \in \tilde{A}} \sum_{e \in \tilde{M}} s(f,e) \;&=\; \sum_{f \in \tilde{B}_1} \sum_{e \in M_1} s(f,e) \\ &\quad - \sum_{e \in M_1 \setminus \{e_1\}} s(f_j, e) \;-\; \sum_{f \in \tilde{B}_1 \setminus \{f_j\}} s(f, e_1) - s(f_j, e_1) \; .\end{aligned}$$

  Note that $\sum_{f \in \tilde{B}_1} \sum_{e \in M_1} s(f,e) > 0$ holds because of $B_1 \succ_S M$ by definition of β. What's more, we have $s(f_j, e_1) \leq 0$ due to (9). It remains to show that

$$\sum_{f \in \tilde{B}_1 \setminus \{f_j\}} s(e_1, f) - \sum_{e \in M_1 \setminus \{e_1\}} s(f_j, e) \geq 0 \tag{10}$$

  holds.
  Theorem 4.2 yields $B_1 \precsim_{lex} M$. Obviously $B_1 \sim_{lex} M$ yields $B_1 \sim_S M$ in contradiction to the definition of β and thus $B_1 \prec_{lex} M$ must hold. Hence there exists a $1 \leq k \leq l$ such that $e_i \sim f_i$ holds for $i < k$ and $e_k \succ f_k$ is satisfied.
  If $e_1 \succ f_1$ then $\sum_{f \in \tilde{B}_1 \setminus \{f_j\}} s(e_1, f) = l-1$ and (10) is satisfied since $\sum_{e \in M_1 \setminus \{e_1\}} s(f_j, e) \leq l-1$. Hence we assume $k \geq 2$ (note that $l = 1$ implies $k = 1$ and thus w.l.o.g. we assume $l \geq 2$ as well). Since $k \geq 2$ there exists an index r, $1 \leq r \leq k-1$, such that both $e_r \sim f_r$ and $e_r \succ f_{r+1}$ is satisfied, because $e_1 \succsim e_2 \succsim ... \succsim e_l$ holds (see (8)). Let q, $1 \leq q < k$, denote the smallest index such that $e_q \sim f_q$ and $e_q \succ f_{q+1}$ holds. Note that $e_{q+1} \succsim f_{q+1}$ must hold because of $B_1 \precsim_{lex} M$.
  Now we distinguish two cases:

  - $j \leq q$. Clearly we have $\sum_{f \in \tilde{B}_1 \setminus \{f_j\}} S(e_1, f) \geq l-1-(q-1) = l-q$ due to $e_1 \succ f_{q+1}$ and (9).
    Recall that due to the choice of q and k we have $e_j \sim f_j$, $e_{j+1} \sim f_{j+1}$, ..., $e_q \sim f_q$. Thus $e_j \sim e_{j+1} \sim ... \sim e_q$ holds (because $e_t \succ e_{t+1}$ for some $j \leq t \leq q-1$ contradicts to the choice of q since then $e_t \succ f_{t+1}$ holds as well). But this implies $f_j \sim e_q$ and $f_j \precsim e$ holds for all $e \in \{e_2, e_3, ..., e_q\}$. Hence $\sum_{e \in M_1 \setminus \{e_1\}} S(f_j, e) \leq l-1-(q-1) = l-q$ is satisfied as a consequence of (8).
    Herewith (10) holds.

– $j > q$. Because of $e_{q+1} \succsim f_{q+1}$ and (7) we get $f_j \precsim e_{q+1}$. Thus $f_j \precsim e$ holds for all $e \in \{e_2, e_3, ..., e_{q+1}\}$ which implies $\sum_{e \in M_1 \setminus \{e_1\}} S(f_j, e) \leq l-1-q$. Recall that we have $e_q \succ f_{q+1}$ because of the choice of q and thus $e_1 \succ f$ holds for all f_x with $q+1 \leq x \leq l$. This observation and (9) imply $\sum_{f \in \tilde{B}_1 \setminus \{f_j\}} S(e_1, f) \geq l-1-q$ and so (10) is satisfied.

Thus $A \succ_S M$ holds. But we have $|E_A| = |E_{B_1}| + 1$, which is a contradiction to the choice of B_1.
"$\Leftarrow$": Analogous to the proof of the corresponding direction of Theorem 4.2. □

Summarizing our results we state the following Theorem.

Theorem 4.4 *Let $M \in \tau$. Then the following statements are equivalent:*

1. *M is a max-spanning tree*
2. $\nexists B \in \tau : B \succ_{lex} M$
3. $\nexists B \in \tau : B \succ_S M$
4. $\nexists B \in \tau : B \succ_{mxn} M$
5. $\nexists B \in \tau : B \succ_{ps} M$

Proof. "1.$\Leftrightarrow$2." and "1.$\Leftrightarrow$3." were stated in Theorem 4.2 and Theorem 4.3.
The proofs of "4.$\Rightarrow$1." and "5.$\Rightarrow$1." are analogous to the one for "2.$\Rightarrow$1." (see proof of Theorem 4.2). There the assumption that M is not a max-spanning tree is led to a contradiction by creating a tree T that satisfies $T \succ_{lex} M$. But $T \succ_{mxn} M$ and $T \succ_{ps} M$ hold as well and thus an analogous contradiction can be created in either case.

"2.$\Rightarrow$4.": This can be shown analogous to direction "$\Rightarrow$" in Theorem 4.2, because due to that theorem a tree M that satisfies condition 2. is a max-spanning tree. However, we now create cycle K by adding f_r to M instead of f_{j-1}, which finally yields a tree $A_1 = B \setminus \{f_r\} \cup \{e\}$ for some $e \in M \setminus B$ with $e \succsim f_r$. Recall that $M \succeq_{lex} B$ holds, and thus $M \succ_{lex} B$ holds because $M \sim_{lex} B$ contradicts to our assumption $B \succ_{mxn} M$. $M \succ_{lex}$ implies $e_1 \succsim f_1$, and hence $e_1 \sim f_1$ must hold as $e_1 \succ f_1$ contradicts to $B \succ_{mxn} M$. $B \succ_{mxn}$ M and $f_1 \sim e_1$ imply $f_r \succ e_r$. Hence we have $e \in \{e_1, e_2, ..., e_{r-1}\}$ and $f_{r-1} \succsim f_r \succ e_r$. Thus, if $e \neq e_1$, we have $A_1 \succ_{mxn} M$ because of $f_1 \sim e_1$ and $f_{r-1} \succ e_r$. If $e = e_1$ we have $f_1 \succsim e_2$ and $f_{r-1} \succ e_r$ and so in either case we have $A_1 \succ_{mxn} M$. Repeating this procedure $r-2$ times yields a tree $A_{r-1} \succ_{mxn} M$ with $A_{r-1} \setminus M = \{f_1\}$ and $M \setminus A_{r-1} = \{e_r\}$. Thus $A_{r-1} \succ_{lex} M$ holds which is a contradiction.

"2.$\Rightarrow$5.": Analogous to "2.$\Rightarrow$4.", but now K is created by adding f_1 instead of f_{j-1}. Thus we get a tree $A_1 = B \cup \{e\} \setminus \{f_1\}$ with $e \succsim f_1$. Note that this implies $b(e) \geq b(f_1)$. Recall that $b(\tilde{B}) > b(\tilde{M})$ holds, where $\tilde{B} := B \setminus M$ and $\tilde{M} := M \setminus B$. Hence with $\tilde{A}_1 := A_1 \setminus M$ and $M_1 := M \setminus A_1$ we have $b(\tilde{A}_1) = b(\tilde{B}) - b(f_1) > b(\tilde{M}) - b(e) \geq b(M_1)$. By repeating this procedure $r-2$ times we get a tree $A_{r-1} \succ_{ps} M$ with $A_{r-1} \setminus M = \{f_r\}$ and $M \setminus A_{r-1} = \{\tilde{e}\}$ for some $\tilde{e} \in M$. This implies $A_{r-1} \succ_{lex} M$ which is a contradiction. □

Remark 1. Note that equivalence "1.$\Leftrightarrow$5." implies that the size of the numbers assigned to the edges is not crucial for the determination of a best tree with respect to $\succsim_{ps}$. I.e. for every assignment of numbers to edges according to Definition 4.3 the set of best trees w.r.t. $\succsim_{ps}$ is the same.
In order to determine the group ranking on E often positional scoring methods [6] are used. As a consequence of the above observation, every positional scoring method that yields

the same ranking $\succsim$ on E yields the same set of best trees w.r.t. $\succsim_{ps}$, irrespective of the numerical values assigned to the edges.

Remark 2. Due to the fact that a max-spanning tree can be determined efficiently by applying a greedy algorithm, for all orders regarded in Theorem 4.4 a socially best tree can be computed in polynomial time.

5 Conclusion

In this paper we have presented different ways to achieve orderings on the set of spanning trees from a group ranking on the edge-set of a graph. Assuming that the given group ranking is a weak order (which does not necessarily allocate a numerical value to each edge), we have shown that the sets of socially best trees according to the concepts discussed in this paper coincide.

In the related work of Perny and Spanjaard [9] a quite general framework for ranking spanning trees on the basis of preference relations is presented. In [9] a main focus is laid on establishing sufficient conditions for an order on the power set of the edge-set under which a greedy algorithm is able to determine the set of best trees with respect to the corresponding order. The orders $\succsim_{lex}$, $\succsim_{S}$, $\succsim_{mxn}$ and $\succsim_{ps}$ however[2] do not belong to the class of orders for which these conditions hold, even if the group ranking is assumed to be a partial order on the set of edges. As a consequence of our paper, a socially best tree which respect to each of these orders can be computed efficiently by simply determining a max-spanning tree using a greedy algorithm.

[2]if their definitions adequately are extended to the power set of the edge-set

References

[1] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice Hall, 1993.

[2] S. Barbera, W. Bossert, and P.K. Pattanaik. Ranking sets of objects. In *Handbook of Utility Theory*, volume 2, pages 893–977. 2004.

[3] J.J. Bartholdi, C.A. Tovey, and M.A. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6:157–165, 1989.

[4] M.W. Bern and D. Eppstein. Worst-case bounds for subadditive geometric graphs. In *Symposium on Computational Geometry*, pages 183–188, 1993.

[5] D. Black. *The Theory of Committees and Elections*. Cambridge University Press, Cambridge, 1958.

[6] S.J. Brams and P.C. Fishburn. Voting procedures. In K.J. Arrow, A.K. Sen, and K. Suzumura, editors, *Handbook of Social Choice and Welfare*, volume 1, pages 173–236. Elsevier, 2002.

[7] A.H. Copeland. *A 'Reasonable' Social Welfare Function*. Notes from a Seminar on Applications of Mathematics to the Social Sciences, University of Michigan, 1951.

[8] H. Nurmi. *Voting Procedures under Uncertainty*. Springer Verlag, Berlin, 2007.

[9] P. Perny and O. Spanjaard. A preference-based approach to spanning trees and shortest paths problems. *European Journal of Operational Research*, 127(3):584–601, May 2005.

[10] D.G. Saari. *Basic Geometry of Voting*. Springer Verlag, Berlin, 1995.

[11] P. Slater. Inconsistencies in a schedule of paired comparisons. *Biometrica*, 48:303–312, 1961.

Andreas Darmann, Christian Klamler
Institute of Public Economics
University of Graz
8010 Graz, Austria
Email: {andreas.darmann,christian.klamler}@uni-graz.at

Ulrich Pferschy
Institute of Statistics and Operations Research
University of Graz
8010 Graz, Austria
Email: pferschy@uni-graz.at