Economics Bulletin

Volume 34, Issue 1

It is hard to agree on a spanning tree

Andreas Darmann
Institute of Public Economics, University of Graz

Abstract

We consider the problem of finding a "fair" or "acceptable" spanning tree in an undirected graph when each member of a group of agents proposes a spanning tree. An "acceptable" spanning tree in that respect is a spanning tree which does not differ in more than a given number of edges from each of the single proposals. We show that, from a computational perspective, determining if such a spanning tree exists is a difficult (NP-complete) problem.

This research was supported by the Austrian Science Fund (FWF): P23724-G11.
Citation: Andreas Darmann, (2014) "It is hard to agree on a spanning tree", *Economics Bulletin*, Vol. 34 No. 1 pp. 34-40.
Contact: Andreas Darmann - andreas.darmann@uni-graz.at.
Submitted: December 17, 2013. **Published:** January 14, 2014.

1. Introduction

Spanning trees of undirected graphs play a major role in the construction of networks of various kind. Apart from the underlying optimization exercise itself, in the last decades spanning trees have received a lot of attention in connection with fairness issues. In that context, the overwhelming focus was laid on dividing the cost of a spanning tree of minimum total cost between a set of agents in a fair manner (among many others, see, e.g., the works of Bird (1976), Kar (2002), Dutta and Kar (2004), Bergantiños and Vidal-Puga (2007), and Bogomolnaia and Moulin (2010)).

The task of *constructing* a “fair” spanning tree itself, i.e., a tree which is acceptable to the whole group of agents, has received comparatively little attention, some representatives being the works of Darmann et al. (2009), Darmann (2013), and Escoffier et al. (2013). Similar to their works, we also consider the problem of finding a “fair” spanning tree from a computational perspective. However, in our context, we consider the situation in which each agent proposes a specific solution, i.e., spanning tree. The goal is to determine a spanning tree which is fair or acceptable in the sense of not differing "too much" from each of the solutions proposed by the agents.

In principle, we are thus concerned with the computational complexity involved in the aggregation of a number of spanning trees proposed by the agents into a single spanning tree.[1] In a different, preference-based environment, Endriss and Grandi (2012) consider the problem of aggregating directed graphs (proposed by agents) into a single graph; instead of focusing on computational complexity however, Endriss and Grandi (2012) choose an axiomatic viewpoint.

The problem we consider arises in situations in which a network in the form of a spanning tree needs to be constructed (e.g., sewage systems, telecommunication or power networks and pipelines of any kind), and the respective decision makers have, possibly differing, opinions on how the actual network should look like. As an example, consider the situation in which an oil pipeline system should be built, connecting all the countries involved. Each of the countries, however, (for political, economic or environmental reasons) proposes a different specific solution of how the system should connect the countries. The task now is to find a solution, i.e., a spanning tree, which all of the countries “accept”.

In our framework, we use the following intuitive measure of acceptance: an agent accepts a spanning tree T, if the number of edges that are in T but not in the spanning tree proposed by the agent, does not exceed a given upper bound. In this work, we show that it is computationally intractable to find a tree that is acceptable to all agents. This result adds to the results of Darmann et al. (2009), where the computational complexity of finding such a fair spanning tree is analyzed when agents approve or disapprove of single edges (instead of proposing a whole spanning tree).

[1] This falls into the scope of Computational Social Choice (for overviews of that area see Endriss (2011) and Lang (2005)).

2. Formal framework and problem definition

We start with the formal framework of this paper. An undirected graph $G = (V, E)$ consists of a finite set V and a set E of two-element subsets of V. We call the elements of V vertices and the elements of E edges. A cycle is a sequence of vertices $v_0, v_1, \ldots, v_n$ with $n \geq 3$ such that (i) $v_0 = v_n$, (ii) $v_i \neq v_j$ for $(i, j) \neq (0, n)$, and (iii) $\{v_{i-1}, v_i\} \in E$ for $1 \leq i \leq n$. A subset $T \subseteq E$ with $|T| = |V| - 1$ is called a spanning tree of G, if (V, T) contains no cycle.

Now, given a graph $G = (V, E)$ and a set A of agents, each $a \in A$ proposes a spanning tree T_a of G. The goal is to find a spanning tree T that minimizes the "distance" to the agents' proposals. Using the symmetric set difference as basis, there are two natural ways of approaching this problem. The first one aims at finding a spanning tree T which minimizes the total distance, i.e., minimizes $\sum_{a \in A} |T \setminus T_a|$. The second one uses a more egalitarian approach and looks for a spanning tree T that minimizes the maximum distance, i.e., minimizes $\max_{a \in A} |T \setminus T_a|$.
These two different approaches raise the question of the computational complexity involved in each of the problems. It is not hard to see that the first approach reduces to the classical maximum spanning tree problem and is hence easy to solve. In contrast, we will show that the second approach leads to a problem which is computationally difficult. Before going into details, that problem is formally defined as follows.

Definition 2.1 *(*ACCEPTABLE-TREE*)*

GIVEN:	*Set A of agents, undirected graph G, spanning trees T_a of G for $a \in A$, $k \in \mathbb{N}$.*
QUESTION:	*Is there a spanning tree T of G such that $|T \setminus T_a| \leq k$ for all $a \in A$?*

3. ACCEPTABLE-TREE is NP-complete

In this section, we show that ACCEPTABLE-TREE is an NP-complete problem. We will prove this by providing a reduction from HALF 2-SAT, a special case of the MINIMUM 2-SATISFIABILITY PROBLEM (MIN 2-SAT).

Definition 3.1 *(*HALF 2-SAT*)*

GIVEN:	*Set X of variables with $|X| = 2n$ for some $n \in \mathbb{N}$, set C of (disjunctive) clauses over X such that every clause is made up of exactly two variables.*
QUESTION:	*Is there a truth assignment τ for X that satisfies all clauses of C, such that the number of variables set to true under τ is exactly $\frac{|X|}{2}$?*

Note that in HALF 2-SAT, the clauses consist of variables and not of literals, i.e., there are no negated literals in HALF 2-SAT. First, we show that HALF 2-SAT itself is NP-complete.

Theorem 3.1 HALF 2-SAT *is* NP*-complete.*

Proof. Clearly, HALF 2-SAT is in NP. To show NP-hardness, we provide a reduction from MIN 2-SAT. Given a set X' of variables, a set C' of (disjunctive) clauses made up of exactly two variables of X', and $k \in \mathbb{N}$, MIN 2-SAT is the task to decide if there exists a truth assignment τ' setting to true at most k variables of X' that satisfies all clauses of C'. MIN 2-SAT is known to be NP-complete (see Alimonti et al. (1997)). Note that the problem is equivalent to deciding if we can satisfy all clauses in C' by setting to true *exactly* k variables[2] – we will consider that formulation of MIN 2-SAT in this proof.
Let $\mathcal{I}' = (X', C', k)$ be an arbitrary instance of MIN 2-SAT. W.l.o.g. we can assume that $|X'| = 2n$ for some $n \in \mathbb{N}$ (otherwise we can add a dummy variable).
If $k = n$, then $\mathcal{I}'$ is an instance of HALF 2-SAT and there is nothing to show.
(i) Assume $k > n$. Introduce $\ell = 2k - 2n$ new variables $y_1, \dots, y_\ell$. Consider the instance $\mathcal{I}'' = (X'', C', k)$, where $X'' = X' \cup \{y_j | 1 \leq j \leq \ell\}$. Note that $|X''| = 2n + \ell = 2k$, and hence $\mathcal{I}''$ is an instance of HALF 2-SAT. Clearly, $\mathcal{I}'$ is a "yes"-instance of MIN 2-SAT if and only if $\mathcal{I}''$ is a "yes"-instance of HALF 2-SAT.

(ii) Assume $k < n$. Introduce $\ell = 2n - 2k + 2$ new variables $y_1, \dots, y_\ell$ and the clauses $D_{i,j} := (y_i \vee y_j)$ for $1 \leq i < j \leq \ell$. Let $D := \{D_{i,j} | 1 \leq i < j \leq \ell\}$. As above, let $X'' = X' \cup \{y_j | 1 \leq j \leq \ell\}$. Let $\tilde{C} = C' \cup D$ and $\tilde{k} = k + \ell - 1$. Note that $\tilde{\mathcal{I}} = (X'', \tilde{C}, \tilde{k})$ is an instance of HALF 2-SAT, because $|X''| = 2n + \ell = 4n - 2k + 2 = 2(2n - k + 1)$ and $\tilde{k} = k + \ell - 1 = 2n - k + 1$.
Next, we show that D can be satisfied by setting exactly $\ell - 1$ variables to true, but cannot be satisfied by setting less than $\ell - 1$ variables to true:
Let τ be the truth assignment defined by setting to true exactly the variables $y_1, y_2 \dots, y_{\ell-1}$. It is easy to see that τ is a satisfying truth assignment for D, because by construction every clause in D contains one of the variables set to true under τ. On the other hand, if there are two variables y_g, y_h, for some $1 \leq g < h \leq \ell$, not set to true under a truth assignment ψ, then the clause $D_{g,h}$ – which, by construction is contained in D – is not satisfied by ψ.
Thus, D can be satisfied by setting exactly $\ell - 1$ variables to true but cannot be satisfied with setting to true a smaller number of variables. As an immediate consequence, C' can be satisfied by setting exactly k variables to true if and only if $\tilde{C}$ can be satisfied by setting exactly $\tilde{k} = k + (\ell - 1)$ variables to true. □

Now, we are ready to prove that deciding if there exists a spanning tree which is "acceptable" to all agents is NP-complete, and thus computationally difficult.

Theorem 3.2 ACCEPTABLE-TREE *is* NP*-complete.*

Proof. ACCEPTABLE-TREE is in NP: Given a certificate – i.e., a spanning tree T – it can be verified in polynomial time, if $|T \setminus T_a| \leq k$ holds for all $a \in A$.
To prove NP-hardness, we reduce HALF 2-SAT to ACCEPTABLE-TREE. Given an instance $\mathcal{I} = (X, C)$ of HALF 2-SAT, let $X = \{x_1, x_2, \dots, x_{2n}\}$ and $C = \{C_1, C_2, \dots, C_m\}$ for some $n, m \in \mathbb{N}$. Let, for $1 \leq j \leq m$, $C_j = (x_{j_1} \vee x_{j_2})$ with $1 \leq j_1 < j_2 \leq n$. From $\mathcal{I}$, we construct

[2] If we can satisfy the clauses with setting to true exactly k variables, then obviously we can satisfy them with setting to true at most k variables. If, on the other hand, the clauses can be satisfied with setting to true $r < k$ variables, then we can additionally set to true $(k - r)$ arbitrary variables to get a satisfying truth assignment that sets to true exactly k variables.

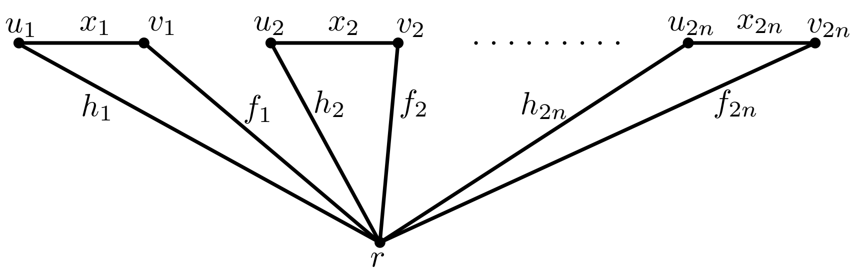

Figure 1: Graph $G = (V, E)$ in instance $\mathcal{J}$ of ACCEPTABLE-TREE

an instance $\mathcal{J}$ of ACCEPTABLE-TREE as follows.
First, we define the graph $G = (V, E)$ (see also Fig. 1): we introduce the set of vertices $V := \{r\} \cup \{u_t, v_t | 1 \leq t \leq 2n\}$ and, for $1 \leq t \leq 2n$, the edges $f_t = \{r, v_t\}$, $h_t = \{r, u_t\}$, and $x_t = \{u_t, v_t\}$ (abusing notation, we identify a variable with an edge of the same label). Hence, using the shortcuts $F = \{f_t | 1 \leq t \leq 2n\}$ and $H = \{h_t | 1 \leq t \leq 2n\}$, we formally define $E := H \cup X \cup F$. Note that $|V| = 4n + 1$, which implies that any spanning tree of G must contain exactly $|V| - 1 = 4n$ edges.
Next we introduce the set of agents $A = \{\alpha, \beta\} \cup \{\gamma_j | 1 \leq j \leq m\}$. Agent α proposes the spanning tree $T_\alpha = H \cup X$ of G, and β proposes the spanning tree $T_\beta = H \cup F$. Finally, for each $1 \leq j \leq m$, agent γ_j – representing clause C_j – proposes the spanning tree containing edges x_{j_1}, x_{j_2}, all edges in $F \setminus \{f_{j_1}, f_{j_2}\}$, and all edges in H; that is,

$$T_{\gamma_j} := H \cup \{x_{j_1}\} \cup \{x_{j_2}\} \cup (F \setminus \{f_{j_1}, f_{j_2}\})$$

In what follows, we prove that the following claim holds: $\mathcal{I} = (X, C)$ is a "yes"-instance of HALF 2-SAT if and only if in instance $\mathcal{J}$ of ACCEPTABLE-TREE there is a spanning tree T such that $|T \setminus T_a| \leq n$ for each $a \in A$.

"$\Rightarrow$": Let τ be a truth assignment that satisfies all clauses in C and sets to true exactly n variables of X. Let $\{x_{t_1}, x_{t_2}, \ldots, x_{t_n}\}$ be the set of variables set to true under τ. Consider the spanning tree T of G given by

$$T := H \cup \{x_{t_1}, x_{t_2}, \ldots, x_{t_n}\} \cup (F \setminus \{f_{t_1}, f_{t_2}, \ldots, f_{t_n}\})$$

Note that T contains exactly n edges of X and n edges of F. Thus,

$$|T \cap X| = |T \cap F| = n \tag{1}$$

We need to check that $|T \setminus T_a| \leq n$ holds for each $a \in A$. Clearly, $|T \setminus T_\alpha| = n$ and $|T \setminus T_\beta| = n$ hold. Now, consider agent γ_j for some $j \in \{1, \ldots, m\}$. For calculating $|T \setminus T_{\gamma_j}|$, note that

$$T \setminus T_{\gamma_j} = [(X \cap T) \setminus (X \cap T_{\gamma_j})] \uplus [(F \cap T) \setminus (F \cap T_{\gamma_j})] \tag{2}$$

because the edge-set H is contained in both trees. Rewriting the second set difference in the above equation yields

$$(F \cap T) \setminus (F \cap T_{\gamma_j}) = (F \cap T) \setminus (F \setminus \{f_{j_1}, f_{j_2}\}) = \{f_{j_1}, f_{j_2}\} \cap T$$

Together with $X \cap T_{\gamma_j} = \{x_{j_1}, x_{j_2}\}$, equation (2) hence becomes

$$T \setminus T_{\gamma_j} = [(X \cap T) \setminus \{x_{j_1}, x_{j_2}\}] \uplus [\{f_{j_1}, f_{j_2}\} \cap T] \tag{3}$$

Since τ is a satisfying truth assignment, at least one of the variables $\{x_{j_1}, x_{j_2}\}$ that make up clause C_j is set to true under τ. I.e., at least one of the edges $\{x_{j_1}, x_{j_2}\}$ is contained in T. We distinguish the following cases.

(i) $|\{x_{j_1}, x_{j_2}\} \cap T| = 1$: W.l.o.g. let $x_{j_1} \in T$. Hence, $x_{j_2} \notin T$. By construction of the graph G and because of $H \subset T$, this implies $f_{j_1} \notin T$ and $f_{j_2} \in T$. Thus, $\{f_{j_1}, f_{j_2}\} \cap T = \{f_{j_2}\}$. Since $x_{j_1} \in T$ and $x_{j_2} \notin T$, with (1) we get $|(X \cap T) \setminus \{x_{j_1}, x_{j_2}\}| = n - 1$. Equation (3) hence implies $|T \setminus T_{\gamma_j}| = (n-1) + 1 = n$.

(ii) $|\{x_{j_1}, x_{j_2}\} \cap T| = 2$: That is, $x_{j_1} \in T$ and $x_{j_2} \in T$. Thus, $f_{j_1} \notin T$ and $f_{j_2} \notin T$ due to $H \subset T$. As a consequence, $\{f_{j_1}, f_{j_2}\} \cap T = \emptyset$. In addition, $|(X \cap T) \setminus \{x_{j_1}, x_{j_2}\}| = n - 2$ because T contains exactly n edges of X (stated in (1)). With (3), this yields $|T \setminus T_{\gamma_j}| = (n-2) + 0 = n - 2$.

Summing up, $|T \setminus T_a| \leq n$ holds for each $a \in A$.

"$\Leftarrow$": On the other hand, let T' be a spanning tree of G with $|T' \setminus T_a| \leq n$ for each $a \in A$. First, we show that this implies the existence of a spanning tree T of G with $|T \setminus T_a| \leq n$ for each $a \in A$ such that $H \subset T$ holds:
If $H \subset T'$, there is nothing to show. Assume $H \not\subset T'$. By construction, for each $1 \leq t \leq 2n$ such that $h_t \notin T'$, we must have $\{x_t, f_t\} \subset T'$. Create T from T' by replacing, for each such index t with $\{x_t, f_t\} \subset T$, the edge f_t with h_t. Since for each agent $a \in A$, $H \subset T_a$ holds, $|T \setminus T_a| \leq |T' \setminus T_a| \leq n$ follows. Therewith, there is a spanning tree T of G with $|T \setminus T_a| \leq n$ for each $a \in A$ such that $H \subset T$ holds.
Now, consider the agents in A. Observe that $|T \setminus T_\alpha| \leq n$ implies $|X \cap T| \geq n$, and $|T \setminus T_\beta| \leq n$ implies $|F \cap T| \geq n$. Due to $H \subset T$ and the fact that $|T| = 4n$, this means that

$$|X \cap T| = |F \cap T| = n \tag{4}$$

holds.
Next, we show that for each γ_j, at least one of $\{x_{j_1}, x_{j_2}\}$ is contained in T. Assume that for some $1 \leq j \leq m$, both $x_{j_1} \notin T$ and $x_{j_2} \notin T$ hold. Since T is a spanning tree of G, we can conclude that $f_{j_1} \in T$ and $f_{j_2} \in T$ hold. As stated in (4), $|X \cap T| = n$. In particular, with the fact that $x_{j_1} \notin T$ and $x_{j_2} \notin T$, this means

$$|(X \cap T) \setminus \{x_{j_1}, x_{j_2}\})| = n \tag{5}$$

Since (i) x_{j_1}, x_{j_2} are the only edges in X contained in T_{γ_j}, and (ii) f_{j_1} and f_{j_2} are the only edges in F that are not contained in T_{γ_j}, we get

$$T \setminus T_{\gamma_j} = [(X \cap T) \setminus (X \cap T_{\gamma_j})] \uplus [(F \cap T) \setminus (F \cap T_{\gamma_j})] = [(X \cap T) \setminus \{x_{j_1}, x_{j_2}\})] \uplus \{f_{j_1}, f_{j_2}\}$$

With (5), $|T \setminus T_{\gamma_j}| = n + 2$ follows, which contradicts our assumption that $|T \setminus T_a| \leq n$ is satisfied for each $a \in A$.
Thus, for each γ_j, at least one of $\{x_{j_1}, x_{j_2}\}$ is contained in T. Hence, the truth assignment φ that sets to true exactly the variables in $X \cap T$, for each clause C_j sets to true at least one of the variables $\{x_{j_1}, x_{j_2}\}$ contained in the clause. Due to (4), φ sets to true exactly n of the $2n$ variables in X, which completes the proof. $\square$

References

Alimonti, P., Ausiello, G., Giovaniello, L., and Protasi, M. (1997). On the complexity of approximating weighted satisfiability problems. Rapporto Tecnico RAP 38.97, Dipartimento di Informatica e Sistemistica, Università degli Studi di Roma La Sapienza.

Bergantiños, G. and Vidal-Puga, J. (2007). A fair rule in minimum cost spanning tree problems. *Journal of Economic Theory*, 137(1):326–352.

Bird, C. (1976). On cost allocation for a spanning tree: a game theoretic approach. *Networks*, 6:335–350.

Bogomolnaia, A. and Moulin, H. (2010). Sharing the cost of a minimal cost spanning tree: beyond the folk solution. *Games and Economic Behavior*, 69(2):238–248.

Darmann, A. (2013). Popular spanning trees. *International Journal of Foundations of Computer Science*, 24(5):655–677.

Darmann, A., Klamler, C., and Pferschy, U. (2009). Maximizing the minimum voter satisfaction on spanning trees. *Mathematical Social Sciences*, 58(2):238–250.

Dutta, B. and Kar, A. (2004). Cost monotonicity, consistency and minimum cost spanning tree games. *Games and Economic Behavior*, 48:223–248.

Endriss, U. (2011). Computational social choice: Prospects and challenges. *Procedia Computer Science*, 7:68–72.

Endriss, U. and Grandi, U. (2012). Graph aggregation. In *Proceedings of the 4th International Workshop on Computational Social Choice (COMSOC-2012)*. AGH University of Science and Technology, Kraków.

Escoffier, B., Gourvès, L., and Monnot, J. (2013). Fair solutions for some multiagent optimization problems. *Autonomous Agents and Multi-Agent Systems*, 26(2):184–201.

Kar, A. (2002). Axiomatization of the Shapley value on minimum cost spanning tree games. *Games and Economic Behavior*, 38(2):265–277.

Lang, J. (2005). Some representation and computational issues in social choice. In *ECSQARU*, pages 15–26.