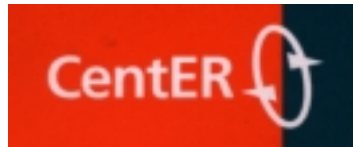


Minimum Cost Spanning Tree Games and Population Monotonic Allocation Schemes

Authors	Norde,H.W.; Moretti,S.; Tijs,S.H.
Published in	CentER Discussion Paper
Publication Date	2001
Link	https://research.tilburguniversity.edu/en/publications/794e124d-6be4-494d-a14f-4c1b5ede7e4d
Citation	Norde, H W, Moretti, S & Tijs, S H 2001 'Minimum Cost Spanning Tree Games and Population Monotonic Allocation Schemes' CentER Discussion Paper, vol. 2001-18, Microeconomics, Tilburg.
Download Date	2026-05-18 21:56:13
Rights	<p>General rights</p> <p>Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.</p> <ul style="list-style-type: none"> - Users may download and print one copy of any publication from the public portal for the purpose of private study or research. - You may not further distribute the material or use it for any profit-making activity or commercial gain - You may freely distribute the URL identifying the publication in the public portal" <p>Take down policy</p> <p>If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.</p>



No. 2001-18

**MINIMUM COST SPANNING TREE GAMES AND
POPULATION MONOTONIC ALLOCATION
SCHEMES**

By Henk Norde, Stefano Moretti and Stef Tijs

March 2001

ISSN 0924-7815

Discussion paper

Minimum cost spanning tree games and population monotonic allocation schemes

Henk Norde¹, Stefano Moretti² and Stef Tijs¹

February 28, 2001

Abstract: In this paper we present the Subtraction Algorithm that computes for every classical minimum cost spanning tree game a population monotonic allocation scheme. As a basis for this algorithm serves a decomposition theorem that shows that every minimum cost spanning tree game can be written as nonnegative combination of minimum cost spanning tree games corresponding to 0-1 cost functions. It turns out that the Subtraction Algorithm is closely related to the famous algorithm of Kruskal for the determination of minimum cost spanning trees. For variants of the classical minimum cost spanning tree games we show that population monotonic allocation schemes do not necessarily exist.

Key-words: Minimum cost spanning tree games, population monotonic allocation schemes.

1 Introduction

The objective in minimum cost spanning tree problems is the construction of a network of minimal cost which provides for every node in the network a connection with the source. Examples of minimum cost spanning tree problems are the problem of building a network of computers that connects every computer with some server or the problem of building a drainage system that connects every house in a city with the water purifier.

Another example of a minimum cost spanning tree problem is the problem of carpooling. Suppose that three employees of a firm consider the

¹Department of Econometrics and Operations Research and CentER, Tilburg University, P.O. Box 90153, 5000 LE Tilburg, The Netherlands.

²Institute for Applied Mathematics, National Research Council, Via de Marini 6, 16149 Genova, Italy.

possibility of carpooling in order to reduce their daily travel cost. The cost of driving a car from one employee to another or from one employee to the firm are given in figure 1. Here the employees are denoted by 1, 2, and 3 and

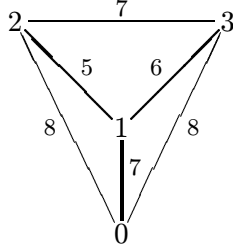


Figure 1: The cost of driving and a minimum spanning tree.

the firm by 0. A minimum cost spanning tree in this network is $\{01, 12, 13\}$ with cost 18. This tree corresponds to the plan of carpooling in which employees 2 and 3 drive their car in solitude to employee 1 where all employees take one car in order to drive together to the firm.

Having solved the problem of finding a minimum cost spanning tree the employees are confronted with the problem of how to divide the cost of 18 among the employees in a fair way. At this stage cooperative game theory enters the scene. The employees consider the *minimum cost spanning tree game* (N, c) (Bird (1976)), where $N = \{1, 2, 3\}$ and $c : 2^N \setminus \{\emptyset\} \rightarrow \mathbb{R}$ is the characteristic function which computes for every $S \in 2^N \setminus \{\emptyset\}$ the cost $c(S)$ of a network of minimal cost connecting every employee in S with the firm. So $c(123) = 18$ and, e.g., $c(23) = 15$ since $\{02, 23\}$ is a minimum spanning tree for $S = \{2, 3\}$. One way of dividing the joint costs $c(N)$ in a fair way is by means of a *core allocation* (Gillies (1953)), which is a vector $(x_i)_{i \in N}$ that is efficient, i.e. $\sum_{i \in N} x_i = c(N)$, and gives no subgroup an incentive to deviate, i.e. $\sum_{i \in S} x_i \leq c(S)$ for every $S \in 2^N \setminus \{\emptyset\}$. Bird (1976) showed how to compute a core element of a minimum cost spanning tree game. First one has to find a minimum spanning tree by means of Prim's Algorithm (Prim (1957)), which forms in every step of the algorithm an edge between a node which is not connected yet with the source and the source or a node which is already connected with the source. Secondly the Bird rule assigns the cost of an edge which forms in some step of the algorithm to the node which gets a connection with the source in that same step. In the example of figure 1 Prim's Algorithm first forms edge 01, then 12, and finally 13 and the Bird rule yields the core allocation $x = (7, 5, 6)$.

Suppose now that a fourth employee is asking whether he can join the carpoolers 1, 2, and 3. The cost of driving from employee 4 to the other employees and to the firm are given in figure 2, as well as a minimum spanning tree for the new situation. Application of the Bird rule to this new

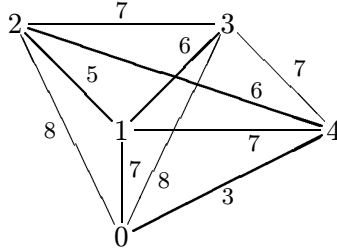


Figure 2: The cost of driving and a minimum spanning tree in the new situation.

situation yields the allocation $x = (5, 6, 6, 3)$. In the new situation employee 2 has to pay 6, whereas in the old situation he only paid 5. Therefore, if the employees use the Bird rule in order to divide joint costs, employee 2 will veto the entrance of employee 4.

The central question in this paper is whether every minimum cost spanning tree game has a *population monotonic allocation scheme (pmas)* (Sprumont (1990)), which is an allocation scheme that provides a core element for the game and all its subgames and which, moreover, satisfies a monotonicity condition in the sense that players have to pay less in larger coalitions. We will answer this question in the affirmative and we will provide the Subtraction Algorithm, that computes for every minimum cost spanning tree game a pmas. We will show that this algorithm is closely related to Kruskal's algorithm for finding a minimum spanning tree (Kruskal (1956)). The Subtraction Algorithm is based upon a decomposition theorem, which shows that every minimum cost spanning tree game can be written as a non-negative combination of minimum cost spanning tree games with 0-1 cost functions. Another approach has been followed by Kent and Skorin-Kapov (1996). They established the existence of a pmas for minimum cost spanning tree games by considering the dual of the LP-problem, corresponding to the problem of finding a minimum spanning tree for the grand coalition.

This paper is organized as follows. Some preliminaries are introduced in section 2. In section 3 the decomposition theorem is provided and section 4 focuses on minimum cost spanning tree games with 0-1 cost functions. The

Subtraction Algorithm is presented in section 5 and section 6 concludes with examples that show that variants of the classical minimum cost spanning tree games do not necessarily have a pmas.

2 Preliminaries

In this section we introduce some terminology on graphs and cooperative games.

A *complete weighted graph* is a tuple $\langle N', w \rangle$ where

- i) $N' = \{0, 1, \dots, n\}$;
- ii) $w : E \rightarrow \mathbb{R}_+$, where $E = \{S : S \subseteq N', |S| = 2\}$.

Elements of N' are called *nodes*. Node 0 is called the *source* and $N = \{1, \dots, n\}$ the set of *players*. Elements of E are called *edges* and for an $l \in E$ the nonnegative number $w(l)$ represents the weight or cost of edge l . If $w(l) \in \{0, 1\}$ for every $l \in E$ the cost function w is called *simple*. The *carrier* $Ca(w)$ of w is the set of edges with positive cost, i.e. $Ca(w) = \{l \in E : w(l) > 0\}$. A subset Γ of E is called a *network*. The cost of network Γ is $w(\Gamma) = \sum_{l \in \Gamma} w(l)$. A *path* from i to j in Γ is a sequence of nodes $i = i_0, i_1, \dots, i_k = j$ such that $\{i_s, i_{s+1}\} \in \Gamma$ for every $s \in \{0, \dots, k-1\}$. A network Γ is a *spanning network* for S ($S \subseteq N$) if for every $l \in \Gamma$ we have $l \subseteq S \cup \{0\}$ and if for every $i \in S$ there is a path in Γ from i to 0.

A *cooperative (cost) game* is a tuple (N, c) where $N = \{1, \dots, n\}$ is the set of players and $c : 2^N \setminus \{\emptyset\} \rightarrow \mathbb{R}$ its characteristic (cost) function. A *population monotonic allocation scheme* or pmas (Sprumont (1990)) of the game (N, c) is a table $x = \{x_{S,i}\}_{S \in 2^N \setminus \{\emptyset\}, i \in S}$ with the properties

- i) $\sum_{i \in S} x_{S,i} = c(S)$ for all $S \in 2^N \setminus \{\emptyset\}$;
- ii) $x_{S,i} \geq x_{T,i}$ for all $S, T \in 2^N \setminus \{\emptyset\}$ and $i \in N$ with $i \in S \subset T$.

A pmas provides a cost allocation vector for every coalition in a monotonic way, i.e. the cost allocated to some player decreases if the coalition to which he belongs becomes larger.

3 Minimum cost spanning tree games

The main aim of this section is to provide a decomposition theorem for minimum cost spanning tree games: every minimum cost spanning tree

game is a nonnegative combination of minimum cost spanning tree games which correspond to simple cost functions. First we recall the definition of minimum cost spanning tree games (see Bird (1976)).

Definition 1 Let $\langle N', w \rangle$ be a complete weighted graph. The *minimum cost spanning tree (mcst) game* (N, c) , corresponding to $\langle N', w \rangle$, is defined by

$$c(S) = \min\{w(\Gamma) : \Gamma \text{ is a spanning network for } S\}$$

for every $S \in 2^N \setminus \{\emptyset\}$.

In a mcst game the number $c(S)$ is the cost of a cheapest network, which connects every member of S with the source and which uses only edges in $S \cup \{0\}$. Always a cheapest network without cycles, i.e. a *tree*, can be chosen.

Example 1 Consider the complete weighted graph $\langle N', w \rangle$ with $N' = \{0, 1, 2, 3\}$ and cost function w as depicted in figure 3. If $S = \{2, 3\}$ then a

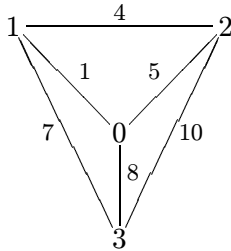


Figure 3: The cost function of Example 1.

minimum cost spanning network for S is $\Gamma = \{02, 03\}$ with cost 13, whereas a minimum cost spanning network for N is $\Gamma = \{01, 12, 13\}$ with cost 12. Proceeding in this way we find that the mcst game (N, c) , corresponding to $\langle N', w \rangle$, is given by

$$\begin{aligned} c(123) &= 12, \\ c(12) &= 5, & c(13) &= 8, & c(23) &= 13, \\ c(1) &= 1, & c(2) &= 5, & c(3) &= 8. \end{aligned}$$

If no confusion can occur about the set of nodes $N' = \{0, 1, \dots, n\}$ and hence about the set of players $N = \{1, \dots, n\}$ we will denote the complete weighted graph $\langle N', w \rangle$ shortly by its cost function w and the mcst game (N, c) by its characteristic function c . If the cost of all edges in $Ca(w)$

are lowered by the cost of an edge in $Ca(w)$ with minimal cost we are left with a cost function with smaller carrier. The following lemma establishes a relation between the corresponding mcs games.

Lemma 1 *Let w be a cost function with $Ca(w) \neq \emptyset$ and let $\alpha := \min\{w(l) : l \in Ca(w)\}$. Let w' be the simple cost function defined by $w'(l) := 1$ if $l \in Ca(w)$ and $w'(l) := 0$ otherwise. Let w'' be the cost function defined by $w''(l) := w(l) - \alpha w'(l)$ for every $l \in E$. Finally, let c, c' and c'' be the mcs games corresponding to w, w' and w'' respectively. Then we have $w = \alpha w' + w''$ and $c = \alpha c' + c''$.*

Proof It follows by definition that $w = \alpha w' + w''$. In order to prove that $c = \alpha c' + c''$, i.e. $c(S) = \alpha c'(S) + c''(S)$ for every $S \in 2^N \setminus \{\emptyset\}$, let $S \in 2^N \setminus \{\emptyset\}$. Let Γ' be a minimum cost spanning network for S in w' without cycles, i.e. Γ' is a minimum cost spanning tree for S in w' . Write $\Gamma' = L^0 \cup L^1$ where $L^0 := \{l \in \Gamma' : w'(l) = 0\}$ and $L^1 := \{l \in \Gamma' : w'(l) = 1\}$. Clearly, $|\Gamma'| = |L^0| + |L^1|$. Since Γ' is a tree we also have $|\Gamma'| = |S|$. Hence $c'(S) = w'(\Gamma') = |L^1| = |S| - |L^0|$.

It suffices to show that there exists a minimum cost spanning tree Γ'' for S in w'' with $L^0 \subseteq \Gamma''$. Since then Γ'' contains at most $|\Gamma'' \setminus L^0| = |S| - |L^0|$ edges in $Ca(w')$ and hence $w'(\Gamma'') \leq |S| - |L^0| = w'(\Gamma')$. Therefore Γ'' is also a minimum cost spanning tree for S in w' . Having $w = \alpha w' + w''$ and the fact that Γ'' is a minimum cost spanning tree for S in both w' and w'' we may conclude that Γ'' is also a minimum cost spanning tree for S in w . So, $c(S) = w(\Gamma'') = \alpha w'(\Gamma'') + w''(\Gamma'') = \alpha c'(S) + c''(S)$.

In order to show that there is a minimum cost spanning tree Γ'' for S in w'' with $L^0 \subseteq \Gamma''$ take an arbitrary minimum cost spanning tree Γ for S in w'' . If $L^0 \subseteq \Gamma$ we are done. If $L^0 \not\subseteq \Gamma$ choose an $l \in L^0 \setminus \Gamma$. Since $\Gamma \cup \{l\}$ contains a cycle C , whereas Γ' , and hence L^0 , do not contain cycles, we can find an edge $l' \in C$ with $l' \notin L^0$. Define $\tilde{\Gamma} := (\Gamma \cup \{l\}) \setminus \{l'\}$. Since $w''(l) = 0$ and $w''(l') \geq 0$ we find that also $\tilde{\Gamma}$ is a minimum cost spanning tree for S in w'' . Moreover $|\tilde{\Gamma} \cap L^0| = |\Gamma \cap L^0| + 1$. Repeating this argument results in the tree Γ'' with the desired properties. ■

Now we are able to prove the main theorem of this section.

Theorem 1 *Let w be a cost function with $Ca(w) \neq \emptyset$ and let c be the corresponding mcs game. Then there exists a sequence of simple cost functions w_1, \dots, w_k , with $Ca(w) = Ca(w_1) \supsetneq Ca(w_2) \supsetneq \dots \supsetneq Ca(w_k)$, and positive*

numbers $\alpha_1, \dots, \alpha_k$ such that

$$w = \sum_{j=1}^k \alpha_j w_j.$$

Moreover, if c_1, \dots, c_k are the mcst games corresponding to w_1, \dots, w_k respectively, we have

$$c = \sum_{j=1}^k \alpha_j c_j.$$

Proof The proof is by induction to $|Ca(w)|$.

If $|Ca(w)| = 1$ then $Ca(w)$ has a unique element, say l^* . Defining $\alpha := w(l^*)$ and the simple cost function w_1 by $w_1(l^*) := 1$ and $w_1(l) := 0$ if $l \neq l^*$ we clearly have $w = \alpha_1 w_1$. Moreover, if c_1 is the mcst game corresponding to w_1 one easily verifies that $c = \alpha_1 c_1$.

Now let $m \in \mathbb{N}, m \geq 2$ and suppose that the theorem has been proved for every cost function w with $|Ca(w)| \leq m-1$. Consider a cost function w with $|Ca(w)| = m$. According to Lemma 1 there is a simple cost function w_1 , namely the simple cost function with same carrier as w , a positive number α_1 and a cost function w'' with $Ca(w'') \subsetneq Ca(w)$ such that $w = \alpha_1 w_1 + w''$. Moreover, if c_1 and c'' are the mcst games corresponding to w_1 and w'' respectively we have $c = \alpha_1 c_1 + c''$. Application of the induction hypothesis to w'' finishes the proof. ■

Example 2 Consider the cost function w and corresponding mcst game c of Example 1. Application of Theorem 1 yields

$$w = \alpha_1 w_1 + \alpha_2 w_2 + \alpha_3 w_3 + \alpha_4 w_4 + \alpha_5 w_5 + \alpha_6 w_6$$

where $\alpha_1 = 1, \alpha_2 = 3, \alpha_3 = 1, \alpha_4 = 2, \alpha_5 = 1$ and $\alpha_6 = 2$, and the simple cost functions w_1, \dots, w_6 are specified by

edge l	01	02	03	12	13	23
$w_1(l)$	1	1	1	1	1	1
$w_2(l)$	0	1	1	1	1	1
$w_3(l)$	0	1	1	0	1	1
$w_4(l)$	0	0	1	0	1	1
$w_5(l)$	0	0	1	0	0	1
$w_6(l)$	0	0	0	0	0	1

Computing the mcst games c_1, \dots, c_6 corresponding to w_1, \dots, w_6 respectively we get

coalition S	1	2	3	12	13	23	123
$c_1(S)$	1	1	1	2	2	2	3
$c_2(S)$	0	1	1	1	1	2	2
$c_3(S)$	0	1	1	0	1	2	1
$c_4(S)$	0	0	1	0	1	1	1
$c_5(S)$	0	0	1	0	0	1	0
$c_6(S)$	0	0	0	0	0	0	0

One easily verifies that $\sum_{i=1}^6 \alpha_i c_i$ coincides with the mcst game c , as computed in Example 1.

4 Simple cost functions

In this section we will focus on simple cost functions. We will show that a mcst game corresponding to a simple cost function has a population monotonic allocation scheme. Using Theorem 1 we obtain as a corollary that every mcst game has a population monotonic allocation scheme.

Let w be a simple cost function and let $S \in 2^N \setminus \{\emptyset\}$ be a coalition. Two nodes i and j in $S \cup \{0\}$ are (w, S) -connected if there exists a sequence of nodes $i = i_0, \dots, i_k = j$ in $S \cup \{0\}$ with $w(\{i_s, i_{s+1}\}) = 0$ for every $s \in \{0, \dots, k-1\}$. A (w, S) -component of $S \cup \{0\}$ is a maximal subset of $S \cup \{0\}$ with the property that any two nodes in this subset are (w, S) -connected. The number of (w, S) -components is denoted by $n(w, S)$. Clearly the collection of (w, S) -components form a partition of $S \cup \{0\}$.

Lemma 2 *Let w be a simple cost function and let c be the corresponding mcst game. Then we have*

$$c(S) = n(w, S) - 1$$

for every $S \in 2^N \setminus \{\emptyset\}$.

Proof Let $S \in 2^N \setminus \{\emptyset\}$. If $n(w, S) = 1$ then $S \cup \{0\}$ is the unique (w, S) -component. Therefore $\Gamma = \{l \in E : l \subseteq S \cup \{0\}, w(l) = 0\}$ is a spanning network for S with $w(\Gamma) = 0$. Hence $c(S) = 0 = n(w, S) - 1$.

Now suppose $n(w, S) \geq 2$. Let C_0, C_1, \dots, C_k ($k \geq 1$) be all (w, S) -components. Clearly $S \cup \{0\} = \cup_{i=0}^k C_i$ and $n(w, S) = k + 1$. Without loss of

generality we may assume that $0 \in C_0$. For every $i \in \{1, \dots, k\}$ select some node $n_i \in C_i$. Consider the network

$$\Gamma = \{l \in E : l \subseteq S \cup \{0\}, w(l) = 0\} \cup \{\{n_i, 0\} : i \in \{1, \dots, k\}\}.$$

The network Γ is a spanning network for S : nodes in C_0 are connected with source 0 via edges in Γ of zero cost, nodes in C_i with $i \in \{1, \dots, k\}$ are connected with the source via node n_i . Moreover $w(\Gamma) = k$. It suffices to show that for any spanning tree Γ' for S we have $w(\Gamma') \geq k$, since then Γ is a minimum cost spanning network for S in w and hence we have $c(S) = w(\Gamma) = k = n(w, S) - 1$. So, let Γ' be a spanning tree for S . Define, for every $i \in \{0, \dots, k\}$, $\Gamma_i := \Gamma' \cap \{l \in E : l \subseteq C_i, w(l) = 0\}$. Since Γ' , and hence Γ_i , does not contain cycles we have $|\Gamma_i| \leq |C_i| - 1$ for every $i \in \{0, \dots, k\}$. Write $\Gamma' = L^0 \cup L^1$ where $L^0 := \{l \in \Gamma' : w(l) = 0\}$ and $L^1 := \{l \in \Gamma' : w(l) = 1\}$. Since $L^0 \subseteq \cup_{i=0}^k \Gamma_i$ we have

$$|L^0| \leq \sum_{i=0}^k |\Gamma_i| \leq \sum_{i=0}^k |C_i| - (k+1) = |S| + 1 - (k+1) = |S| - k.$$

Therefore

$$w(\Gamma') = |L^1| = |\Gamma'| - |L^0| = |S| - |L^0| \geq k.$$

■

Example 3 Consider the complete weighted graph $\langle N', w \rangle$ with $N' = \{0, \dots, 8\}$ and simple cost function w specified by $\{l \in E : w(l) = 0\} = \{01, 23, 24, 34, 45, 67\}$. Let c be the corresponding mcst game. The edges with zero cost are depicted in figure 4. Clearly, $\{0, 1\}$, $\{2, 3, 4, 5\}$, $\{6, 7\}$ and

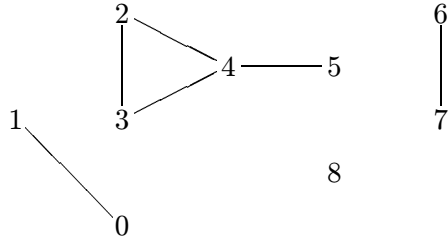


Figure 4: The cost function of Example 3.

$\{8\}$ are all (w, N) -components. Therefore $c(N) = n(w, N) - 1 = 4 - 1 = 3$.

If we consider for example coalition $S = \{2, 3, 5, 6\}$ we get that $\{0\}$, $\{2, 3\}$, $\{5\}$ and $\{6\}$ are all (w, S) -components. Therefore we also have $c(S) = n(w, S) - 1 = 4 - 1 = 3$.

In order to show that a mcst game corresponding to a simple cost function has a pmas we need some more notation. A bijection $\pi : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ is called a *permutation of $N = \{1, \dots, n\}$* . The set of all permutations of N is denoted by P_N . If w is a simple cost function, $S \in 2^N \setminus \{\emptyset\}$ and $i \in S$ then the (w, S) -component to which i belongs is denoted by $C_i(w, S)$.

Definition 2 Let w be a simple cost function and let $\pi \in P_N$. The vector $x^{\pi, w} = (x_{S,i}^{\pi, w})_{S \in 2^N \setminus \{\emptyset\}, i \in S}$ is defined in the following way:

$$x_{S,i}^{\pi, w} = \begin{cases} 0 & \text{if } 0 \in C_i(w, S) \\ 0 & \text{if } 0 \notin C_i(w, S) \text{ and } \pi(i) \neq \min_{j \in C_i(w, S)} \pi(j) \\ 1 & \text{if } 0 \notin C_i(w, S) \text{ and } \pi(i) = \min_{j \in C_i(w, S)} \pi(j) \end{cases}$$

for every $S \in 2^N \setminus \{\emptyset\}$, $i \in S$.

The vector $x^{\pi, w}$ provides for every coalition $S \in 2^N \setminus \{\emptyset\}$ a division of the cost $c(S)$ in the following way: all members of the (w, S) -component containing the source 0 do not have to pay anything whereas the (unit) cost of all other (w, S) -components is allocated to the member in the component with the lowest index according to π .

Example 4 Consider the simple cost function w of Example 3 and let $\pi \in P_N$ be given by $\pi(1) = 2$, $\pi(2) = 7$, $\pi(3) = 5$, $\pi(4) = 3$, $\pi(5) = 6$, $\pi(6) = 8$, $\pi(7) = 1$ and $\pi(8) = 4$. Then $x_{N,1}^{\pi, w} = x_{N,2}^{\pi, w} = x_{N,3}^{\pi, w} = x_{N,5}^{\pi, w} = x_{N,6}^{\pi, w} = 0$ and $x_{N,4}^{\pi, w} = x_{N,7}^{\pi, w} = x_{N,8}^{\pi, w} = 1$. Moreover, for $S = \{2, 3, 5, 6\}$ we get $x_{S,2}^{\pi, w} = 0$ and $x_{S,3}^{\pi, w} = x_{S,5}^{\pi, w} = x_{S,6}^{\pi, w} = 1$.

In the following lemma we prove that the vector $x^{\pi, w}$ is a pmas for the mcst game corresponding to simple cost function w .

Lemma 3 Let w be a simple cost function, c the corresponding mcst game, and $\pi \in P_N$. Then $x^{\pi, w}$ is a pmas for c .

Proof Let $S \in 2^N \setminus \{\emptyset\}$. Every (w, S) -component which does not contain the source 0 contains precisely one player $i \in S$ with $x_{S,i}^{\pi, w} = 1$. Therefore

$$\sum_{i \in S} x_{S,i}^{\pi, w} = n(w, S) - 1 = c(S).$$

Now let $i \in N$ and $S, T \in 2^N \setminus \{\emptyset\}$ be such that $i \in S \subset T$. In order to show that $x_{S,i}^{\pi,w} \geq x_{T,i}^{\pi,w}$ it suffices to show that $x_{T,i}^{\pi,w} = 1$ implies $x_{S,i}^{\pi,w} = 1$. So, assume $x_{T,i}^{\pi,w} = 1$, i.e. $0 \notin C_i(w, T)$ and

$$\pi(i) = \min_{j \in C_i(w, T)} \pi(j).$$

Obviously we have $C_i(w, S) \subseteq C_i(w, T)$, which implies $0 \notin C_i(w, S)$ and

$$\pi(i) = \min_{j \in C_i(w, S)} \pi(j).$$

Therefore $x_{S,i}^{\pi,w} = 1$. ■

As a corollary we get the main theorem of this paper.

Theorem 2 *Every mcs game has a pmas.*

Proof The theorem follows directly from Theorem 1, Lemma 3 and the observation that if $x^1 = (x_{S,i}^1)_{S \in 2^N \setminus \{\emptyset\}, i \in S}$ is a pmas for c^1 and $x^2 = (x_{S,i}^2)_{S \in 2^N \setminus \{\emptyset\}, i \in S}$ for c^2 then $\alpha x^1 + \beta x^2 := (\alpha x_{S,i}^1 + \beta x_{S,i}^2)_{S \in 2^N \setminus \{\emptyset\}, i \in S}$ is a pmas for $\alpha c^1 + \beta c^2$ for every $\alpha \geq 0$ and $\beta \geq 0$. ■

5 Algorithms

Two famous algorithms for the determination of a minimum cost spanning tree are the algorithm of Prim (Prim (1957)) and the algorithm of Kruskal (Kruskal (1956)). In order to describe both algorithms briefly let w be a cost function and let $S \in 2^N \setminus \{\emptyset\}$. A minimum cost spanning tree for S can be obtained in the following two ways.

Prim's Algorithm: In the first step form an edge of minimal cost between a node in S and the source 0. In every subsequent step form an edge of minimal cost between a node in S which is not connected yet with the source, directly or indirectly, and the source or with a node in S which is already connected with the source, directly or indirectly. In every step of the algorithm there is precisely one node in S which gets a connection with the source, so the algorithm stops after precisely $|S|$ steps.

Kruskal's Algorithm: In the first step form an edge between

nodes in $S \cup \{0\}$ of minimal cost. In every subsequent step form an edge between nodes in $S \cup \{0\}$ of minimal cost which does not form a cycle with the edges which have already been formed. The algorithm also stops after precisely $|S|$ steps.

The algorithm of Prim has proven its use in cost allocation. If one assigns the cost of an edge, which is formed in some step of the algorithm, to the player who gets a connection with the source, directly or indirectly, in that same step then one obtains a core element of the corresponding mcst game (see Bird (1976) for more details). In the following example we will demonstrate that such a procedure does not necessarily generate a pmas of the corresponding mcst game.

Example 5 Consider the complete weighted graph $\langle N', w \rangle$ with $N' = \{0, 1, 2, 3\}$ and cost function w as depicted in figure 5. Application of Prim's

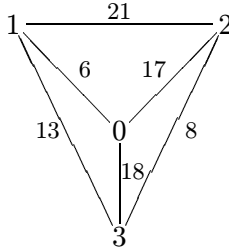


Figure 5: The cost function of Example 5.

algorithm for $N = \{1, 2, 3\}$ yields the formation of edge 01 first, followed by the formation of edge 13 and edge 23. The cost of edge 01 is assigned to player 1, the cost of edge 13 to player 3 and the cost of edge 23 to player 2. Following the same procedure for all other coalitions we get the following table

S	1	2	3
123	6	8	13
12	6	17	*
13	6	*	13
23	*	17	8
1	6	*	*
2	*	17	*
3	*	*	18

This table does not provide a pmas of the corresponding mcst game: in coalition $S = \{2, 3\}$ player 3 has to pay 8 which is strictly less than the amount 13 which he has to pay in the larger coalition $N = \{1, 2, 3\}$.

A basis for an algorithm that finds a pmas in any mcst game is provided by Theorem 1 and Lemma 3. First decompose a cost function w with $Ca(w) \neq \emptyset$ as a positive combination of simple cost functions w_1, \dots, w_k , with a strictly decreasing carrier $Ca(w_1) \supsetneq \dots \supsetneq Ca(w_k)$:

$$w = \sum_{j=1}^k \alpha_j w_j.$$

Theorem 1 tells us that the same decomposition is true for the mcst games c and c_1, \dots, c_k , corresponding to w and w_1, \dots, w_k respectively:

$$c = \sum_{j=1}^k \alpha_j c_j.$$

Subsequently, fix some permutation $\pi \in P_N$. Compute, for every $i \in \{1, \dots, k\}$ the vector x^{π, w_i} . According to Lemma 3 the vector x^{π, w_i} is a pmas for c_i for every $i \in \{1, \dots, k\}$. Consequently the vector

$$x^{\pi, w} := \sum_{j=1}^k \alpha_j x^{\pi, w_j}$$

is a pmas for mcst game c . For the sake of completeness note that for $w = 0$ the vector $x^{\pi, w} := 0$ is a pmas for the corresponding minimum cost spanning tree game $c = 0$. An alternative way of describing this algorithm is the following.

Subtraction Algorithm for the computation of a pmas of a mcst game.

Initialisation: Let $\langle N', w \rangle$ be a complete weighted graph and let $\pi \in P_N$. Define $x = \{x_{S,i}\}_{S \in 2^N \setminus \{\emptyset\}, i \in S}$ by $x_{S,i} := 0$ for every $S \in 2^N \setminus \{\emptyset\}, i \in S$.

```

Algorithm:  WHILE   $w \neq 0$ 
            DO     $\alpha := \min\{w(l) : l \in E, w(l) > 0\}$ 
                  for every  $S \in 2^N \setminus \{\emptyset\}, i \in S$ :
                    IF       $0 \notin C_i(w, S)$  and
                           $\pi(i) = \min_{j \in C_i(w, S)} \pi(j)$ 
                    THEN    $x_{S,i} := x_{S,i} + \alpha$ 
                    END
                  for every  $l \in E$  with  $w(l) > 0$ :
                     $w(l) := w(l) - \alpha$ 
            END

```

In the following example we illustrate the Subtraction Algorithm.

Example 6 Consider the complete weighted graph of Example 5 and let $\pi \in P_N$ be given by $\pi(i) = i$ for every $i \in N$. In every step of the Subtraction Algorithm some of the coefficients $x_{S,i}$ will be raised by some amount α . Which coefficients $x_{S,i}$ will be raised? Coefficient $x_{S,i}$ will be raised if there is no path in $S \cup \{0\}$ of zero cost from i to source 0 ($0 \notin C_i(w, S)$), and if there is no path in $S \cup \{0\}$ of zero cost from i to some node $j \in S$ with $\pi(j) < \pi(i)$.

In the first step of the algorithm $\alpha = 6$, the cost of edge 01. Since all edges have positive cost all coefficients $x_{S,i}$ will be raised by 6. At the end of step 1 the cost of every edge will be lowered by 6, so $w(01) = 0$.

In the second step of the algorithm $\alpha = 2$, the cost of edge 23. Since edge 10 is a path from 1 to source 0 of cost zero all coefficients $x_{S,1}$ with $1 \in S$ will not be raised in this step (and in subsequent) steps, whereas all other coefficients are raised by 2. At the end of step 2 the cost of every edge with positive cost will be lowered by 2, so $w(23) = w(01) = 0$.

In the third step of the algorithm $\alpha = 5$, the cost of edge 13. Since edge 32 is a path of zero cost which connects player 3 with player 2, which has a lower index according to π ($\pi(2) < \pi(3)$), the coefficients $x_{123,3}$ and $x_{23,3}$ will not be raised in this and further steps. All coefficients, which remain to be raised, are increased by 5 and at the end of step 3 the cost of every edge with positive cost will be lowered by 5, so $w(13) = w(23) = w(01) = 0$.

In step 4 we have $\alpha = 4$, the cost of edge 02. Since player 2 is $(w, 123)$ -connected with the source, via path 23, 31, 10, and player 3 is $(w, 13)$ -connected with the source, via path 31, 10, the corresponding coefficients will not be raised anymore, whereas all coefficients, which remain to be raised, are increased by 4. At the end of step 4 the cost of every edge with positive cost will be lowered by 4, so $w(02) = w(13) = w(23) = w(01) = 0$.

In step 5 we have $\alpha = 1$, the cost of edge 03. Since edge 20 is a path from 2 to source 0 of zero cost the coefficients $x_{12,2}$, $x_{23,2}$, and $x_{2,2}$ will not be raised any further. The only coefficient, which remains to be raised, $x_{3,3}$, is increased by 1. At the end of step 5 every edge with positive cost is lowered by 1, so $w(02) = 0$.

In step 6 we have $\alpha = 3$, the cost of edge 12. Since edge 30 is a path from 3 to source 0 of zero cost coefficient $x_{3,3}$ will not be raised anymore. The cost of the only edge with positive cost, edge 12, is lowered by 3 and the algorithm stops. The pmas of the corresponding mcst game, created in the Subtraction Algorithm, is given by

S	1	2	3
123	6	6 + 2 + 5	6 + 2
12	6	6 + 2 + 5 + 4	*
13	6	*	6 + 2 + 5
23	*	6 + 2 + 5 + 4	6 + 2
1	6	*	*
2	*	6 + 2 + 5 + 4	*
3	*	*	6 + 2 + 5 + 4 + 1

=

S	1	2	3
123	6	13	8
12	6	17	*
13	6	*	13
23	*	17	8
1	6	*	*
2	*	17	*
3	*	*	18

Note that this scheme coincides with the one in Example 5, except for the coefficients $x_{123,2}$ and $x_{123,3}$, which have been interchanged. Note moreover that player 2 in coalition $\{1, 2, 3\}$ has to pay the cost of edge 13, although he does not belong to this edge.

In every step of the Subtraction Algorithm a multiple of a simple cost function is subtracted from cost function w . Moreover, the same multiple of the pmas of the mcst game corresponding to this simple cost function is added to table x .

Remark Consider a complete weighted graph $\langle N', w \rangle$ where, in order to simplify arguments, all edges have different positive cost. Let $\pi \in P_N$ and let $x^{\pi, w}$ be the pmas generated by the Subtraction Algorithm. Let $y = (x_{N,i}^{\pi, w})_{i \in N}$ be the corresponding core allocation for the grand coalition N . Moreover, let $\Gamma = \{l_1, \dots, l_n\}$ be the unique minimum cost spanning tree for N with $w(l_1) < w(l_2) < \dots < w(l_n)$. So, according to Kruskal's Algorithm, edge l_1 forms in the first step, edge l_2 in the second step, etcetera. Let $i_1 \in N$ be the unique player which is connected via network $\Gamma_1 = \{l_1\}$ with the source 0 or with some node $j \in N$ with $\pi(j) < \pi(i_1)$, let i_2 be the unique player in $N \setminus \{i_1\}$ which is connected via network $\Gamma_2 = \{l_1, l_2\}$ with the source 0 or with some node $j \in N$ with $\pi(j) < \pi(i_2)$, etcetera. Note

that in Example 6 we have $l_1 = 01$, $l_2 = 23$, $l_3 = 13$, and $i_1 = 1$, $i_2 = 3$ and $i_3 = 2$. One easily verifies that $y_{i_k} = w(l_k)$ for every $k \in \{1, \dots, n\}$. Stated differently, the Subtraction Algorithm allocates the cost of an edge which forms in some step of Kruskal's Algorithm to the player which gets a connection with the source or with a player with a lower index according to π .

6 Variants of mcst games

In this final section we will consider some variants of minimum cost spanning tree games. One variant is the class of *monotonic minimum cost spanning tree games* which are characterized by the fact that coalitions are allowed to use networks which contain nodes outside the coalition. Two other variants are obtained by considering directed weighted graphs. Here the aim of coalitions is to construct a directed network such that every player in the coalition is connected with the source via a directed path. This approach leads to the class of *directed minimum cost spanning tree games* and *monotonic directed minimum cost spanning tree games*. For any of these new classes of games we will present an example that does not have a pmas. However, for special cases of directed networks Moretti et al. (2001) have established existence of a pmas.

First we consider the class of monotonic minimum cost spanning tree games.

Definition 3 Let $\langle N', w \rangle$ be a complete weighted graph. The *monotonic minimum cost spanning tree game* (N, c^{mon}) , corresponding to $\langle N', w \rangle$, is defined by

$$c^{\text{mon}}(S) = \min\{w(\Gamma) : \Gamma \text{ is a spanning network for some coalition } T \supseteq S\}$$

for every $S \in 2^N \setminus \{\emptyset\}$.

In the following example we present a monotonic minimum cost spanning tree game without a pmas.

Example 7 Consider the complete weighted graph $\langle N', w \rangle$ with $N' = \{0, 1, 2, 3, 4, 5, 6\}$ and cost function w as depicted in figure 6. All edges which are depicted have cost 1, whereas all other edges have cost 10. A minimum cost spanning tree for $S = \{1, 2, 3\}$ is $\{04, 05, 14, 24, 35\}$ so $c^{\text{mon}}(123) = 5$. A minimum cost spanning tree for $S = \{1, 2\}$ is $\{04, 14, 24\}$ so $c^{\text{mon}}(12) = 3$.

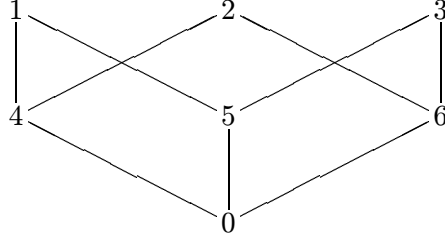


Figure 6: The cost function of Example 7.

In a similar way one gets $c^{\text{mon}}(13) = c^{\text{mon}}(23) = 3$. If c^{mon} has a pmas $\{x_{S,i}\}_{S \in 2^N \setminus \{\emptyset\}, i \in S}$ then

$$\begin{aligned}
10 &= 2c^{\text{mon}}(123) \\
&= 2(x_{123,1} + x_{123,2} + x_{123,3}) \\
&\leq x_{12,1} + x_{13,1} + x_{12,2} + x_{23,2} + x_{13,3} + x_{23,3} \\
&= x_{12,1} + x_{12,2} + x_{13,1} + x_{13,3} + x_{23,2} + x_{23,3} \\
&= c^{\text{mon}}(12) + c^{\text{mon}}(13) + c^{\text{mon}}(23) \\
&= 9,
\end{aligned}$$

which yields a contradiction.

In order to provide the definition of directed minimum cost spanning tree games and monotonic directed minimum cost spanning tree games we need some more terminology. A *complete directed weighted graph* is a tuple $\langle N', w \rangle$ where

- i) $N' = \{0, 1, \dots, n\}$;
- ii) $w : D \rightarrow \mathbb{R}_+$, where $D = \{(i, j) : i, j \in N', i \neq j\}$.

Elements of D are called *directed arcs*. A *directed path* from i to j in network $\Gamma \subseteq D$ is a sequence of nodes $i = i_0, i_1, \dots, i_k = j$ such that $(i_s, i_{s+1}) \in \Gamma$ for every $s \in \{0, \dots, k-1\}$. Network Γ is a *spanning network* for S ($S \subseteq N$) if for every $(i, j) \in \Gamma$ we have $\{i, j\} \subseteq S \cup \{0\}$ and if for every $i \in S$ there is a directed path in Γ from i to 0.

Definition 4 Let $\langle N', w \rangle$ be a complete directed weighted graph. The *directed minimum cost spanning tree game* (N, c) , corresponding to $\langle N', w \rangle$, is defined by

$$c(S) = \min\{w(\Gamma) : \Gamma \text{ is a spanning network for } S\}$$

for every $S \in 2^N \setminus \{\emptyset\}$, whereas the *monotonic directed minimum cost spanning tree game* (N, c^{mon}) , corresponding to $\langle N', w \rangle$, is defined by

$$c^{\text{mon}}(S) = \min\{w(\Gamma) : \Gamma \text{ is a spanning network for some coalition } T \supseteq S\}$$

for every $S \in 2^N \setminus \{\emptyset\}$.

We conclude this section with two examples which show that directed minimum cost spanning tree games and monotonic directed minimum cost spanning tree games do not necessarily have a pmas.

Example 8 Consider the complete directed weighted graph $\langle N', w \rangle$ with $N' = \{0, 1, 2, 3, 4, 5, 6\}$ and cost function w as depicted in figure 7. All directed arcs which are depicted have cost 0 whereas all other directed arcs have cost 1. Let (N, c) be the directed minimum cost spanning tree game,

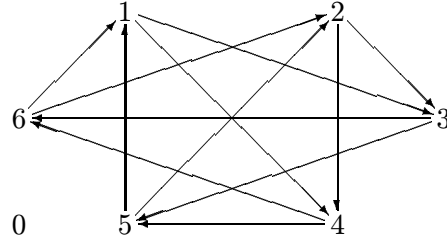


Figure 7: The cost function of Example 8.

corresponding to $\langle N', w \rangle$, and suppose that $x = \{x_{S,i}\}_{S \in 2^N \setminus \{\emptyset\}, i \in S}$ is a pmas for c . A minimum cost spanning network for N is obtained by taking all directed arcs with cost 0 and directed arc $(1,0)$. So $c(123456) = 1$. Now consider $S = \{1, 3, 4\}$. We have

$$\begin{aligned} 2 = c(134) &= x_{134,1} + x_{134,3} + x_{134,4} \\ &\leq x_{14,1} + x_{3,3} + x_{14,4} \\ &= x_{14,1} + x_{14,4} + x_{3,3} \\ &= c(14) + c(3) \\ &= 1 + 1 = 2, \end{aligned}$$

and hence $x_{134,3} = x_{3,3} = c(3) = 1$. Since also $c(13) = c(4) = 1$ we get in a similar way that $x_{134,4} = 1$. Therefore $x_{134,1} = c(134) - x_{134,3} - x_{134,4} = 0$ and hence, by population monotonicity, $x_{N,1} \leq x_{134,1} = 0$. By considering respectively coalitions 234, 356, 456, 512 and 612 we get via

analogous arguments that the numbers $x_{N,2}, \dots, x_{N,6}$ are all nonpositive. This contradicts however that $\sum_{i \in N} x_{N,i} = c(N) = 1$.

Example 9 Consider the complete directed weighted graph $\langle N', w \rangle$ with $N' = \{0, 1, 2, 3, 4, 5, 6\}$ and cost function w as depicted in figure 8. All directed arcs which are depicted have cost 0, whereas all other directed arcs have cost 1. Let c^{mon} be the monotonic directed minimum cost spanning

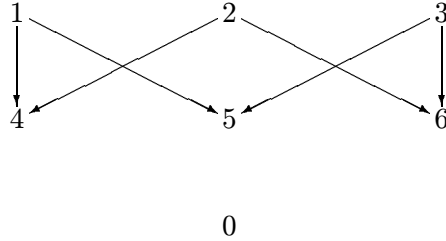


Figure 8: The cost function of Example 9.

tree game corresponding to $\langle N', w \rangle$. A minimum cost spanning network for $S = \{1, 2, 3\}$ is $\{(1, 4), (2, 4), (3, 5), (4, 0), (5, 0)\}$ so $c^{\text{mon}}(123) = 2$. A minimum cost spanning network for $S = \{1, 2\}$ is $\{(1, 4), (2, 4), (4, 0)\}$ so $c^{\text{mon}}(12) = 1$. In a similar way one gets $c^{\text{mon}}(13) = c^{\text{mon}}(23) = 1$. Since $2c^{\text{mon}}(123) > c^{\text{mon}}(12) + c^{\text{mon}}(13) + c^{\text{mon}}(23)$ we conclude in a similar way as in Example 7 that c^{mon} has no pmas.

In Moretti et al. (2001) a special subclass of directed minimum cost spanning tree problems is considered, which show up in considering the problem of connecting houses in the mountains with a water purifier. It is shown that the games corresponding to these problems always have a pmas.

References

- Bird, C.G. (1976), On cost allocation for a spanning tree: A game theoretic approach, *Networks* 6, 335-350.
- Gillies, D.B. (1953), Some theorems on n -person games, Ph.D. Dissertation, Princeton University Press, Princeton, New Jersey.
- Kent, K.J. and D. Skorin-Kapov (1996), Population monotonic cost allocations on MSTs, Discussion Paper, State University of New York at Stony Brook.

Kruskal, J.B. (1956), On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society* 7, 48-50.

Moretti, S., Norde, H., Pham Do, K.H. and S. Tijs (2001), Connection problems in mountains and monotonic allocation schemes, CentER Discussion Paper, Tilburg University.

Prim, R.C (1957), Shortest connection networks and some generalizations. *Bell Systems Technical Journal* 36, 1389-1401.

Sprumont, Y. (1990), Population monotonic allocation schemes for cooperative games with transferable utility, *Games and Economic Behavior* 2, 378-394.