CentER

Discussion Paper

No. 2007–89

A VERTEX ORIENTED APPROACH TO MINIMUM COST SPANNING TREE PROBLEMS

By Bariş Çiftçi, Stef Tijs

November 2007

ISSN 0924-7815

TILBURG UNIVERSITY

A Vertex Oriented Approach to Minimum Cost Spanning Tree Problems*

Barış Çiftçi† and Stef Tijs

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

Abstract

In this paper we consider spanning tree problems, where n players want to be connected to a source as cheap as possible. We introduce and analyze ($n!$) vertex oriented construct and charge procedures for such spanning tree situations leading in n steps to a minimum cost spanning tree and a cost sharing where each player pays the edge which he chooses in the procedure. The main result of the paper is that the average of the $n!$ cost sharings provided by our procedure is equal to the P-value for minimum cost spanning tree situations introduced and characterized by Branzei et al. (2004). As a side product, we find a new method, the vertex oriented procedure, to construct minimum cost spanning trees.

Keywords: Minimum cost spanning tree games, algorithm, value, cost sharing.
JEL code: C71, D72

*We thank Rodica Branzei and Stefano Moretti for helpful and inspiring comments.

† *E-mail addresses:* B.B.Ciftci@uvt.nl (B.B. Ciftci), S.H.Tijs@uvt.nl (S. Tijs).

1 Introduction

Consider a group of agents that needs to be connected directly or via other agents to a unique supplier of a source. Assume that the construction of the links is costly. Then, the first important question is how to find the cheapest set of links that will connect each agent to the source. This question constitutes one of the most well-known problems of combinatorial optimization: the minimum cost spanning tree (mcst) problem. The operations research literature on mcst problems has provided many algorithmic solutions to the problem and has discussed the computational properties of these solutions. We can mention, for example, the two most famous algorithms, the Kruskal algorithm (Kruskal,1956) and the Prim algorithm (Prim, 1957). An historic overview of the algorithms provided for the mcst problem can be found in Graham and Hell (1985).

If the cost of the construction is to be covered by the agents, the second important question that arises in mcst situations is how to allocate the cost of the mcst among the agents in a fair way. This cost allocation problem is introduced in the economics literature by Claus and Kleitman (1973). The seminal paper by Bird (1976) provided the first game theoretical treatment of this problem by associating a coalitional game with transferable utility to mcst problems. Then, solution concepts of cooperative game theory are implemented in this game and proposed as appropriate cost allocations for mcst problems by several studies: Granot and Huberman (1981,1984) analyzed the core and the nucleolus; Kar (2002) studied the Shapley value of this game. Recently, Bergantiños and Vidal-Puga (2007) associated another coalitional game with mcst problems and studied the Shapley value.

However, cost allocation rules for mcst problems can also be defined directly without appealing to the underlying cost game. In particular, one can make use of an algorithm to construct a mcst and allocate the cost of each edge constructed by the algorithm among the agents by following an appropriate method. Cost allocation rules which follow such a procedure are called construct and charge rules in Moretti et al. (2005). Construct and charge rules proposed in the literature mainly focus on the two well-known algorithms, the Kruskal algorithm and the Prim algorithm, in order to construct a mcst: In particular, the Bird rule (Bird, 1976) and the extended Bird rule (Dutta and Kar, 2004) rely on the Prim algorithm while the P-value (Branzei et al., 2004), the equal remaining obligations rule (Feltkamp et al., 1994a,b) and the obligation rules (Tijs et al., 2006) rely on the Kruskal algorithm.

In this paper we consider a new construct and charge procedure, which we call the vertex oriented construct and charge procedure, for spanning tree situations leading to a mcst and a cost sharing where each player pays the edge which he chooses in the procedure. Consider mcst problems, where n players $1, ..., n$ want to be connected to a source 0 as cheap as possible. Given an ordering $\sigma = (\sigma(1), \sigma(2), ..., \sigma(n))$ of the players the n-step procedure is as follows. In step 1 player $\sigma(1)$ connects with one of the vertices $0, \sigma(2), ..., \sigma(n)$ in a way as cheap as possible. Assume that in the steps $1, 2, ..., k$ already a forest is constructed by the players $\sigma(1), ..., \sigma(k)$ with k edges. Then, player $\sigma(k+1)$ belongs to one of the trees of this forest and has to construct an edge as cheap as possible avoiding a cycle starting from a point of the tree to which he belongs. It turns out that this procedure indeed leads to a mcst. Let us clarify the situation with an example where 3 players $1, 2, 3$ want to be connected directly or indirectly with a source 0 and where the cost situation is represented in Figure 1. Take the ordering $\sigma = (1, 2, 3)$. In step 1 player 1 constructs and pays the

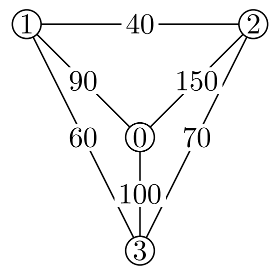

Figure 1: An mcst situation with three agents

cheapest of the edges $(1, 0), (1, 2), (1, 3)$ which is edge $(1, 2)$ with cost 40. Then player 2 constructs and pays edge $(1, 3)$ with cost 60 and this is the cheapest of his allowed edges $(2, 0), (2, 3), (1, 0), (1, 3)$. Finally player 3 chooses edge (1,0) which is the cheapest of his allowed edges $(3, 0), (1, 0), (2, 0)$. Note that $(3, 2)$ is not allowed for player 3 because it generates a cycle. The result is the mcst with edges $(1, 2), (1, 3), (1, 0)$ and the cost share vector $(40, 60, 90)$. In the next table we see the construct and charge results for all orderings of the players.

	Constructed edges by			Costs for		
σ	1	2	3	1	2	3
(1,2,3)	(1,2)	(1,3)	(1,0)	40	60	90
(1,3,2)	(1,2)	(1,0)	(1,3)	40	90	60
(2,1,3)	(1,3)	(1,2)	(1,0)	60	40	90
(2,3,1)	(1,0)	(1,2)	(1,3)	90	40	60
(3,1,2)	(1,2)	(1,0)	(1,3)	40	90	60
(3,2,1)	(1,0)	(1,2)	(1,3)	90	40	60

Table 1: Construct and charge results for the mcst situation in Figure 1

The vertex oriented construct and charge procedure provides a cost allocation for each ordering of the players. However, the cost allocation provided by our procedure for a particular ordering of the players can be considered as unfair, since the right to construct an edge between two players is first given to the one which precedes the other in the ordering. A typical method of achieving fairness for the allocations depending on the ordering of players is averaging the allocations over the set of all orderings. Hence, we focus on the

V-value, the average of the cost allocations provided by the vertex oriented procedure over the set of all orderings of players. The main result of our study is that the V-value is equal to the P-value for mcst situations introduced and characterized by Branzei et al. (2004). [1]

The vertex oriented construct and charge procedure gives a new method to construct mcst's. Hence, we will now try to provide a brief comparison of the vertex oriented construct and charge procedure with the Prim and the Kruskal algorithms and with construct and charge rules relying on these algorithms. The Prim algorithm can be described as follows: In every iteration of the Prim algorithm, a player who is not connected yet with the source constructs an edge between herself and either the source or another player which is already connected with the source in the previous iterations of the algorithm. Hence, the Prim algorithm is also vertex oriented. Moreover, similar to our algorithm, every player has the right to construct the cheapest allowed edge. But, the main difference is that the set of edges allowed for construction by the Prim algorithm is restricted to the ones which provide a connection with the source. The Bird rule assigns the cost of an edge constructed in some iteration of the Prim algorithm to the player which constructs that edge and gets a connection with the source in that same iteration. For example, in the mcst problem represented in Figure 1, player 1 constructs and pays the edge $(1, 0)$ with cost 90 according to the Bird rule because player 1 has the cheapest direct connection with the source. However, player 1 is adjacent to all the edges contained in the unique mcst of the problem and the edge $(1, 0)$ is the most expensive one among these edges. From this aspect, the Bird rule can be considered as unfair. A similar unfairness argument holds for the generalization of the Bird rule to mcst problems involving more than one mcst's provided by Dutta and Kar (2004).

The Kruskal algorithm selects and adds edges to the spanning tree in increasing order of their costs such that an edge is added only if it does not create a cycle with the previously added edges. Hence, the Kruskal algorithm is an edge oriented algorithm. In other words, the decision on the construction of an edge is taken by the researcher in the Kruskal algorithm. In the vertex oriented construct and charge procedure and in the Prim algorithm, this decision is left to the players. Construct and charge rules relying on the Kruskal algorithm specify what fraction of the cost an edge constructed by the Kruskal algorithm will be paid by each player. However, the V-value determines these fractions by averaging the cost allocations corresponding to the orderings of the players.

The outline of the paper is as follows. Section 2 provides some elementary graph theoretical definitions and defines mcst problems and the P-value formally. Section 3 formally introduces the vertex oriented construct and charge procedure and presents two preliminary results. In Section 4, we prove the coincidence of the V-value with the P-value. We conclude in Section 5.

[1] The P-value coincides with the Equal Remaining Obligations rule which has been introduced in Feltkamp et al. (1994) for minimum cost spanning extension problems.

2 Preliminaries

An (undirected) graph G is an ordered pair $\langle V, E\rangle$, where $V = V(G)$ is a nonempty and finite set of vertices and $E = E(G)$ is a set of edges $\{i,j\}$ with $i,j \in V, i \neq j$. The *complete graph* on a set V of vertices is the graph $\langle V, E_V\rangle$, where $E_V = \{\{i,j\}|i,j \in V, i \neq j\}$. A *walk* between vertices i and j in a graph $G = \langle V, E\rangle$ is a sequence of vertices $i = i_0, i_1, ..., i_k = j, k \geq 1$, such that $\{i_s, i_{s+1}\} \in E$ for each $s \in \{0, ..., k-1\}$. A *path* between vertices i and j in a graph G is a walk between vertices i and j in which all edges are distinct. A *cycle* in G is a path from i to i for some $i \in V$. Two nodes $i, j \in V$ are said to be *connected* in G if $i = j$ or if there exists a path between i and j in G. G is called *connected* if, for all $i, j \in V$, G contains a path between i and j. Given a path $P = (i_0, i_1, ..., i_k)$ between vertices i and j in graph G, we say that an edge $\{u, v\} \in E$ is on path P if there exists $m \in \{0, 1, ..., k-1\}$ such that $u = i_m$ and $v = i_{m+1}$ or $v = i_m$ and $u = i_{m+1}$. For any graph $G = \langle V, E\rangle$ which does not contain any cycles and for vertices $i, j \in V$ which are connected in G, the unique path between i and j in G is denoted by $P_G(i,j)$. With an abuse of notation, we denote the set of edges on path $P_G(i,j)$ by $P_G(i,j)$, too.

A *subgraph* of $G = \langle V, E\rangle$ is a graph $G' = \langle V', E'\rangle$ with $V' \subset V$ ($V' \neq \emptyset$) and $E' \subset E$. A *restriction* of $G = \langle V, E\rangle$ to $V' \subset V$ ($V' \neq \emptyset$) is a subgraph $\left\langle V', E_{|V'}\right\rangle$ of G where $E_{|V'} = \{\{u,v\} \in E | u \in V' \text{ and } v \in V'\}$. A *component* G' of G is a maximally connected subgraph of G, i.e., G' is the only connected subgraph of G containing G' as a subgraph. The component of a graph G which contains vertex i is denoted by $C_i(G)$. A connected graph which does not contain any cycles is called a *tree*. A subgraph $G' = \langle V', E'\rangle$ of G is called a *spanning tree* in G if it is a tree with $V' = V$. We denote the set of spanning trees of $\langle V, E\rangle$ by $\Gamma^{\langle V,E\rangle}$.

In this paper we consider mcst situations in which a group of agents is willing to be connected to a supplier of a service (source) as cheap as possible. Every mcst situation can be represented by a tuple $\langle V, E_V, w\rangle$, where $\langle V, E_V\rangle$ is a complete graph on $V = \{0, 1, ..., n\}$ which is the union of the agent set $N = \{1, ..., n\}$ and the source 0 to be connected. The function $w : E_V \to \mathbb{R}_+$ is called a *weight function* and associates with each edge $e \in E_V$ the weight $w(e)$ which represents the cost of constructing e. Obviously, the minimum cost network that would connect all the agents to the source has to form a spanning tree of $\langle V, E_V\rangle$. Because, if this is not the case, then the network contains a cycle and removal of any link from this cycle will result in a cheaper network which still connects every agent to the source. Therefore, given a mcst problem $\langle V, E_V, w\rangle$, we are interested in finding a spanning tree of $\langle V, E\rangle$ with minimal cost, i.e., a minimum cost spanning tree of $\langle V, E\rangle$. Formally, the cost of a spanning tree, Γ is given by $c(\Gamma) = \sum_{e\in E(\Gamma)} w(e)$ and Γ is called a mcst if it satisfies $c(\Gamma) = \min_{\Gamma' \in \Gamma^{\langle V,E\rangle}} c(\Gamma')$.

Observe that an mcst situation with agent set N, $\langle V, E_V, w\rangle$, can be identified with the weight function, w. Hence, we will denote the set of mcst situations with agent set N as $\mathcal{W}^V = \mathbb{R}_+^{E_V}$.

We will use the following well-known results in graph theory about trees.

Theorem 2.1 *Let $\langle V, E_V, w\rangle$ be a mcst situation.*

(1) (Gondran and Minoux, 1984, Property 2, p.132) Let Γ be a subgraph of $\langle V, E_V\rangle$. Then, Γ is a spanning tree of $\langle V, E_V\rangle$ if and only if Γ has $|V|-1$ edges and does not contain any cycle.

(2) (Gondran and Minoux, 1984, Theorem 4, p.137) A spanning tree Γ of $\langle V, E_V\rangle$ is minimal if and only if $w(e) \geq w(f)$ for every $e \in E_V \backslash E(\Gamma)$ and every $f \in P_\Gamma(e)$.

2.1 The P-value for Cost Sharing in mcst Situations

Each mcst situation involves the construction of a mcst as well as the allocation of the cost of the mcst among its users in a fair way. Branzei, Moretti, Norde and Tijs (2004) introduce and characterize the P-value to solve the cost sharing problem in mcst situations. The P-value makes use of the Kruskal algorithm in order to construct a mcst. In the following, we will provide the notation and the definitions required to introduce the P-value.

Let $\Pi(E_V)$ stand for the set of all bijections $\pi : \{1, ..., |E_V|\} \to E_V$. Obviously, for each mcst situation $\langle V, E_V, w\rangle$, there exists a bijection $\pi \in \Pi(E_V)$ that orders the edges in increasing order with respect to their costs, i.e., $w(\pi(1)) \leq w(\pi(2)) \leq ... \leq w(\pi(|E_V|))$. The column vector $(w(\pi(1)), w(\pi(2)), ..., w(\pi(|E_V|)))^t$ is denoted by w^π.

For any $\pi \in \Pi(E_V)$, one can define the set $K^\pi = \{w \in \mathbb{R}_+^{E_V} | w(\pi(1)) \leq w(\pi(2)) \leq ... \leq w(\pi(|E_V|))\}$, i.e., the set of weight functions which result in the same increasing order on the set of edges with respect to their costs. It can easily be observed that K^π is a cone in $\mathbb{R}_+^{E_V}$ which is called in Branzei et al. (2004) as the *Kruskal cone with respect to* π. Obviously, $\bigcup_{\pi\in\Pi(E_V)} K^\pi = \mathbb{R}_+^{E_V} = \mathcal{W}^V$.

Branzei et al. (2004) introduce the P-value in two steps. First a value P^π is defined on each cone K^π ($\pi \in \Pi(E_V)$) and then it is proved that these P^π-values can be patched together to the whole cone of mcst situations.

Let $\langle V, E_V, w\rangle$ be an mcst situation and $\pi \in \Pi(E_V)$ be such that $w \in K^\pi$. In order to define the P^π-value on K^π, we will consider the Kruskal algorithm when it selects the edges with respect to order π. The P^π-value distributes the cost of the edges that are constructed by the Kruskal algorithm among the agents whose *connectivity*, i.e., the number of nodes in N that an agent is connected, increases with the construction of the edge. To do so, we will consider a sequence of $|E_V|+1$ graphs: $\langle V, F^{\pi,0}\rangle, \langle V, F^{\pi,1}\rangle, ..., \langle V, F^{\pi,|E_V|}\rangle$ such that $F^{\pi,0} = \emptyset$ and $F^{\pi,k} = F^{\pi,k-1} \cup \{\pi(k)\}$ for every $k \in \{1, ..., |E_V|\}$. The connectivity of an agent i in $\langle V, F^{\pi,k}\rangle$ is denoted by $n_i(F^{\pi,k})$. Note that $n_i(F^{\pi,k}) = 1$ when i is not connected to any other agent in N in $\langle V, F^{\pi,k}\rangle$. The P^π-value will distribute the cost of a Kruskal edge proportionally to the change in the *connection vectors* resulting from the introduction of the edge by the algorithm. Connection vectors $b^{\pi,k} \in \mathbb{R}^N$ are defined for each $k \in \{0, 1, ..., |E_V|\}$

as follows

$$b_i^{\pi,k} = \begin{cases} 0 & \text{if } i \text{ is connected to 0 in } \langle V, F^{\pi,k} \rangle \\ \frac{1}{n_i(F^{\pi,k})} & \text{otherwise} \end{cases} \tag{1}$$

for each $i \in N$.

The contribution matrix with respect to $\pi \in \Pi(E_V)$ is the matrix $M^\pi \in \mathbb{R}^{N \times E_V}$ where the rows correspond to the agents and the colums to the edges. It lists the change in the connectivity of the agents, i.e., the k-th column of M^π equals

$$M^\pi e^k = b^{\pi,k-1} - b^{\pi,k} \tag{2}$$

for each $k \in \{1, ..., |E_V|\}$. Here e^k stands for the column vector such that $e_i^k = 1$ if $i = k$ and $e_i^k = 0$ for each $i \in \{1, ..., |E_V|\} \backslash \{k\}$.

Example 2.1 *Consider the mcst situation $\langle V, E_V, w \rangle$ with $V = \{0, 1, 2, 3\}$ and w as depicted in Figure 1. $w \in K^\pi$, with $\pi(1) = \{1, 2\}$, $\pi(2) = \{1, 3\}$, $\pi(3) = \{2, 3\}$, $\pi(4) = \{0, 1\}$, $\pi(5) = \{0, 3\}$ and $\pi(6) = \{0, 2\}$.*

The sequence of the graphs $\langle V, F^{\pi,k} \rangle$ formed by Kruskal algorithm and the corresponding connection vectors are given in the table below.

$\langle V, \emptyset \rangle$	$b^{\pi,0} = (1, 1, 1)^t$
$\langle V, \{\{1, 2\}\} \rangle$	$b^{\pi,1} = (\frac{1}{2}, \frac{1}{2}, 1)^t$
$\langle V, \{\{1, 2\}, \{1, 3\}\} \rangle$	$b^{\pi,2} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^t$
$\langle V, \{\{1, 2\}, \{1, 3\}, \{2, 3\}\} \rangle$	$b^{\pi,3} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^t$
$\langle V, \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{0, 1\}\} \rangle$	$b^{\pi,4} = (0, 0, 0)^t$
$\langle V, \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{0, 1\}, \{0, 3\}\} \rangle$	$b^{\pi,5} = (0, 0, 0)^t$
$\langle V, \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{0, 1\}, \{0, 3\}, \{0, 2\}\} \rangle$	$b^{\pi,6} = (0, 0, 0)^t$

Then the contribution matrix M^π is given by

$$M^\pi = \begin{pmatrix} \frac{1}{2} & \frac{1}{6} & 0 & \frac{1}{3} & 0 & 0 \\ \frac{1}{2} & \frac{1}{6} & 0 & \frac{1}{3} & 0 & 0 \\ 0 & \frac{2}{3} & 0 & \frac{1}{3} & 0 & 0 \end{pmatrix}.$$

◇

Observe that the zero columns in M^π correspond to the edges which are rejected in the Kruskal algorithm. Moreover, each column $M^\pi e^k$ with $(M^\pi e^k)_i \neq 0$ for some $i \in N$ corresponds to the edge $\pi(k)$ constructed at stage k in the Kruskal algorithm. Notice that the sum of the elements of such a column equals 1. Then, $(M^\pi e^k)_i$ $(i \in N)$, the difference between i's connectivity resulting from the construction of $\pi(k)$, represents the fraction of the cost of the edge $\pi(k)$ to be paid by agent i.

We are now ready to define the P^π-value on K^π. For each $\pi \in \Pi(E_V)$, the P^π-value is defined as the map $P^\pi : K^\pi \to \mathbb{R}^N$, where $P^\pi(w) = M^\pi w^\pi$ for each mcst situation w in the cone K^π.

Branzei et al. (2004) show that it is possible to patch these P^π-values by the help of the following lemma. We include the proof for the sake of completeness.

Lemma 2.1 *Let $\pi \in \Pi(E_V)$, $w \in K^\pi$. Assume that $w_t^\pi = w_{t+1}^\pi$ for some $t \in \{1, ..., |E_V| - 1\}$. Then for the ordering $\pi' \in \Pi(E_V)$ such that $\pi'(t) = \pi(t+1), \pi'(t+1) = \pi(t)$ and $\pi'(i) = \pi(i)$ for every $i \in \{1, ..., |E_V|\} \backslash \{t, t+1\}$, we have that $w \in K^{\pi'}$ and $P^\pi(w) = P^{\pi'}(w)$.*

Proof. It is obvious that $w \in K^{\pi'}$. Put $a = w_t^\pi$. Note that $b^{\pi,k} = b^{\pi',k}$ for all $k \in \{1, ..., |E_V|\} \backslash \{t\}$. Hence, $w_k^\pi M^\pi e^k = w_k^{\pi'} M^{\pi'} e^k$ for all $k \in \{1, ..., |E_V|\} \backslash \{t, t+1\}$ and

$$\begin{aligned}
& w_t^{\pi'} M^{\pi'} e^t + w_{t+1}^{\pi'} M^{\pi'} e^{t+1} \\
&= a(b^{\pi',t-1} - b^{\pi',t}) + a(b^{\pi',t} - b^{\pi',t+1}) \\
&= a(b^{\pi',t-1} - b^{\pi',t+1}) = a(b^{\pi,t-1} - b^{\pi,t+1}) \\
&= a(b^{\pi,t-1} - b^{\pi,t}) + a(b^{\pi,t} - b^{\pi,t+1}) \\
&= w_t^\pi M^\pi e^t + w_{t+1}^\pi M^\pi e^{t+1}.
\end{aligned} \tag{3}$$

So, $M^\pi w^\pi = M^{\pi'} w^{\pi'}$ and hence, $P^\pi(w) = P^{\pi'}(w)$. $\square$

Notice that the allocation of the cost of a single edge by the P-value may change with the order on the set of edges constructed by the Kruskal algorithm. However, Lemma 2.1, as can easily be seen from equality (3), implies that the P-value allocates the cost of the edges which have the same cost in the same way independent of the order considered. Hence, for every order that a weight function is compatible with, the P-value results in the same allocation. This is stated in the following proposition.

Proposition 2.1 *If $w \in K^\pi \cap K^{\pi'}$ with $\pi, \pi' \in \Pi(E_V)$, then $P^\pi(w) = P^{\pi'}(w)$.*

Finally, the P-value is defined as the map $P : \mathcal{W}^V \to \mathbb{R}^N$, where

$$P(w) = P^\pi(w) = M^\pi w^\pi \tag{4}$$

for every $w \in \mathcal{W}^V$ and $\pi \in \Pi(E_V)$ such that $w \in K^\pi$.

Example 2.2 *Consider the mcst situation in Example 2.1. $w^\pi = (40, 60, 70, 90, 100, 150)^t$. Hence, $P(w) = M^\pi w^\pi = (60, 60, 70)^t$.* $\diamond$

3 The Vertex Oriented Construct and Charge Procedure

In this section, we first provide the formal definition of the vertex oriented procedure which we call the V-algorithm. Then, we will show that the V-algorithm gives a new method to construct a mcst for every mcst situation and moreover, every mcst in a mcst situation can be constructed by the V-algorithm.

Let $\Pi(N)$ stand for the set of all bijections $\sigma : N \to N$, where $\sigma(i) = j$ means that player j is in the i-th position with respect to σ.

Let $\langle V, E_V, w\rangle$ be a mcst situation. Then the V-algorithm for mcst situations is defined as follows:

(**Step 1**) Pick $\sigma \in \Pi(N)$.
(**Step 2**) Set $V_i^0 = \{i\}$ for each $i \in V$ and set $\Gamma^{\sigma,0} = \emptyset$.
(**Step 3**) For $k = 1$ to n:

- Choose an edge $e_{\sigma(k)} = \{u_{\sigma(k)}, v_{\sigma(k)}\}$ with $u_{\sigma(k)} \in V_{\sigma(k)}^{k-1}$, $v_{\sigma(k)} \in V \setminus V_{\sigma(k)}^{k-1}$ and $w(e_{\sigma(k)}) \leq w(\{u', v'\})$ for all $\{u', v'\} \in E_V$ with $u' \in V_{\sigma(k)}^{k-1}$, $v' \in V \setminus V_{\sigma(k)}^{k-1}$.
- For all $j \in V_{v_{\sigma(k)}}^{k-1}$ and for all $i \in V_{\sigma(k)}^{k-1}$: Set $V_i^k = V_j^k = V_{\sigma(k)}^{k-1} \cup V_{v_{\sigma(k)}}^{k-1}$.
- For all $j \in V \setminus \left(V_{v_{\sigma(k)}}^{k-1} \cup V_{\sigma(k)}^{k-1}\right)$: Set $V_j^k = V_j^{k-1}$.
- Set $\Gamma^{\sigma,k} = \Gamma^{\sigma,k-1} \cup \{e_{\sigma(k)}\}$.

(**Step 4**) Set $\Gamma^\sigma = \langle V, \Gamma^{\sigma,n}\rangle$ and $v^\sigma = (w(e_i))_{i=1}^n$.

Example 3.1 *Consider the mcst situation $\langle V, E_V, w\rangle$ with $V = \{0, 1, 2, 3\}$ and w as depicted in Figure 1.*

Let $\sigma \in \Pi(N)$ be such that $\sigma(i) = i$ for every $i \in N$. The related V-algorithm is described as follows:

(Step 1) *Let $\sigma \in \Pi(N)$ be such that $\sigma(i) = i$ for every $i \in N$.*
(Step 2) *$V_i^0 = \{i\}$ for every $i \in V$ and $\Gamma^{\sigma,0} = \emptyset$.*
(Step 3) *Step 3 consists of the following three iterations:*

- *$k = 1$: $\sigma(1) = 1$ and $V_1^0 = \{1\}$. Then, $e_1 = \{1, 2\}$; $V_1^1 = V_2^1 = \{1, 2\}$; $\Gamma^{\sigma,1} = \{\{1, 2\}\}$. $V_0^1 = V_0^0$ and $V_3^1 = V_3^0$.*
- *$k = 2$: $\sigma(2) = 2$ and $V_2^1 = \{1, 2\}$. Then, $e_2 = \{1, 3\}$; $V_1^2 = V_2^2 = V_3^2 = \{1, 2, 3\}$; $\Gamma^{\sigma,2} = \{\{1, 2\}, \{1, 3\}\}$. $V_0^2 = V_0^1$.*
- *$k = 3$: $\sigma(3) = 3$ and $V_3^2 = \{1, 2, 3\}$. Then, $e_3 = \{0, 1\}$; $V_1^3 = V_2^3 = V_3^3 = V_0^3 = \{0, 1, 2, 3\}$; $\Gamma^{\sigma,3} = \{\{1, 2\}, \{1, 3\}, \{0, 1\}\}$.*

(Step 4) *$\Gamma^\sigma = \langle V, \Gamma^{\sigma,3}\rangle$ and $v^\sigma = (40, 60, 90)$.* $\diamond$

We start our analysis of the V-algorithm with the following two preliminary results:

(i) The V-algorithm generates an efficient solution for mcst problems, i.e., it provides a mcst for every mcst situation.

(ii) Every mcst can be constructed by the V-algorithm.

Theorem 3.1 *For every mcst situation $\langle V, E_V, w\rangle \in \mathcal{W}^V$ and permutation on the set of players, $\sigma \in \Pi(N)$, the V-algorithm results in a mcst $\Gamma^\sigma = \langle V, \Gamma^{\sigma,n}\rangle$ of $\langle V, E_V, w\rangle$.*

Proof. Pick $\langle V, E_V, w\rangle \in \mathcal{W}^V$ and $\sigma \in \Pi(N)$. We will prove that $\Gamma^\sigma = \langle V, \Gamma^{\sigma,n}\rangle$ as obtained in Step 4 of the V-algorithm is a mcst of $\langle V, E_V\rangle$. It's obvious that Γ^σ has n edges and does not contain any cycle. Hence, it immediately follows from result (1) of Theorem 2.1 that Γ^σ is a spanning tree of $\langle V, E_V\rangle$. It remains to show that Γ^σ is of minimal weight.

We will prove this by induction: Assume that there exists a mcst Γ_k of $\langle V, E_V\rangle$ which contains $\Gamma^{\sigma,k}$ for every $k \in \{1, 2, ..., m-1\}$ ($m \in \{2, ..., n\}$). First, let's show that the induction hypothesis is true for $k = 1$, i.e., there exists a mcst of G which contains $\Gamma^{\sigma,1} = \{e_{\sigma(1)}\} = \{\{u_{\sigma(1)}, v_{\sigma(1)}\}\}$. Suppose not. Notice first that $V^0_{\sigma(1)} = \{\sigma(1)\}$. Pick a mcst Γ of $\langle V, E_V\rangle$. Then Γ contains a path $P_\Gamma(e_{\sigma(1)})$ which connects $u_{\sigma(1)}$ and $v_{\sigma(1)}$. Then, by result (2) of Theorem 2.1, $w(e) \leq w(e_{\sigma(1)})$ for every $e \in P_\Gamma(e_{\sigma(1)})$. However, there exists an edge $\{\sigma(1), \tilde{v}\} \in P_\Gamma(e_{\sigma(1)})$ for some $\tilde{v} \in V \setminus \{\sigma(1)\}$ and $w(\{\sigma(1), \tilde{v}\}) \geq w(e_{\sigma(1)})$ by the selection of $e_{\sigma(1)}$ by the V-algorithm. But then $\Gamma \cup e_{\sigma(1)} \setminus \{\sigma(1), \tilde{v}\}$ is again a mcst of G which contains $e_{\sigma(1)}$, a contradiction.

Now, let $e_{\sigma(m)} = \{u_{\sigma(m)}, v_{\sigma(m)}\}$ with $u_{\sigma(m)} \in V^{m-1}_{\sigma(m)}$ be the edge that is constructed in the m^{th} step by the V-algorithm. If $e_{\sigma(m)}$ is contained in Γ_{m-1}, then we are done. Hence, assume that $e_{\sigma(m)} \notin \Gamma_{m-1}$. Then, Γ_{m-1} contains the unique path $P_{\langle V,\Gamma_{m-1}\rangle}(e_{\sigma(m)})$. Obviously, $P_{\langle V,\Gamma_{m-1}\rangle}(e_{\sigma(m)})$ has to contain another edge $\tilde{e} = \{\tilde{u}, \tilde{v}\}$ with $\tilde{u} \in V^{m-1}_{\sigma(m)}$ and $\tilde{v} \in V \setminus V^{m-1}_{\sigma(m)}$. Now, on the one hand, result (2) of Theorem 2.1 implies that $w(e_{\sigma(m)}) \geq w(\tilde{e})$ while, on the other hand, the choice of $e_{\sigma(m)}$ implies that $w(e_{\sigma(m)}) \leq w(\tilde{e})$. Then, $\Gamma_m = \Gamma_{m-1} \cup e_m \setminus \tilde{e}$ is a mcst of G which contains $e_{\sigma(m)}$. Hence, we can conclude that Γ^σ is a mcst of $\langle V, E_V\rangle$. □

Theorem 3.2 *Let $\langle V, E_V, w\rangle$ be a mcst situation and Γ be a mcst of $\langle V, E_V\rangle$. Then Γ can be constructed by the V-algorithm for any permutation $\sigma \in \Pi(N)$.*

Proof. Let Γ be a mcst of G and suppose that it can not be constructed by the V-algorithm for $\sigma \in \Pi(N)$. Starting with σ construct Γ by using the V-algorithm as far as possible. Then, there exists $k \in \{1, ..., n-1\}$ such that $e_{\sigma(j)} \in E(\Gamma)$ for every $j \in \{1, ..., k\}$ and $e_{\sigma(k+1)} = \{u_{\sigma(k+1)}, v_{\sigma(k+1)}\} \notin E(\Gamma)$. Then, there exists $\tilde{e} = \{\tilde{u}, \tilde{v}\} \in P_\Gamma(e_{\sigma(k+1)})$ such that $\tilde{u} \in V^k_{\sigma(k+1)}$ and $\tilde{v} \notin V^k_{\sigma(k+1)}$. Moreover, $w(e_{\sigma(k+1)}) < w(\tilde{e})$ by the choice of $e_{\sigma(k+1)}$. Then, $\langle V, E(\Gamma) \cup e_{\sigma(k+1)} \setminus \tilde{e}\rangle$ is a spanning tree of G with total weight less than that of Γ, a contradiction. □

4 The V-Value

The V-value, $v : \mathcal{W}^V \to \mathbb{R}^N$, is defined by

$$v(w) = \frac{\sum_{\sigma\in\Pi(N)} v^\sigma}{n!} \tag{5}$$

for each $w \in \mathcal{W}^V$, where v^σ is the allocation vector provided by the V-algorithm with respect to $\sigma \in \Pi(N)$. Our main result in this paper is the coincidence of the P-value with the V-value. In order to present this result, we need the following two lemmas.

Lemma 4.1 *Let $\langle V, E_V, w\rangle$ be a mcst situation and $\sigma \in \Pi(N)$. Let $k \in \{1, 2, ..., n\}$ be such that both $V^{k-1}_{\sigma(k)} \backslash \{\sigma(k)\} \neq \emptyset$ and $V \backslash V^{k-1}_{\sigma(k)} \neq \emptyset$. Let $\{u, v\} \in E_V$ with $u \in V^{k-1}_{\sigma(k)} \backslash \{\sigma(k)\}$ and $v \in V \backslash V^{k-1}_{\sigma(k)}$. Then, $w(\{u, v\}) \geq w(e)$ for every $e \in P_{\langle V,\Gamma^{\sigma,k-1}\rangle}(\sigma(k), u)$.*

Proof. Pick a mcst situation $\langle V, E_V, w\rangle$, $\sigma \in \Pi(N)$ and $k \in \{1, 2, ..., n\}$. Assume that both $V^{k-1}_{\sigma(k)} \backslash \{\sigma(k)\} \neq \emptyset$ and $V \backslash V^{k-1}_{\sigma(k)} \neq \emptyset$. Pick $\{u, v\} \in E_V$ with $u \in V^{k-1}_{\sigma(k)} \backslash \{\sigma(k)\}$ and $v \in V \backslash V^{k-1}_{\sigma(k)}$. We will show that there exists $t \in \{1, 2, ..., k-1\}$ such that $\sigma(t) \in V^{k-1}_{\sigma(k)} \backslash \{\sigma(k)\}$, $u \in V^{t-1}_{\sigma(t)}$, $e_{\sigma(t)} \in P_{\langle V,\Gamma^{\sigma,k-1}\rangle}(\sigma(k), u)$ and hence, $w(e_{\sigma(t)}) \leq w(\{u, v\})$.

Obviously, there exists $\bar{t} \in \{1, 2, ..., k-1\}$ such that $\sigma(\bar{t}) = u$ and we know that $u \in V^{\bar{t}-1}_{\sigma(\bar{t})}$. Hence, if $e_{\sigma(\bar{t})} \in P_{\langle V,\Gamma^{\sigma,k-1}\rangle}(\sigma(k), u)$, then we are done. Assume that $e_{\sigma(\bar{t})} \notin P_{\langle V,\Gamma^{\sigma,k-1}\rangle}(\sigma(k), u)$.

We will show that there exists $\hat{t} \in \{\bar{t}+1, ..., k-1\}$ [2] such that $u = \sigma(\bar{t}) \in V^{\hat{t}-1}_{\sigma(\hat{t})}$ and $e_{\sigma(\hat{t})} \in P_{\langle V,\Gamma^{\sigma,k-1}\rangle}(\{\sigma(k), u\})$. Suppose not. Let's denote the set

$$\left\{j \in V^{k-1}_{\sigma(k)} | e_{\sigma(\bar{t})} \in P_{\langle V,\Gamma^{\sigma,k-1}\rangle}(\sigma(k), j)\right\} \text{ by } C_1.$$

We know that $e_{\sigma(\bar{t})} = \{u_{\sigma(\bar{t})}, v_{\sigma(\bar{t})}\}$ for some $u_{\sigma(\bar{t})} \in V^{\bar{t}-1}_{\sigma(\bar{t})}$, $v_{\sigma(\bar{t})} \in V \backslash V^{\bar{t}-1}_{\sigma(\bar{t})}$ and $e_{\sigma(\bar{t})} \notin P_{\langle V,\Gamma^{\sigma,k-1}\rangle}(\sigma(k), u)$. Then obviously $v_{\sigma(\bar{t})} \in C_1$ and hence, $C_1 \neq \emptyset$. Clearly, there exists $\hat{t}_1 \in \{\bar{t}+1, ..., k-1\}$ such that $\sigma(\hat{t}_1) \in C_1$ and $u \in V^{\hat{t}_1-1}_{\sigma(\hat{t}_1)}$. Hence, if $e_{\sigma(\hat{t}_1)} \in P_{\langle V,\Gamma^{\sigma,k-1}\rangle}(\sigma(k), u)$, then, we are done.

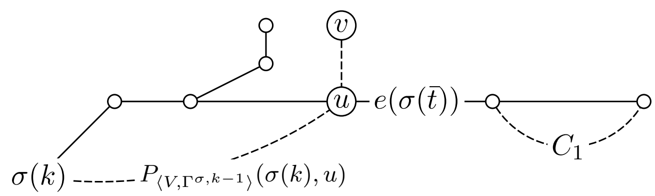

Figure 2: An auxiliary figure for the proof of Lemma 4.1

If $e_{\sigma(\hat{t}_1)} \notin P_{\langle V,\Gamma^{\sigma,k-1}\rangle}(\sigma(k), u)$, then we can show by using a similar argument as above that the set $C_2 = \left\{j \in V^{k-1}_{\sigma(k)} | e_{\sigma(\hat{t}_1)} \in P_{\langle V,\Gamma^{\sigma,k-1}\rangle}(\sigma(k), j)\right\} \neq \emptyset$ and there exists $\hat{t}_2 \in \{\hat{t}_1 + 1, ..., k-1\}$ such that $\sigma(\hat{t}_2) \in C_2$ and $u \in V^{\hat{t}_2-1}_{\sigma(\hat{t}_2)}$. Hence, if $e_{\sigma(\hat{t}_2)} \in P_{\langle V,\Gamma^{\sigma,k-1}\rangle}(\{\sigma(k), u\})$, then we are done. But, if $e_{\sigma(\hat{t}_2)} \notin P_{\langle V,\Gamma^{\sigma,k-1}\rangle}(\{\sigma(k), u\})$, then, since E_V is finite, by repeating the argument above finitely many times, one reaches a $\hat{t} \in \{\bar{t}+1, ..., k-1\}$ such that $e_{\sigma(\hat{t})} \in P_{\langle V,\Gamma^{\sigma,k-1}\rangle}(\sigma(k), u)$ and $u = \sigma(\bar{t}) \in V_{\sigma(\hat{t})}$. This proves that there exists $t \in \{1, ..., k-1\}$ such that $\sigma(t) \in V^{k-1}_{\sigma(k)} \backslash \{\sigma(k)\}$, $u \in V^{t-1}_{\sigma(t)}$, $e_{\sigma(t)} \in P_{\langle V,\Gamma^{\sigma,k-1}\rangle}(\sigma(k), u)$ and hence, $w(e_{\sigma(t)}) \leq w(\{u, v\})$.

We know that $e_{\sigma(t)} = \{u_{\sigma(t)}, v_{\sigma(t)}\}$ with $u_{\sigma(t)} \in V^{t-1}_{\sigma(t)}$ and $v_{\sigma(t)} \in V \backslash V^{t-1}_{\sigma(t)}$. Now we can use the whole argument given above to show that there exists $s \in \{1, 2, ..., k-1\}$ such that $\sigma(s) \in V^{k-1}_{\sigma(k)} \backslash \{\sigma(k)\}$, $v_{\sigma(t)} \in V^{s-1}_{\sigma(s)}$ and $e_{\sigma(s)} \in P_{\langle V,\Gamma^{\sigma,k-1}\rangle}(\sigma(k), u) \backslash \{\sigma(t)\}$. Hence, if $u_{\sigma(t)} \notin V^{s-1}_{\sigma(s)}$, then $w(e_{\sigma(s)}) \leq w(e_{\sigma(t)}) \leq w(\{u, v\})$ by the selection of $e_{\sigma(s)}$ by the V-algorithm. On

[2] Notice that $\bar{t} \leq k-2$ when $e_{\sigma(\bar{t})} \notin P_{\langle V,\Gamma^{\sigma,k-1}\rangle}(\sigma(k), u)$.

the other hand, if $u_{\sigma(t)} \in V^{s-1}_{\sigma(s)}$, then $u \in V^{s-1}_{\sigma(s)}$ and $v \notin V^{s-1}_{\sigma(s)}$. Hence $w(e_{\sigma(s)}) \leq w(\{u,v\})$ by the selection of $e_{\sigma(s)}$. But, repeating this argument at most $|E(P_{\langle V,\Gamma^{\sigma,k-1}\rangle}(\sigma(k),u))|$ times, we can conclude that $w(\{u,v\}) \geq w(e)$ for every $e \in P_{\langle V,\Gamma^{\sigma,k-1}\rangle}(\sigma(k),u)$. $\square$

In the following, we will denote $\langle V, F^{\pi,t}\rangle$ by its edge set $F^{\pi,t}$ for every mcst situation $\langle V, E_V, w\rangle$, $\pi \in \Pi(E_V)$ and $t \in \{1,2,...,|E_V|\}$. Moreover, for every $\sigma \in \Pi(N)$, we say that $i \in N$ is the last agent in $C_i(F^{\pi,t})$ with respect to σ, if $\sigma^{-1}(i) \geq \sigma^{-1}(j)$ for every $j \in V(C_i(F^{\pi,t}))$. Lastly, both the edge set and the vertex set of $C_i(F^{\pi,t})$ are denoted by $C_i(F^{\pi,t})$ unless there is danger of confusion.

Lemma 4.2 *Let $\langle V, E_V, w\rangle$ be a mcst situation. Let $\pi \in \Pi(E_V)$ be such that $w \in K^\pi$ and $t \in \{1,...,|E_V|-1\}$ be such that $w(\pi(t)) < w(\pi(t+1))$. Let $\sigma \in \Pi(N)$. Then, for every $k \in \{1,2,...,n\}$*

1. *$e_{\sigma(k)} \in C_{\sigma(k)}(F^{\pi,t})$ and hence, $w(e_{\sigma(k)}) \leq w(\pi(t))$ if $\sigma(k)$ is not the last agent in $C_{\sigma(k)}(F^{\pi,t})$ with respect to σ or if $0 \in C_{\sigma(k)}(F^{\pi,t})$.*
2. *$V^{k-1}_{\sigma(k)} \supset C_{\sigma(k)}(F^{\pi,t})$; $e_{\sigma(k)} \notin C_{\sigma(k)}(F^{\pi,t})$ and hence, $w(e_{\sigma(k)}) > w(\pi(t))$ if $\sigma(k)$ is the last agent in $C_{\sigma(k)}(F^{\pi,t})$ with respect to σ and $0 \notin C_{\sigma(k)}(F^{\pi,t})$.*

Proof. Pick a mcst situation $\langle V, E_V, w\rangle$ and $\pi \in \Pi(E_V)$ such that $w \in K^\pi$. Assume that there exists $t \in \{1,...,|E_V|-1\}$ such that $w(\pi(t)) < w(\pi(t+1))$. Pick $\sigma \in \Pi(N)$. Assume that the following induction hypothesis holds for all $k \in \{1,2,...,m-1\}(m \in \{2,...,n\})$.

- If $\sigma(k)$ is not the last player of $C_{\sigma(k)}(F^{\pi,t})$ with respect to σ, then $e_{\sigma(k)} \in C_{\sigma(k)}(F^{\pi,t})$.
- If $\sigma(k)$ is the last player of $C_{\sigma(k)}(F^{\pi,t})$ with respect to σ and $0 \in C_{\sigma(k)}(F^{\pi,t})$, then $e_{\sigma(k)} \in C_{\sigma(k)}(F^{\pi,t})$.
- If $\sigma(k)$ is the last player of $C_{\sigma(k)}(F^{\pi,t})$ with respect to σ, and $0 \notin C_{\sigma(k)}(F^{\pi,t})$, then $e_{\sigma(k)} \notin C_{\sigma(k)}(F^{\pi,t})$.

Let's prove the basis step ($k=1$). We know that $V^0_{\sigma(1)} = \{\sigma(1)\}$. Also, $w(\{\sigma(1),v\}) > w(\pi(t))$ for every $\{\sigma(1),v\} \notin C_{\sigma(1)}(F^{\pi,t})$ and $w(\{\sigma(1),v\}) \leq w(\pi(t))$ for every $\{\sigma(1),v\} \in C_{\sigma(1)}(F^{\pi,t})$. Assume first that $\sigma(1)$ is not the last player of $C_{\sigma(1)}(F^{\pi,t})$ with respect to σ. Then, $E(C_{\sigma(1)}(F^{\pi,t})) \neq \emptyset$ and obviously $e_{\sigma(1)} \in C_{\sigma(1)}(F^{\pi,t})$. Assume now that $\sigma(1)$ is the last player of $C_{\sigma(1)}(F^{\pi,t})$ with respect to σ and $0 \in C_{\sigma(1)}(F^{\pi,t})$. Then, obviously, $E(C_{\sigma(1)}(F^{\pi,t})) = \{\{0,\sigma(1)\}\}$ and hence, $e_{\sigma(1)} = \{0,\sigma(1)\}$. Assume lastly that $\sigma(1)$ is the last player of $C_{\sigma(1)}(F^{\pi,t})$ with respect to σ and $0 \notin C_{\sigma(1)}(F^{\pi,t})$. Then, obviously $E(C_{\sigma(1)}(F^{\pi,t})) = \emptyset$. Hence $e_{\sigma(1)} \notin E(C_{\sigma(1)}(F^{\pi,t}))$ and $w(e_{\sigma(1)}) > w(\pi(t))$.

We will now show that the induction hypothesis is true for k=m. Firstly, observe that:

(i) For every $\{u,v\} \in E_V$ such that $u \in C_{\sigma(m)}(F^{\pi,t})$ and $v \in V \backslash C_{\sigma(m)}(F^{\pi,t})$, $w(\{u,v\}) > w(\pi(t))$, because $\{u,v\} \notin F^{\pi,t}$.

(ii) If $V^{m-1}_{\sigma(m)} \backslash C_{\sigma(m)}(F^{\pi,t}) \neq \emptyset$, then $w(\{u,v\}) > w(\pi(t))$ for every $\{u,v\} \in E_V$ with $u \in V^{m-1}_{\sigma(m)} \backslash C_{\sigma(m)}(F^{\pi,t})$ and $v \in V \backslash V^{m-1}_{\sigma(m)}$, because there exists $\{u',v'\} \in P_{\langle V,\Gamma^{\sigma,m-1}\rangle}(\sigma(m),u)$ such that $u' \in C_{\sigma(m)}(F^{\pi,t})$ and $v' \notin C_{\sigma(m)}(F^{\pi,t})$. And by (i) $w(\{u',v'\}) > w(\pi(t))$. Then, by Lemma 4.1, $w(\{u,v\}) \geq w(\{u',v'\}) > w(\pi(t))$.

Now, (i) in conjunction with (ii) implies that for every $\{u,v\} \in E_V$ with $u \in V^{m-1}_{\sigma(m)}$ and $v \in V \backslash V^{m-1}_{\sigma(m)}$, $w(\{u,v\}) > w(\pi(t))$ if either $u \notin C_{\sigma(m)}(F^{\pi,t})$ or $v \notin C_{\sigma(m)}(F^{\pi,t})$.

Consider the graph $\langle V, \Gamma^{\sigma,m-1}\rangle$. Since $\langle V, \Gamma^{\sigma,m-1}\rangle$ does not contain any cycles, result (1) of Theorem 2.1 implies that the restriction of $\langle V, \Gamma^{\sigma,m-1}\rangle$ to $V(C_{\sigma(m)}(F^{\pi,t}))$ is connected if and only if it has $|V(C_{\sigma(m)}(F^{\pi,t}))|-1$ edges. Moreover, we know by the induction hypothesis that $e_{\sigma(k)} \in C_{\sigma(k)}(F^{\pi,t})$ for every $k \in \{1,...,m-1\}$. Hence, if $\sigma(m)$ is not the last player of $C_{\sigma(m)}(F^{\pi,t})$ with respect to σ or $0 \in C_{\sigma(m)}(F^{\pi,t})$, then the restriction of $\langle V, \Gamma^{\sigma,m-1}\rangle$ to $V(C_{\sigma(m)}(F^{\pi,t}))$ has less than $|V(C_{\sigma(m)}(F^{\pi,t}))|-1$ edges, and hence, it fails to be connected. Thus, both $V^{m-1}_{\sigma(m)} \cap C_{\sigma(m)}(F^{\pi,t}) \neq \emptyset$ and $(V \backslash V^{m-1}_{\sigma(m)}) \cap C_{\sigma(m)}(F^{\pi,t})) \neq \emptyset$. Moreover, since $C_{\sigma(m)}(F^{\pi,t})$ is connected in $\langle V, F^{\pi,t}\rangle$, there exists $\{u,v\} \in E(C_{\sigma(m)}(F^{\pi,t}))$ such that $u \in V^{m-1}_{\sigma(m)} \cap C_{\sigma(m)}(F^{\pi,t})$ and $v \in (V \backslash V^{m-1}_{\sigma(m)}) \cap C_{\sigma(m)}(F^{\pi,t}))$. Therefore, $e_{\sigma(m)} \in C_{\sigma(m)}(F^{\pi,t})$ and hence, $w(e_{\sigma(m)}) \leq w(\pi(t))$ if $\sigma(m)$ is not the last player of $C_{\sigma(m)}(F^{\pi,t})$ with respect to σ or $0 \in C_{\sigma(m)}(F^{\pi,t})$. This proves part (1) of Lemma 4.2.

If $\sigma(m)$ is the last player of $C_{\sigma(m)}(F^{\pi,t})$ with respect to σ and $0 \notin C_{\sigma(m)}(F^{\pi,t})$, then the restriction of $\langle V, \Gamma^{\sigma,m-1}\rangle$ to $V(C_{\sigma(m)}(F^{\pi,t}))$ has $|V(C_{\sigma(m)}(F^{\pi,t}))|-1$ edges, and hence, it is connected. But, then $V^{m-1}_{\sigma(m)} \supset C_{\sigma(m)}(F^{\pi,t})$. Hence $e_{\sigma(m)} \notin C_{\sigma(m)}(F^{\pi,t})$ and $w(e_{\sigma(m)}) > w(\pi(t))$. This proves part (2) of Lemma 4.2. □

We are now ready to prove the equivalence of the P-value and the V-value.

Theorem 4.1 *$v(w) = P(w)$ for every mcst situation $w \in \mathcal{W}^V$.*

Proof. First recall that the allocation of the cost of a single edge by the P-value may change with respect to the order of the edges under consideration. But, the allocation of the cost of the edges with same cost is the same regardless of the order considered. Hence, we will show below that the allocation of the cost of the edges with the same cost by the P-value and by the V-value are equal to each other.

Pick an mcst situation $\langle V, E_V, w\rangle$ and $a \in \cup_{e \in E_V}\{w(e)\}$.
Let $E_a = \{e \in E_V | w(e) = a\}$. Assume that $|E_a| = m$ for some $m \in \{1,2,...,|E_V|\}$ and $|\{w \in E_V | w(e) < a\}| = t$ for some $t \in \{0,1,...,|E_V|-m\}$. In the following we say that the V-value assigns the cost of $\left(\frac{k}{n!}a\right)$ to agent i for the construction of the edges in E_a, if i chooses to construct an edge from E_a at k ($k \in \{0,1,...,n!\}$) of the $n!$ orders on the set of players during the V-algorithm.

Firstly, for all $\pi \in \Pi(E_V)$ such that $w \in K^\pi$, the P^π-value (and hence the P-value) allocates the cost of (the edges in) E_a as

$$\sum_{k=1}^{m} w^\pi_{t+k} M^\pi e^{t+k} = a(b^{\pi,t} - b^{\pi,t+m}), \tag{6}$$

where the equality is implied by equation (3) of Lemma 2.1. Notice that $b^{\pi,t} = b^{\pi',t}$ ($b^{\pi,t+m} = b^{\pi',t+m}$), since $\langle V, F^{\pi,t}\rangle = \langle V, F^{\pi',t}\rangle$ ($\langle V, F^{\pi,t+m}\rangle = \langle V, F^{\pi',t+m}\rangle$) for every $\pi, \pi' \in \Pi(E_V)$ such that $w \in K^{\pi}$ and $w \in K^{\pi'}$. Hence, for every $\pi \in \Pi(E_V)$ with $w \in K^{\pi}$, we will denote $F^{\pi,t}$ as F^t, $F^{\pi,t+m}$ as F^{t+m}, $b^{\pi,t}$ as b^t and $b^{\pi,t+m}$ as b^{t+m}. Moreover, $\langle V, F^t\rangle$ and $\langle V, F^{t+m}\rangle$ will simply be denoted with their sets of edges, F^t and F^{t+m}, respectively.

Pick an agent $i \in N$. We will show by considering several cases that the allocation of the cost of the edges in E_a to i is done in the same way by the P-value and by the V-value.

Firstly, if $0 \in C_i(F^t)$, then clearly both $b_i^t = b_i^{t+m} = 0$. Also part (1) of Lemma 4.2 implies that $v_i^\sigma < a$ for every $\sigma \in \Pi(N)$. Then, both the V-value and the P-value assign 0 to i for the cost of construction of the edges in E_a. Hence, in the following, we will assume that $0 \notin C_i(F^t)$.

Case 1: $C_i(F^{t+m}) = C_i(F^t)$. Then, clearly, $n_i(F^t) = n_i(F^{t+m})$ and hence, $b_i^t - b_i^{t+m} = 0$. So, by (6) the P-value does not allocate any cost to i for the construction of edges in E_a. Let's now consider the V-value.

We know by part (1) of Lemma 4.2 that $v_i^\sigma < a$ for every $\sigma \in \Pi(N)$ such that i is not the last player of $C_i(F^t)$ with respect to σ. We will now show that $v_i^\sigma \neq a$ for every $\sigma \in \Pi(N)$ such that i is the last player of $C_i(F^t)$ with respect to σ, too. Suppose on the contrary that there exists $\sigma \in \Pi(N)$ such that i is the last player of $C_i(F^t)$ with respect to σ and $v_i^\sigma = a$, i.e., there exists $\{u, v\} \in E_a$ such that $u \in V_i^{\sigma^{-1}(i)-1}$, $v \in V \backslash V_i^{\sigma^{-1}(i)-1}$ and $w(\{u, v\}) \leq w(\{u', v'\})$ for every $\{u', v'\} \in E_V$ with $u' \in V_i^{\sigma^{-1}(i)-1}$ and $v' \in V \backslash V_i^{\sigma^{-1}(i)-1}$. Since $C_i(F^{t+m}) = C_i(F^t)$, there are two possibilities regarding the edge $\{u, v\}$. Either both $u, v \in C_i(F^t)$ or both $u, v \notin C_i(F^t)$. Assume first that both $u, v \in C_i(F^t)$. We know by part (2) of Lemma 4.2 that $V_i^{\sigma^{-1}(i)-1} \supset C_i(F^t)$. Hence, both $u, v \in C_i(F^t) \subset V_i^{\sigma^{-1}(i)-1}$ contradicting that $v \in V \backslash V_i^{\sigma^{-1}(i)-1}$. Now, assume that both $u, v \notin C_i(F^t)$. Then, Lemma 4.1 implies that $w(\{u, v\}) \geq w(e)$ for every $e \in P_{\langle V, \Gamma^{\sigma,\sigma^{-1}(i)-1}\rangle}(i, u)$. But since $u \notin C_i(F^t) = C_i(F^{t+m})$, there exists $e \in P_{\langle V, \Gamma^{\sigma,\sigma^{-1}(i)-1}\rangle}(i, u)$ such that $e \notin \langle V, F^{t+m}\rangle$. Then $w(e) > a = w(\{u, v\})$, a contradiction. Thus, $v_i^\sigma > a$ for every $\sigma \in \Pi(N)$ such that i is the last player of $C_i(F^t)$ with respect to σ. Hence, we can conclude that the V-value does not allocate any cost to i for the construction of edges in E_a, too.

Case 2: $C_i(F^{t+m}) \neq C_i(F^t)$. Then, there exists $i = i_1, i_2, ..., i_k$ $(2 \leq k \leq m + 1)$ such that $i_s \in N$ for every $s \in \{1, ..., k\}$, $C_{i_s}(F^t) \neq C_{i_r}(F^t)$ for every $r, s \in \{1, ..., k\}$ with $r \neq s$ and $\cup_{s=1}^k C_{i_s}(F^t) = C_i(F^{t+m})$.

Case 2.1: $0 \in C_i(F^{t+m})$. Then $b_i^t = \frac{1}{|V(C_i(F^t))|}$; $b_i^{t+m} = 0$ and hence, $b_i^t - b_i^{t+m} = \frac{1}{|V(C_i(F^t))|}$. Then by (6), i pays $\frac{a}{|V(C_i(F^t))|}$ for the construction of the edges in E_a with respect to the P-value. On the other hand, we know by part (1) of Lemma 4.2 that $v_i^\sigma < a$ for every $\sigma \in \Pi(N)$ such that i is not the last player of $C_i(F^t)$ with respect to σ. Moreover, again by part (1) of Lemma 4.2, we know that $v_j^\sigma \leq a$ for every $j \in C_i(F^{t+m})$ and $\sigma \in \Pi(N)$. Then, $v_i^\sigma = a$ for every $\sigma \in \Pi(N)$ such that i is the last player of $C_i(F^t)$ with respect to σ. Since, i is the last player of $C_i(F^t)$ with respect to σ for $\frac{1}{|V(C_i(F^t))|}n!$ orders on the set of players, i pays $\frac{a}{|V(C_i(F^t))|}$ for the construction of the edges in E_a with respect to the V-value.

Case 2.2: $0 \notin C_i(F^{t+m})$. Then by (6) P-value allocates to i

$$\begin{aligned} a(b_i^t - b_i^{t+m}) &= a\left(\frac{1}{|V(C_i(F^t))|} - \frac{1}{|V(C_i(F^{t+m}))|}\right) \\ &= a\left(\frac{|V(C_i(F^{t+m}))| - |V(C_i(F^t))|}{|V(C_i(F^{t+m}))||V(C_i(F^t))|}\right). \end{aligned} \tag{7}$$

On the other hand, we know by part (1) of Lemma 4.2 that $v_i^\sigma < a$ for every $\sigma \in \Pi(N)$ such that i is not the last player of $C_i(F^t)$ with respect to σ. Moreover, we know by part (2) of Lemma 4.2 that $v_i^\sigma > a$ when i is the last player of $C_i(F^{t+m})$ with respect to $\sigma \in \Pi(N)$. Then $v_i^\sigma = a$ when i is the last player of $C_i(F^t)$ but a player from another component is the last player of $C_i(F^{t+m})$. There are $\left(\frac{|V(C_i(F^{t+m}))|-|V(C_i(F^t))|}{|V(C_i(F^{t+m}))|}n!\right)$ orders such that a player from another component is the last player of $C_i(F^{t+m})$. In $\frac{1}{|V(C_i(F^t))|}$ of these orders i is the last player of $C_i(F^t)$. Then, the V-value assigns to i

$$\begin{aligned} &= a\left(\frac{1}{|V(C_i(F^t))|}\right)\left(\frac{|V(C_i(F^{t+m}))| - |V(C_i(F^t))|}{|V(C_i(F^{t+m}))|}\right) \\ &= a\left(\frac{|V(C_i(F^{t+m}))| - |V(C_i(F^t))|}{|V(C_i(F^{t+m}))||V(C_i(F^t))|}\right) \end{aligned} \tag{8}$$

which is equivalent to (7).

Lastly, observe that both $a \in \cup_{e \in E_V}\{w(e)\}$ and $i \in N$ are random, hence, we can conclude that $v(w) = P(w)$. □

Remark 1: Norde et al. (2004) introduce the P^τ-values for mcst situations for every ordering τ of the players. P^τ-values are also construct and charge rules which rely on the Kruskal algorithm and it is shown in Tijs et al. (2006) that the average of the P^τ-values over the set of all orderings of players is equal to the P-value. Actually, the cost allocation v^σ is equal to $P^{\bar{\sigma}}$, where $\bar{\sigma}$ stands for the reverse ordering of σ. This can be shown by constructing a proof which is similar to the proof of Theorem 4.1. Hence, we preferred a direct proof of the coincidence of the P-value and the V-value.

Remark 2: Bergantiños and Vidal-Puga (2007) associate an optimistic transferable utility game with mcst problems where the worth of a coalition is defined as the cost of connection, assuming that the rest of the agents are already connected to the source. They show that the Shapley value of this game is equal to the P-value. One can show easily that, for every ordering σ of the players, the v^σ value is equal to the marginal of the game associated by Bergantiños and Vidal-Puga (2007) to mcst problems for the same ordering. Hence, the coincidence of the V-value and the P-value can be proved by making use of the result regarding the coincidence of the Shapley value and the P-value in the game associated by Bergantiños and Vidal-Puga (2007). However, we believe that our effort for establishing a graph theoretical proof for the coincidence of the P-value and the V-value is worthwhile, because the V-algorithm can be extended easily to generalizations of the mcst problems and one can easily extend our graph theoretical proof for such situations. For example, in a companion study, we consider the extensions of the V-algorithm for mcst problems involving multiple sources and show, by extending the proofs in this paper, that the V-value coincides with the extended P-value for the mcst situations considered.

5 Conclusions

The literature on mcst problems have provided many construct and charge rules. These rules rely on the two well-known algorithms: the Kruskal algorithm and the Prim algorithm. The P-value is one such construct and charge rule which relies on the Kruskal algorithm. It has been shown that the P-value satisfies many desirable properties including core selectivity, equal treatment of equals, (strong) cost monotonicity and population monotonicity. For an overview of the interesting properties of the P-value, we refer to Bergantiños and Vidal-Puga (2005). In this study, we have shown that this important rule can be achieved by following a vertex oriented procedure which also determines a mcst in a new way. That is, we have shown that the P-value can be obtained as an average of the players' own choices in the vertex oriented algorithm.

Finally, we want to remark that Bergantiños and Vidal-Puga (2005) define a construct and charge procedure which is similar to the vertex oriented construct and charge procedure: The procedure they define is also dependent on the orderings of the players and in the procedure, each player pays the cost of the edge she preferred to construct. But, contrary to the vertex oriented construct and charge procedure, it only works on mcst problems with irreducible cost matrices, i.e., their procedure may not be efficient if the cost matrix of the mcst situation under investigation is not irreducible. If the procedure is applied to the associated irreducible cost matrix of a mcst situation, then the average of the cost allocations obtained over the set of orderings of the players is equal to the P-value.

References

[1] Bergantiños, G., and Vidal-Puga, J., J., (2005). Several approaches to the same rule in cost spanning tree problems. Mimeo.

[2] Bergantiños, G., and Vidal-Puga, J., J., (2007). The optimistic TU game in minimum cost spanning tree problems. *International Journal of Game Theory*, 36: 223-239.

[3] Bird, C., G., (1976). On cost allocation for a spanning tree: a game theoretic approach. *Networks*, 6: 335-350.

[4] Branzei, R., Moretti, S., Norde, H., Tijs, S., (2004). The P-value for cost sharing in minimum cost spanning tree situations. *Theory and Decision*, 56: 47-61.

[5] Claus, A., and Kleitman, D., J., (1973). Cost allocation for a spanning tree. *Networks*, 3: 289-304.

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

[7] Feltkamp, V., Tijs, S., Muto, S., (1994a). Minimum Cost Spanning Extension Problems : The Proportional Rule and the Decentralized Rule. Discussion Paper 96, Tilburg University, Center for Economic Research, The Netherlands.

[8] Feltkamp, V., Tijs, S., Muto, S., (1994b). On the Irreducible Core and the Equal Remaining Obligations Rule of Minimum Cost Spanning Extension Problems. Discussion Paper 106, Tilburg University, Center for Economic Research, The Netherlands.

[9] Gondran, M., and Minoux, M., (1984). Graphs and Algorithms. John Wiley and Sons, New York.

[10] Graham, R., L., and Hell, P., (1985). On the history of minimum spanning tree problem. *Annals of the History of Computing*, 7: 43-57.

[11] Granot, D., and Huberman, G., (1981). Minimum cost spanning tree games. *Mathematical Programming*, 21: 1-18.

[12] Granot, D., and Huberman, G., (1984). On the core and nucleolus of the minimum cost spanning tree games. *Mathematical Programming*, 29: 323-347.

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

[14] Kruskal, J., B., Jr., (1956). On the shortest spanning subtree of a graph and the Travelling Salesman Problem. *Proceedings of the American Mathematical Society*, 7: 48-50.

[15] Moretti, S., Branzei, R., Norde, H., Tijs, S., (2005). Cost Monotonic Construct and Charge Rules for Connection Situations. Discussion Paper 104, Tilburg University, Center for Economic Research, The Netherlands.

[16] Norde, H., Moretti, S., Tijs, S., (2004). Minimum cost spanning tree games and population monotonic allocation schemes. *European Journal of Operational Research*, 154: 84-97.

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

[18] Tijs, S., Branzei, R., Moretti, S., Norde, H., (2006). Obligation Rules for minimum cost spanning tree situations and their monotonicity properties. *European Journal of Operational Research*, 175: 121-134.