doi: 10.20532/cit.2016.1002842

Monotonic Cost Allocation Rule in Steiner Tree Network Games

Darko Skorin-Kapov

Robert B. Willumstad School of Business, Adelphi University, Garden City, New York, USA

We investigate the cost allocation strategy associated with the problem of providing some network service from source to a number of users, via the *Minimum Cost Steiner Tree Network* that spans the source and all the receivers. The cost of such a Steiner tree network, is distributed among its receivers. The objective of this paper is to develop a reasonably fair and computationally efficient cost allocation rule associated with the above cost allocation problem. Since finding the optimal Steiner tree is an *NP*-hard problem, the input to our cost allocation problem is the best known solution obtained using some heuristic. In order to allocate the cost of this Steiner tree to the users (receiver nodes), we formulate the associated *Steiner Tree Network (STN)* game in characteristic function form. It is well known that the core of the general *STN* game might be empty. We propose a new cost allocation rule for the modified *STN* game which might be attractive to network users due to its monotonic properties, associated with network growth.

ACM CCS (2012) Classification: Networks ⟶ Network Algorithms ⟶ Network Economics;
Mathematics of computing ⟶ Discrete mathematics ⟶ Combinatorics ⟶ Combinatorial optimization;
Theory of computation ⟶ Theory and algorithms for application domains ⟶ Algorithmic game theory and mechanism design ⟶ Algorithmic game theory;
Applied computing ⟶ Operations research ⟶ Decision analysis ⟶ Multi-criterion optimization and decision-making

Keywords: communication network, cost allocation, cooperative games, mathematical programming, Steiner trees

1. Introduction

Consider a connected directed network. Assume that a common source provides a service, which is required by users residing at some network nodes, and assume that any node receiving the service can in turn deliver it to its adjacent nodes. Each user is required to be connected, perhaps through other (switching) nodes, to the common source. With each edge (link) we associate the cost of using that edge to provide service. The set of users should be linked to the common source at a minimum cost. We refer to this problem as to the *Minimum Cost Steiner Tree Network* problem. This is an extensively studied problem in computer science and operations research which appears in many applications. Some applications include distribution networks, e.g. multicast routing (broadcasting financial information for example), water supply, electricity, internet services, cablevision, design of VSLI, and design of optical and wireless communication networks.

It is well known that finding an optimal Steiner tree in networks is a computationally prohibitive problem, and numerous researchers and practitioners have proposed various exact and heuristic approaches to solve it. It appears that in some practical application the rapid network changes requiring quick computation makes search for the "optimal solution" less attractive and the quicker heuristic is often used instead. (For a set of related articles and surveys see for example, [1,2] and [3]). The example of such a quick heuristic is a minimum cost spanning tree based heuristic which simply finds a minimum cost spanning tree spanning all nodes (receivers and Steiner nodes) and then prunes (eliminates) all Steiner nodes that are leafs. Herein, we will use this heuristic whenever the network is expanded by the addition of some new nodes.

In this paper, we are concerned with network cost allocation. The cost of a service network is shared by users who possibly have conflicting objectives. They might cooperate in order to decrease their joint cost if they are charged a “fair share” of the cost. There is no general answer to the fairness issue for network cost allocation problems. Cooperative game theory has been used to analyze several classes of network problems in the literature. An often used approach is to formulate the associated cost allocation problem as a cooperative game in characteristic function form, followed by the evaluation of various game theoretic solution concepts in the context of a particular problem. The examples related to work herein include: spanning tree games [4], Steiner tree games [5] and [6], network flow games [7], cost allocation arising from routing in networks [8], capacitated network design games [9] and [10], and hub network games [11] and [12]. For a survey and references on combinatorial cost allocation models in networks see for example [13,14].

Central to the cooperative game theory is the cost allocation solution known as the *core of a cooperative game*. The core consists of stable cost allocation solutions which provide no incentive for any coalition of users to secede and build their own subnetwork.

Let us first informally describe the related *Steiner Tree Network (STN)* game, as previously considered in the literature. Given a complete network *G*, the value of the characteristic function for each subset of user nodes is defined as the cost of the optimal (or best known) Steiner tree in *G* which spans all the users of that subset. It turns out that for the general case of the *STN* game when not all nodes are receivers, stable cost allocations do not necessarily exist (see, for example, [15]). Some sufficient conditions for their existence, as well as a heuristic algorithm for finding core points of the *STN* game, were presented in [5]). Some special cases were also studied in [6]). For approximation results on the *STN* problem, and their application to associated cooperative games, see [16] and [17], respectively. For a survey on related work see [18].

Researchers have extensively studied a computationally less prohibitive special case of the *STN* game in which users (receivers) reside at all nodes. The associated cooperative games are referred to as the class of *Minimum Cost Spanning Tree (MCST)* games. The non-emptiness of the core of the *MCST* game was demonstrated and analyzed in the literature (see, for example, [19,4,20,21], and [22,23,24,25]). Note however, that the entire core of the *MCST* game has not yet been characterized [26].

Furthermore, several authors studied cost and population monotonicity in the context of the *Minimum Cost Spanning Tree* games. Informally, in those games the cost allocation scheme is considered cost monotonic if the increase (resp. decrease) of the cost of any edge would not decrease (resp. increase) cost to any user. On the other hand, the cost allocation scheme is considered population monotonic if the addition of a new user node would not increase the cost to any of already existing users. For population monotonicity and/or cost monotonicity see for example, [27,28,29] and [30].

Since the *Steiner tree problem* is computationally complex and since the *Steiner Tree Network (STN)* game might have an empty core, it seems reasonable to look for the solutions that have other attractive properties and/or to modify the game in order to make it computationally tractable. In particular, we will look at the monotonicity properties in the *Minimum Cost Spanning Tree (MCST)* game and in some sense try to extend the considerations from the *MCST* game to the *STN* game. The main objective of this paper is to develop a cost allocation rule for the *STN* game or its modification which has potential practical applications in communication networks (for example, in multicasting, see [31]). The input to our cost allocation considerations is the optimal (or the best known) Steiner tree network spanning the source and all the receiver nodes. We then define the *Modified Steiner Tree Network (MSTN)* game in characteristic function form. The value of the characteristic function for each coalition is the cost of the optimal *Steiner Tree Network (STN)* spanning all members of that coalition in the complete graph generated by all nodes (receivers and switching nodes) of the *STN*. This definition is a modification of the characteristic function of the *STN* game. The profound difference between the two definitions is that, in the general *STN* game, each coalition is also allowed to use switching points which were not even used by the grand coalition. The rationale for such restriction and its practical implications are discussed in [6].

[23] developed a cost allocation scheme for the *MCST* game which is cost and population

monotonic and thus creates points in the irreducible core (see also, [4] and [26]).

Note that concepts of cost and population monotonic cost allocation scheme for the *MCST* game can not be naturally extended to the Steiner tree case due to a complex network structure created with the inclusion of switching nodes. Specifically, the addition of another node (receiver or switching node) might result in the elimination of some other switching nodes from the optimal or the best known Steiner tree network. Nevertheless, we extend some monotonicity considerations of minimum cost spanning tree games to a more general Steiner tree case, by developing a new cost allocation rule on Steiner trees that satisfies certain network monotonicity. We propose a cost allocation rule which gives no incentive to any users to block network growth. Namely, if an additional Steiner node is used in the network and/or if an additional user joins the network, the cost will not be increased to any of already existing network users. The informal outline of the rule follows.

The initial input to our considerations is the optimal (or the best known) Steiner tree in the given network. If some Steiner nodes (switching nodes) are not used in the best known solution, they are eliminated from the network. In order to determine/estimate the value of the inclusion of switching nodes in the network, we use our known population and cost monotonic cost allocation scheme for minimum spanning trees as if all remaining Steiner nodes were also users. Then, the cost allocated to Steiner nodes is further distributed among actual user/receiver nodes in such a way that each user pays a fraction of its benefit generated by the inclusion of Steiner nodes.

Further, when an additional node (or a set of nodes) is added to the network we find the minimum cost spanning tree in a new network. If any leaf node or a whole subtree with the exception of its root consisted of non-user nodes, we reduce the cost of edges of such subtrees to zero. With this step we actually performed the minimum cost spanning tree heuristic to find the new Steiner tree. Then, we apply again the above population and cost monotonic cost allocation scheme for spanning trees as if all nodes were users. Finally, we re-distribute the cost of non-user nodes among user nodes in such a way that each user pays at most a fraction of its benefit generated by the inclusion of new Steiner nodes. At this point we eliminate non-used (leaf) Steiner nodes. With this rule the entire cost of the new *STN* is allocated to user nodes and neither of those users pay more than they paid before the network expansion. Namely, this cost allocation rule is monotonic with respect to network growth.

Hence, the main contributions of this work are: (1) the formulation of the population and Steiner nodes monotonicity in the context of the *STN* cost allocation problem and (2) the development of a computationally tractable cost allocation rule for generating some monotonic cost allocation solutions for the above *MSTN* game.

The remainder of the paper is organized as follows. In Section 2 we review some standard definitions and introduce notation; in Section 3 we present some known preliminary results from the literature. In Section 4 we discuss the concept of monotonicity in the context of the *STN* tree game and present the *Steiner Tree Network Cost Allocation (STNCA)* algorithm. It is demonstrated that the *STNCA* algorithm generates monotonic cost allocations for the *MSTN game*. In Section 5, we illustrate the *STNCA* by example. Finally, in Section 6, we summarize our findings and contributions.

2. Basic Definitions

The *Minimum Cost Steiner Tree* problem for directed graphs can be formulated as follows. Given a directed weighted graph $G' = (D \cup S' \cup \{O\}, E)$ with an edge weight (cost) function $w : E \to R^+$, the origin O a set D of demand nodes (users/receivers) and a set S' of potential switch nodes (un-inhabited/Steiner nodes), find a minimum cost directed Steiner tree $ST = (N_{ST}, E_{ST})$ in G, rooted away from node O whose node set contains $D, i.e.N_{ST} = D \cup S \cup \{O\}$ and$S \subseteq'$. Throughout this paper we will assume that the weight function for a graph G under consideration is symmetric, $w(i,j) = w(j,i)$, for all edges $(i,j) \in E$. For such a symmetric case, solving the directed *Minimum Cost Steiner Tree* problem is equivalent to finding a minimum cost Steiner tree in the equivalent undirected weighted graph. It is well-known that this problem is *NP*-hard [32].

The special case of the above problem in which all nodes are receivers ($S' = \emptyset$) is the widely studied and computationally easier *Minimum Cost Spanning Tree Problem.*

For our cost allocation considerations we will use the concept of a cooperative game in the characteristic function form. Let D be a set of players and let us define the characteristic function c by $c : 2^{|D|} \to R^+$, such that $c(\emptyset) = 0$ and for each $Q \subseteq D$, $c(Q)$ is the characteristic function value. We can interpret $c(Q)$ as the cost of providing service to a set of users in Q. Then, the pair (D, c) is a cooperative game.

For $x \in R^{|D|}$ and $Q \subseteq D$, let $x(Q) \equiv \sum_{j \in Q} x_j$ be the part of the total cost paid by coalition Q. A *cost allocation* vector x in game (D, c) satisfies $x(D) = c(D)$, and the theory of cooperative games is concerned with the selection of a reasonable subset of cost allocation vectors.

Central to the theory of cooperative games is the solution concept referred to as the core of a game. The core $C(D, c)$ of a game (D, c) consists of all cost allocation vectors $x \in R^{|D|}(x(D) = c(D))$, such that $x(Q) \leq c(Q)$ for all $Q \subseteq D$. Observe that the core consists of all allocation vectors x which provide no incentive for any coalition to secede and create their own subnetwork.

The vector $x = (x_{i,J})_{i \in J, J \subseteq D}$ is a *population monotonic* cost allocation scheme [32] of the game (D, c) if and only if it satisfies the following conditions:

$$\text{For each } J \subseteq D, \sum_{i \in J} x_{i,J} = c(J) \text{ and} \tag{1}$$

$$\begin{aligned} &\text{For each } J, K \subseteq D, \text{ and} \\ &\quad i \in J, J \subseteq K \Rightarrow x_{i,J} \geq x_{i,K}. \end{aligned} \tag{2}$$

Namely, if a new user joins the network and the cost allocation scheme is applied, the cost will not increase to any previously existing user. None of the network users have incentive to block the network growth.

The cost allocation scheme is said to be *cost monotonic* if in the case of the decrease of any single edge weight, no player cost will increase. Equivalently, if any single edge weight is increased, no player cost will decrease.

3. Preliminaries

Using the notation, from Section 2 input to our cost allocation problem is the best known Steiner tree $ST = (D \cup S \cup \{O\}, E_{ST})$ in a given complete weighted graph $G' = (D \cup S' \cup \{O\}, E)$. We define the characteristic function c_G as follows: $c_G : 2^{|D|} \to R^+$ is such that $c_G(\emptyset) = 0$ and for each $Q \subseteq D$, $c_G(Q)$ is the cost of the minimum Steiner tree spanning all nodes of Q in a complete graph $G = G(D \cup S \cup \{O\})$ generated by the nodes of the best known Steiner tree ST. We refer to the game (D, c_G) as to the *Modified Steiner Tree Network (MSTN)* game. Note that in the general *Steiner Tree Network (STN)* game, the characteristic function is defined on the complete graph $G' = G'(D \cup S')$ induced by the set of nodes which includes all users and all potential Steiner nodes. In the *MSTN* game each coalition agrees to restrict their considerations to the complete graph generated by nodes of the best-known solution of the grand coalition. Clearly, the value of the characteristic function for a particular coalition, as defined in the general *STN* game, might seem more rational, since it allows each coalition to consider the cheapest solution in the entire network. However, the restriction implied by the above modification reduces some technical difficulties while still giving us a reasonably good cost allocation model. These trade-offs were discussed in [6].

It is well-known that the *Minimum Cost Steiner Tree* problem in a directed weighted graph $G' = (D \cup S' \cup \{O\}, E)$ can be formulated as an integer programming problem [5]. For a directed edge $l = (i, j)$ we refer to i as the tail and j as the head of l, and for a subset of nodes $U \subseteq D \cup'$, we denote by $\delta(U)$ the set of all directed edges having their heads, but not their tails, in U. A subset $U \subseteq D \cup S'$, is said to be a cut-set of G, if $U \cap D \neq \emptyset$ and the undirected subgraph $G(U)$ of G induced by U is connected. We denote by U_D the set of all cut-sets of G. Then the *Min Cost Steiner Tree* problem can be formulated as:

$$\begin{aligned} IP(D) : \min\Big\{ &wu : u(\delta(U)) \geq 1, \\ &U \in U_D, u \in \{0, 1\}^E \Big\}, \end{aligned}$$

where $u \equiv u(i, j) = 1$ if $(i, j) \in E$ is used in the directed Steiner tree ST, and 0 otherwise.

Hence, the minimum cost of the directed Steiner tree $ST = (D \cup S \cup \{O\}, E)$ is the best objective function value of the integer programming problem $IP(D)$. Denote by $IP(Q)$, for $Q \subseteq D$, the directed ST problem in the complete network $G = G(D \cup S \cup \{O\})$ induced by nodes of ST obtained from the original problem by replacing D with Q. Then, the pair (D, c_G), where $c_G : 2^{|D|} \to R^+$ is such that $c_G(\emptyset) = 0$ and for each $Q \subseteq D$, $c_G(Q)$ is the minimum objective function value of $IP(Q)$, is the *Modified Steiner Tree Network (MSTN)* game.

Consider now the linear programming relaxation $LP(D)$ of $IP(D)$ (the minimum cost directed Steiner tree problem as defined above), $LP(D) : min\{wu : u(\delta(U)) \geq 1, U \in U_D, u \geq 0\}$, and its dual $DP(D)$,

$$DP(D) : \max\Big\{ \sum_{U \in U_D} v_U :, \sum_{U} \in U_D : \\ e \in \delta(U) v_U \leq w(e), \text{ for all } e \in E; \\ v_U \geq 0 \text{ for all } U \in U_D \Big\}.$$

It turns out that the dual $DP(D)$ can be used to allocate a fraction of the total cost of the Steiner tree ST, while satisfying the core constraints of the *MSTN* game. Namely, if $v_U, U \in U_D$, is a feasible solution to $DP(D)$, and if for each $U \in U_D$ the amount v_U is allocated arbitrarily to users in U, then the cost allocation vector x generated by this operation satisfies the core constraints. Indeed, for any subset of users $Q \subseteq D, c_G(Q) = \sum_e \in E_{ST_Q} w(e)$, where E_{ST_Q} is the set of edges of the directed Steiner tree $ST_Q = (N_{ST_Q} \cup \{O\}, E_{ST_Q})$ that spans $Q \cup \{O\}$ in the complete network induced by nodes of the best known Steiner tree ST. Then, by the construction of x and by dual feasibility:

$$x(Q) \leq \sum_{U \in U_D : U \cap Q \neq \emptyset} v_U \leq \sum_{e} \in E_{ST_Q} w(e) \\ = c_G(Q).$$

Remark: The above implies that for a special case when $S' = \emptyset$, i.e. for a spanning tree game, packing algorithm [34] could be used to generate some core points of the spanning tree game.

[23] used this to develop a cost allocation scheme for the *MCST game* which generates some core points. Note, that the *MSTCAS* also identifies the minimum spanning tree. For completeness, we include herein the slightly modified outline of their *Minimum Spanning Tree Cost Allocation Scheme (MSTCAS).*

***MSTCAS* Algorithm:**

```
Input:  A weighted complete network
          $G = (D \cup \{O\}, E, W(E))$.
Initialization:  Set the cost allocations $x_p$=0, for
                 all $p \in D$, and all dual variables
                 $v_U = 0$ for all cut-sets $U \in U_D$.
Main Step:
"Find a potential allocating set $U$ (the smallest in
size)".
    Do for $k = 1, \ldots |D|$,
        Do for each $p \in D$
            Let $U = \{p\}$.
            Do until $|U| = k$ or no additional nodes
            can be added to $U$
                If $\exists$ $(i,j) \in \delta(U)$ such that $w(i,j)$=0,
                $i \neq O$; let $U = U \cup \{i\}$
            EndDo
"Allocate to users in $U$ as much as possible, divide
equally"
            If for all $e \in \delta(U), w(e) \geq 0$
                Let $y_U = min\{w(e), e \in \delta(U)\}$
                Do for all $p \in D$
                    Let $x_p = x_p + \frac{y_U}{|U|}$
                EndDo
                Do for all $e \in \delta(U)$
                    Let $w(e) = w(e) - y_U$
                EndDo
            EndIf
        EndDo
    EndDo
End
```

[23] also showed that the above *MSTCAS* is cost and population monotonic. [26] have shown that a cost allocation scheme which is cost and population monotonic produces points that coincide with the irreducible core of the *MCST* game.

4. Steiner Tree Network Cost Allocation (*STNCA*) Algorithm

In this section, we first discuss the steps of the *STNCA* algorithm. Then we summarize the *STNCA* algorithm and discuss its efficiency.

In case when all network nodes except source $\{O\}$ are inhabited by users, it is known that the *MSTCAS* from Section 3 constitutes a population monotonic scheme. Consequently, adding the additional set of users to the network and applying *MSTCAS* would not increase the cost to any already existing user and the new cost allocation would be in the core of the new game.

Let us describe the above more formally. Consider the game (D, c_G) on $G = G(D \cup \{O\}, E)$. Using *MSTCAS* find the core cost allocation $x_D(D) = \sum_{i\in D} x_{i,D}$.

The addition of a set of users U to the network G would create a graph $G_1(D \cup U \cup \{O\}, E)$. Applying *MSTCAS* to the game $(D \cup U, c_{G_1})$ would for each $j \in D \cup U$ generate the core cost allocation $x_{j,D\cup U}$. Due to the population monotonicity of the *MSTCAS*, for each $i \in D$, $x_{i,D\cup U} \le x_{i,D}$.

Let us now go back to the original game (D, c_G) on $G(D\cup\{O\}, E(D\cup\{O\}))$ and consider adding a non-empty set S of public (un-inhabited) nodes to the network, thus creating an initial Steiner tree network $G_1 = (D\cup S\cup O, E)$. It seems natural to assume that the coalition D would agree on adding nodes S (all of them) only if that addition is beneficial to the grand coalition D. Consequently, we assume that the best Steiner tree in G_1 spans all the nodes in $D \cup S$. Observe that in the initial step we find the best (perhaps optimal) Steiner tree using the best tool available. In subsequent steps, when the network grows and changes rapidly, we generate new Steiner tree networks using more practical minimum cost spanning tree heuristic.

Consider the minimum cost spanning tree game $(D \cup S, c'_{G_1})$ on G_1 and apply to it the *MSTCAS* scheme as if all nodes in $D \cup S$ were populated with users. The result is the cost allocation $x'_{j,D\cup S}$, for each $j \in D \cup S$. The idea behind this is to determine a reasonable cost for the inclusion of each public node.

The result of this cost allocation

$$x'_{D\cup S}(D) + x'_{D\cup S}(S) = c'_{G_1}(D \cup S) < c_G(D) = x_D(D).$$

Note that we can assume the strict inequality, since otherwise the set of Steiner nodes would not be included into the network.

This implies that the total benefit of the inclusion of Steiner nodes in the network to users in D exceeds the cost assigned to switching nodes i.e.,

$$0 \le x'_{D\cup S}(S) < x_D(D) - x'_{D\cup S}(D) = \sum_{j\in D}(x_{j,D} - x'_{j,D\cup S,}).$$

Note also that since *MSTCAS* is population monotonic, the individual benefit of the inclusion of S in the network is $(x_{j,D} - x'_{j,D\cup S}) \ge 0$, for all $j \in D$. We can assume that $(x_{j,D} - x'_{j,D\cup S}) > 0$, for some $j \in D$ since otherwise we would not include the entire set S in the network.

Now consider the *Steiner Tree Network (STN)* game (D, c_{G_1}) on $G_1 = (D \cup S \cup O, E)$.

Next we distribute the entire network cost to actual users. The cost assigned to switching nodes is determined above using our population and cost monotonic cost allocation scheme for spanning trees as $x'_{D\cup S}(S)$. The addition of the set of switching nodes S is overall beneficial to users in D in the amount of $x_D(D) - x'_{D\cup S}(D)$, where each user i benefits $x_{i,D} - x'_{i,D\cup S}$. It seems reasonable to distribute the cost assigned to switching nodes in such a way that each user pays a corresponding fraction of overall benefit created by the addition of S. Namely, for each $i \in D$, we allocate the cost as follows:

$$x_{i,D\cup S} = x'_{i,D\cup S} + \frac{x_{i,D} - x'_{i,D\cup S}}{\sum_{j\in D}(x_{j,D} - x'_{j,D\cup S})} x'_{D\cup S}(S) \le x_{i,D},$$

(note that the inequality holds since

$$\frac{x'_{D\cup S}(S)}{\sum_{j\in D}(x_{j,D} - x'_{j,D\cup S})} \le 1).$$

Moreover, the above implies that $\sum_{i\in D} x_{i,D\cup S} = x'_{D\cup S}(D) + x'_{D\cup S}(S) = c_{G_1}(D)$ meaning that the entire cost was allocated and that no node had incentive to block the addition of public nodes in S to the network.

4.1. Monotonicity of *STN* game with respect to addition of Steiner nodes

Next we define the cost allocation rule in case new Steiner nodes are added to the network. When we add a new Steiner node or a set of Steiner nodes to the existing *STN*, we will find the new minimum cost *STN* using the quick minimum cost spanning tree based heuristic. We will actually use the *MSTCAS* applied to the network as if all nodes were users to find that tree. Note that this might eventually lead to the elimination of some non-used Steiner nodes from the *STN*. Clearly, that somewhat restricts the generality of our cost allocation rule. Namely, in the case of the network growth, we assume herein the use of a quick heuristic and moreover each subsequent cost allocation will depend on the current one. Consequently, the *STN* under consideration is not necessarily optimal and the allocation is not independent of the sequence in which the network is formed. Nevertheless, our rule still addresses the likely practical situation in which the quick decision on the addition of potential new nodes should be made from the point of view of the current solution.

Let us now add a set of public nodes T to network G_1 and create a network $G_2 = (D \cup S \cup T \cup \{O\}, E)$. We find the minimum cost spanning tree in G_2 using the *MSTCAS*. If for any subtree of that minimum cost spanning all non-root nodes are Steiner nodes, we reduce the costs of edges of such a subtree to zero. The cost of this spanning tree is now equal to the cost of the new *STN* since the new minimum cost *STN* will be obtained by the elimination of leaf Steiner nodes. If the cost of the new *STN* is not better than the original one, we would stop the expansion. Otherwise we repeat the *MSTCAS* as if all nodes were users.

Now for each $i \in D$, the cost allocation rule follows:

$$\begin{aligned} x_{i,D\cup S\cup T} = {} & x'_{i,D\cup S\cup T} \\ & + \frac{x_{i,D\cup S} - x'_{i,D\cup S\cup T}}{x_{D\cup S}(D) - x'_{D\cup S\cup T}(D)} x'_{D\cup S\cup T}(S\cup T). \end{aligned}$$

Since the addition of T improves the cost of *STN* and since *MSTCAS* is population and cost monotonic scheme, $0 < \dfrac{x'_{D\cup S\cup T}(S \cup T)}{x_{D\cup S}(D) - x'_{D\cup S\cup T}(D)} \leq 1$, then we have $x_{i,D\cup S\cup T} \leq x'_{i,D\cup S\cup T} + x_{i,D\cup S} - x'_{i,D\cup S\cup T}$, which implies that $x_{i,D\cup S\cup T} \leq x_{i,D\cup S}$, for all $i \in D$ and the entire cost is allocated, i.e. $c_{G_2}(D) = x_{D\cup S\cup T}(D)$.

At this point we would also remove all non-used (Steiner leafs) public nodes from the network G_2.

Hence, our cost allocation rule is monotonic with respect to the addition of public nodes.

4.2. Monotonicity of *STN* game with respect to addition of users

Let us now turn to the network expansion created by the addition of new users. As already mentioned, if all nodes are users, the *MSTCAS* is population monotonic.

Consider now again the *Modified Steiner Tree Network* game (D, c_{G_1}) on $G_1 = (D \cup S \cup O, E)$, and the cost for each $i \in D$ given as follows:

$$\begin{aligned} x_{i,D\cup S} &= x'_{i,D\cup S} + \frac{x_{i,D} - x'_{i,D\cup S}}{\sum_{j\in D}(x_{j,D} - x'_{j,D\cup S})} x'_{D\cup S}(S) \\ &\leq x_{i,D}, \end{aligned}$$

Suppose now that a set of users U is joining the network, thus creating a network $G_3 = (D \cup U \cup S, E)$. Consider the game $(D \cup U \cup S, c')$ and apply to it the *MSTCAS* scheme as if all nodes in S were also populated with users.

Note that once again for practical purposes we work as if the entire set of un-inhabited nodes S is included in the best solution. Actually, we find the minimum spanning tree which spans $D \cup U \cup S$ and allocate its cost. If all leafs of this minimum spanning tree are receivers this tree is our best *STN*. Otherwise, we further improve the *STN* by reducing the weights of all subtrees that do not contain receivers (except for the root) down to zero. Cost monotonicity of our minimum cost spanning tree cost allocation scheme assures that this only further reduces the cost to all nodes. If $x'_{D\cup U\cup S}(S) = 0$ we are done, otherwise the cost allocation is now defined as:

$$\begin{aligned} x_{i,D\cup U\cup S} = {} & x'_{i,D\cup U\cup S} \\ & + \frac{x_{i,D\cup S} - x'_{i,D\cup U\cup S}}{\sum_{j\in D}(x_{j,D\cup S} - x'_{j,D\cup U\cup S})} x'_{D\cup U\cup S}(S), \end{aligned}$$

for each $i \in D$ and $x_{u,D\cup U\cup S} = x'_{u,D\cup U\cup S}$, for $u \in U$.

By construction, and population and cost monotonicity of the *MSTCAS*,

$$x_{D\cup S}(D) = x'_{D\cup S}(D) + x'_{D\cup S}(S) \geq x'_{D\cup U\cup S}(D) + x'_{D\cup U\cup S}(S),$$

which together with $x'_{D\cup U\cup S}(S) > 0$ implies that $1 \geq \frac{x'_{D\cup U\cup S}(S)}{\sum_{j\in D}(x_{j,D\cup S} - x'_{j,D\cup U\cup S})} > 0$ and together with $x_{j,D\cup S} - x'_{j,D\cup U\cup S} \geq 0$ for all j, in turn implies that:

$x_{i,D\cup U\cup S} \leq x_{i,D\cup S}$, for all $i \in D$ and $x_{D\cup U\cup S}(D\cup U) = c_{G_3}(D\cup U)$.

Hence, our cost allocation rule is monotonic with respect to the addition of new users. Namely, there is no incentive for any user to block the addition of new users to the network.

Next we present an outline of the *STNCA* algorithm.

Remark (The complexity of the *STNCA* Algorithm)

The input network has $|N+1|$ nodes and $|N|**2 + |N|$ edges. The algorithm runs the *MSTCAS*, identifies Steiner leafs and reduces appropriate links, reapplies the *MSTCAS* and finally does a simple cost allocation computational operation for each node. It was reported by [23] that the *MSTCAS* can be implemented so that for any given k, the algorithm examines at most n links for each of n nodes. Since k takes on the values

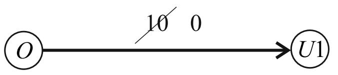
```
STNCA Algorithm
Input:  A Steiner tree $ST = (D \cup S \cup \{O\}, E_{ST})$, with a set of users $D$ and non-leaf
          Steiner tree nodes $S$ in a weighted complete network
        $G = (D \cup S \cup \{O\})$ and the cost for each $i \in D$ given as follows:
        $x_{i,D\cup S} = x'_{i,D\cup S} + \frac{x_{i,D} - x'_{i,D\cup S}}{\sum_{j\in D}(x_{j,D} - x'_{j,D\cup S})} x'_{D\cup S}(S).$
Main Step:  If a set of Steiner nodes $T$ is added to the network we consider a complete
              network $G(N) = G(D \cup S \cup T \cup \{O\})$.
                Perform The $MSTCAS$ on $G(N)$.
                The output is: $x'_{i,D\cup S\cup T}$ for all $i \in D \cup S \cup T$ and the minimum
                cost spanning tree $MCST$ in $G$.
                    If any leaf subtree of $MCST$ does not contain users,
                    reduce its edge weights to zero.
                        If the total cost of $MCST$ is not smaller than the
                        cost of $ST$ stop,
                        otherwise repeat the $MSTCAS$ on $G(N)$.
                        EndIf
                For each $i \in D$,
                $x_{i,D\cup S\cup T} = x'_{i,D\cup S\cup T} + \frac{x_{i,D\cup S} - x'_{i,D\cup S\cup T}}{x_{D\cup S}(D) - x'_{D\cup S\cup T}(D)} x'_{D\cup S\cup T}(S \cup T).$
                Eliminate Steiner leaf nodes (leaf subtrees) from the network.
            EndIf.
            If a set of users $U$ is added to the network we consider a complete network
            $G(N) = G(D \cup U \cup S \cup \{O\})$.
                Perform The $MSTCAS$ on $G(N)$.
                The output is: $x'_{i,D\cup U\cup S}$ for all $i \in D \cup U \cup S$ and the minimum
                cost spanning tree $MCST$ in $G$.
                    If any leaf subtree of $MCST$ does not contain users, reduce
                    its edge weights to zero and repeat $MSTCAS$ on $G(N)$.
                    EndIf
                For $i \in D$, $x_{i,D\cup U\cup S} = x'_{i,D\cup U\cup S} + \frac{x_{i,D\cup S} - x'_{i,D\cup U\cup S}}{\sum_{j\in D}(x_{j,D\cup S} - x'_{j,D\cup U\cup S})} x'_{D\cup U\cup S}(S),$
                and for $u \in U$, $x_{u,D\cup U\cup S} = x'_{u,D\cup U\cup S}$.
                Eliminate Steiner leaf nodes (leaf subtrees) from the network.
            EndIf.
```

1 through $|N|$, the running time of the *MSTCAS* is $O(|N|^3)$. The identification of Steiner leafs involves the inspection of at most $|N| + 1$ links for $|N|$ nodes. Hence, the overall complexity of the *STNCA* algorithm is $O(|N|^3)$.

5. Illustration of *STNCA* Algorithm

We start the illustration with a trivial single user network presented in Figure 1 and then in Figures 2 to 6 we go through several network expansions. In Figures 3 to 6 we assume that networks are complete and that the edges not included have very high costs.

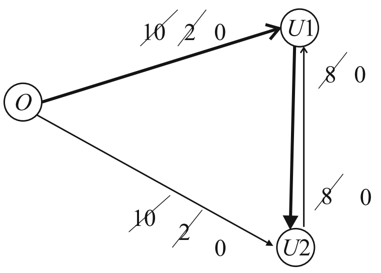
The initial network in Figure 1 has a single allocating set. The entire cost $c(D) = 10$ is paid by user U1.



$c(D)$=10
$x(U1)$=10

Figure 1. The initial network.

Then, user $U2$ joins the network (see, Figure 2). The *STNCA* algorithm results in the cost of spanning network $c(D) = 18$ (bold links) and the cost allocation vector $(x(U1), x(U2)) = (9, 9)$. The cost to user $U1$ was reduced and the cost allocation is in the core of the associated *STN* game.



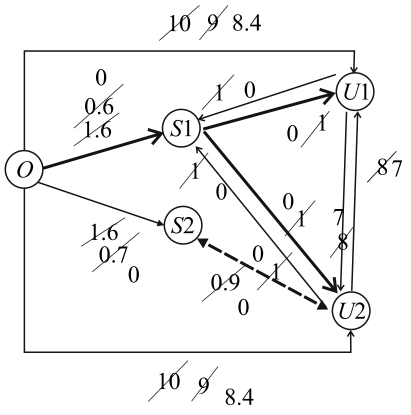
$c(D)$=18
$x(U1)$=8+1=9
$x(U2)$=8+1=9

Figure 2. Add {U2}.

We further expand the network by adding the set of Steiner nodes $\{S1, S2\}$ (see Figure 3). The new Steiner tree is found by the minimum spanning tree heuristic. Namely, the spanning tree is first found by *MSTCAS* and then the cost of link $(U2, S2)$ was reduced to zero. The cost of the Steiner tree is improved to $c(D) = 3.6$. Then we perform our *STNCA* rule and obtain the new cost allocation $(x(U1), x(U2)) = (1.82, 1.78)$. At the end of the process we remove $S2$ from the network.

By this operation, we achieved the cost reduction to both users. Moreover, in this example, the cost allocation is still the core allocation. Observe that although $S2$ is removed from the network, it played some role in the cost allocation. Consequently, the cost allocation is not symmetric. In some sense $U2$ got some credit for giving up the potential of using $S2$.

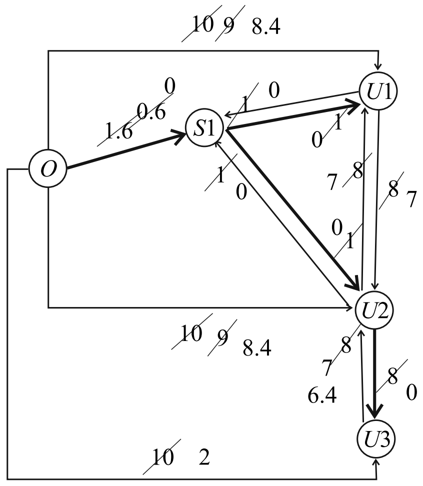
The addition of user $U3$ is presented in Figure 4. After applying the *SETNCA* algorithm, the new cost allocation $(x(U1), x(U2), x(U3)) = (1.82, 1.78, 8)$ is obtained. It is also a core allocation. Moreover, $U1$ and $U2$ did not incur any additional cost for this expansion. One could perhaps argue that $U3$ is entering the network at a relatively low price. $U3$ appears to be far away from the source and is only paying to link itself to $U2$. Nevertheless, $U1$ and $U2$ would



$$c(D) = 3.6$$
$$x'(U1) = 1 + 0.15 = 1.15$$
$$x'(U2) = 0.9 + 0.05 + 0.15 = 1.1$$
$$x'(S1) = 1 + 0.15 = 1.15$$
$$x'(S2) = 0.9 + 0.05 + 0.15 - 1 - 0.9 = 0.2$$
$$x(U1) = 1.15 + (9 - 1.15) \cdot 1.35/(18 - 2.2.25) = 1.82$$
$$x(U2) = 1.1 + (9 - 1.1) \cdot 1.35/(18 - 2.25) = 1.78$$

$S2$ is removed from the network.

Figure 3. Add $\{S1, S2\}$, remove $\{S2\}$.



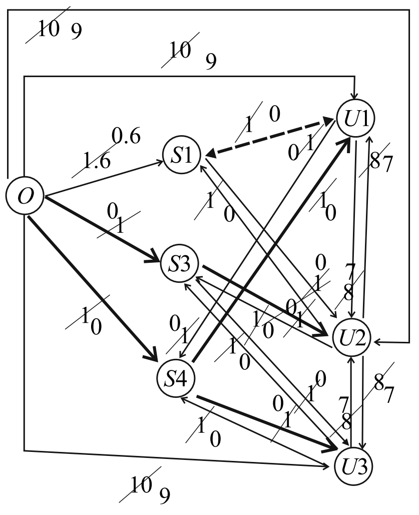
$$
\begin{aligned}
c(D) &= 11.6\\
x'(U1) &= 1 + 0.2 = 1.2\\
x'(U2) &= 1 + 0.2 = 1.2\\
x'(U3) &= 8\\
x'(S1) &= 1 + 0.2 = 1.2\\
x(U1) &= 1.2 + (1.82 - 1.2)\cdot 1.2/(3.6 - 2.4) = 1.82\\
x(U2) &= 1.2 + (1.78 - 1.2)\cdot 1.2/(3.6 - 2.4) = 1.78\\
x(U3) &= 8
\end{aligned}
$$

Figure 4. Add $\{U3\}$.

not block, since there is no cost to them. Moreover, it might be beneficial for them to have $U3$ in the network and potentially achieve better reductions associated with future network expansions.

In the next step we add to the network a set of Steiner nodes $\{S3, S4\}$ (see Figure 5). Note that for this network the core of the *STN* game is empty (as demonstrated in Figure 5).

The *MSTCAS* finds the spanning tree and allocates the cost as if the Steiner nodes were users. After finding that $S1$ is a leaf, we reduce the cost allocated to $S1$ by the weight of link $(U1, S1)$. (It is easy to verify that in this case this is equivalent to the operation in which we would reduce the cost of link $(U1, S1)$ to zero and then repeat the *MSTCAS*). This, tree based heuristic determined the minimum cost Steiner tree with the total cost $c(D) = 5$ (bold links). Here, we find that $S1$ is a leaf which should be eliminated. However, we do not remove it immediately. Since the cost of Steiner tree was reduced from the previous network from Figure 4, we apply *SETNCA* rule to allocate the cost of



$$
\begin{aligned}
c(D) &= 5,\ \text{note that the core is empty since}\\
x(U1) + x(U2) &\Leftarrow 3.6\\
x(U2) + x(U3) &\Leftarrow 3\\
x(U3) + x(U1) &\Leftarrow 3\\
\text{Hence,}&\\
x(U1) + x(U2) + x(U3) &\Leftarrow 4.8 < c(D)\\
x'(U1) &= 1\\
x'(U2) &= 1\\
x'(U3) &= 1\\
x'(S1) &= 1 - 1\\
x'(S3) &= 1\\
x'(S4) &= 1\\
x(U1) &= 1 + (1.82 - 1)\cdot 2/(11.6 - 3) = 1.191\\
x(U2) &= 1 + (1.78 - 1)\cdot 2/(11.6 - 3) = 1.181\\
x(U3) &= 1 + (8 - 1)\cdot 2/(11.6 - 3) = 2.628\\
&S1 \text{ is removed.}
\end{aligned}
$$

Figure 5. Add $\{S3, S4\}$, remove $\{S1\}$.

Steiner nodes (including $S1$) to user nodes and then eliminate $S1$ from the network.

The cost allocation is now $(x(U1), x(U2), x(U3)) = (1.191, 1.181, 2.628)$. Thus, the cost has been reduced to all users. Observe that the core of the *MSTN* game (when $S1$ is eliminated) is not empty. It can be verified that $(2, 2, 1)$ is in the core of the *MSTN* game. The cost allocation generated by our rule is not in the core of the *MSTN* game, since the core constraint $x(U3) \leq 2$ is violated.

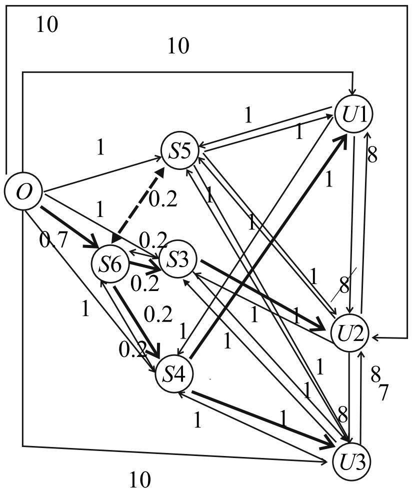
Finally we add a set of Steiner nodes $\{S5, S6\}$ (see Figure 6). We apply the minimum cost spanning tree heuristic to find the new Steiner tree (indicated in bold on Figure 6) with overal cost $c(D) = 4.1$. Consequently, we will eliminate $S5$ from the network. It can be verified that the core of the associated *MSTN* game (with $S5$ eliminated) is not empty and that it contains cost allocation $(1.3, 1.4, 1.4)$. Note however that in this case the spanning tree heuristic is performing poorly and we are actually not making the optimal decision. Namely, the optimal Steiner tree includes $\{S5, S6\}$ and excludes $\{S3, S4\}$ and has the optimal overall cost of 3.9. Note that for this network the cost allocation $(1.3, 1.3, 1.3)$ is in the core of the *STN* game.

It can be verified that the application of the *SETNCA* algorithm results in a new cost allocation $(x(U1), x(U2), x(U3)) = (1.11, 1.09, 1.9)$ in which all users again achieved a cost reduction. However, this cost is not optimal and the cost allocation is not in the core.

An interesting avenue for the future research should include the search for a population monotonic rule that would produce core allocations whenever they exist and/or be as close as possible to the core.

6. Conclusion

We analyzed the cost allocation associated with the *Steiner Tree Network (STN)* problem. In particular, we are interested in the cost allocation strategy associated with a potential network growth. It is well known that the optimization part of the problem is computationally hard (*NP*-hard) and that the game theoretic solution concepts are difficult to analyze and/or compute (for example core might be empty). Consequently, for practical situations when the network is quickly growing, we suggest the use of a simple heuristic for the optimization part and a modification/simplification of the game for the cost allocation analysis. Specifically, we use a minimum cost spanning tree heuristic to find the minimum cost *STN* which is the input to our cost allocation problem. We also modify the *STN* game. The idea of the modification is to restrict strategic considerations to the nodes used in the best known Steiner tree solution. Namely, the value of the characteristic function



$$c(D) = 0.7 + 0.2 + 0.2 + 1 + 1 + 1 = 4.1$$

Core is not empty, it contains:

$$(x(U1), x(U2), x(U3)) = (1.3, 1.4, 1.4)$$

$S5$ is removed / not used

$$x'(U1) = 1$$
$$x'((U2) = 1$$
$$x'(U3) = 1$$
$$x'(S3) = 0.2 + 0.166 = 0.366$$
$$x'(S4) = 0.2 + 0.166 = 0.366$$
$$x'(S6) = 0.2 + 0.166 = 0.366$$
$$x(U1) = 1 + (1.191 - 1) \cdot 1.1/(5 - 3) = 1.11$$
$$x(U2) = 1 + (1.181 - 1) \cdot 1.1/(5 - 3) = 1.09$$
$$x(U3) = 1 + (2.628 - 1) \cdot 1.1/(5 - 3) = 1.9$$

Figure 6. Add $\{S5, S6\}$, remove $\{S5\}$.

for a particular coalition of receiver nodes is defined as the cost of the best solution they could achieve in the complete network generated by nodes of the best known Steiner tree.

Under the above assumptions we develop a cost allocation rule which is, in some sense, the extension of our known population and cost monotonic cost allocation scheme on spanning trees to Steiner tree networks. The new rule has the following desirable monotonic property. If some new nodes (users and/or useful switching nodes) are added to the *STN*, or the cost of some links was reduced, our cost allocation rule assures that the cost is not increased to any previously existing users. Namely, no users would have an incentive to block the network growth in terms of the addition of new users and would have no incentive to block any improvement that could be achieved by the addition of new switching nodes, or by the reduction of cost of some links. Moreover the rule is simple and enables the quick computation of the relatively attractive cost allocation.

Acknowledgement

The author also gratefully acknowledge some constructive comments made by anonymous referees on an earlier version of this paper which led to a significantly improved exposition of the material.

References

[1] M. Bezensek and B. Robic, “A Survey of Parallel and Distributed Algorithms for the Steiner Tree Problem”, *International Journal of Parallel Programming*, vol. 42, pp. 287–319, 2014. http://dx.doi.org/10.1007/s10766-013-0243-z

[2] X. Zhen Cheng and D. Ding-Zhu (Eds.), *Steiner Trees in Industry Series: Combinatorial Optimization*, vol. 11., Springer, 2002. http://dx.doi.org/10.1007/978-1-4613-0255-1

[3] S. Voss, “Modern Heuristic Search Methods for the Steiner Problem in Graphs”, in *Advance in Steiner Trees*, D-Z Du *et al.*, Eds., Kluwer Academic Publishers, 2000, pp. 283-323.

[4] S. Bird, “On Cost Allocation for a Spanning Tree: A Game Theoretic Approach”, *Networks*, vol. 6, pp. 335–350, 1976. http://dx.doi.org/10.1002/net.3230060404

[5] D. Skorin-Kapov, “On the Core of the Minimum Steiner Tree Game in Networks”, *Annals of Operations Research*, vol. 57, pp. 233–249, 1995. http://dx.doi.org/10.1007/BF02099700

[6] D. Skorin-Kapov and J. Skorin-Kapov, “A Note on Steiner Tree Games”, *Networks*, vol. 59, no. 2, pp. 215–225, 2012. http://dx.doi.org/10.1002/net.20436

[7] D. Granot and F. Granot, “On Some Network Flow Games”, *Mathematics of Operations Research*, vol. 17, no. 4, 792–841, 1992. http://dx.doi.org/10.1287/moor.17.4.792

[8] J. A. M. Potters *et al.*, “Traveling Salesman Games”, *Mathematical Programming*, vol. 53 pp. 199–211, 1987. http://dx.doi.org/10.1007/BF01585702

[9] D. Skorin-Kapov, “On a Cost Allocation Problem Arising from a Capacitated Concentrator Covering Problem”, *Operations Research Letters*, vol. 13, no. 5, pp. 315–323, 1993. http://dx.doi.org/10.1016/0167-6377(93)90054-K

[10] D. Skorin-Kapov and H. F. Beltran, “An Efficient Characterization of Cost Allocation Solutions Associated with Capacitated Network Design Problems”, *Telecommunication Systems*, vol. 3, no. 1, pp. 91–107, 1994. http://dx.doi.org/10.1007/BF02110045

[11] D. Skorin-Kapov, “Hub Network Games”, *Networks*, vol. 31, pp. 293–302, 1998. http://dx.doi.org/10.1002/(SICI)1097-0037(199807)31:4<293:AID-NET9>3.0.CO;2-8

[12] D. Skorin-Kapov and J. Skorin-Kapov, “Threshold Based Discounting Network: The Cost Allocation Provided by the Nucleolus”, *European Journal of Operations Research*, vol. 166, no. 1, pp. 154–159, 2005. http://dx.doi.org/10.1016/j.ejor.2004.01.043

[13] N. Nisan *et al.*, Eds., *Algorithmic Game Theory*, Cambridge University Press, 2007. http://dx.doi.org/10.1017/CBO9780511800481

[14] W. W. Sharkey, “Networks Models in Economics”, in *Handbooks in OR&MS*, M.O. Ball *et al.*, Eds., vol. 8, Elsevier, 1995.

[15] A. Tamir, “On the Core of Network Synthesis Game”, *Mathematical Programming*, vol. 50, pp. 123–135, 1991. http://dx.doi.org/10.1007/BF01594930

[16] V. V. Vazirani, “Recent Results on Approximating the Steiner Tree Problem and its Generalizations”, *Theoretical Computer Science*, vol. 235, pp. 205–216, 2000. http://dx.doi.org/10.1016/S0304-3975(99)00192-9

[17] K. Jain and V. V. Vazirani, “Applications of Approximation Algorithms to Cooperative Games”, in *STOC’01*, Hersonissos, Crete, Greece, July 6–8, 2001.

[18] D. Skorin-Kapov, “On Cost Allocation in Steiner Tree Networks,” *Steiner Trees in Industry*, vol. 11, pp. 353–375, 2001. http://dx.doi.org/10.1007/978-1-4613-0255-1_11

[19] H. Aarts, "Minimum Cost Spanning Tree Games", Memorandum No. 1003, Ph.D. thesis, University of Twente, The Netherlands, 1994.

[20] V. Feltkamp *et al.*, "Bird's Tree Allocation Revisited", Center for Econometrics, Tilburg University, Technical Report, 1994.

[21] D. Granot and G. Huberman, "On the Core and Nucleolus of the M.C.S.T. Games", *Math. Programming*, vol. 29, pp. 323–347, 1984. http://dx.doi.org/10.1007/BF02592000

[22] D. Granot and G. Huberman, "Minimum Cost Spanning Tree Games", *Math. Programming*, vol. 21, pp. 1–18, 1981. http://dx.doi.org/10.1007/BF01584227

[23] K. Kent and D. Skorin-Kapov, "Population Monotonic Cost Allocation on MST's", in *Operational Research Proceedings KOI'96*, 1996, pp. 43–48.

[24] D. Skorin-Kapov, "On Some Cost Allocation Problems in Communication Networks", *Mathematical Communications*, vol. 3, pp. 85–94, 1998.

[25] N. Megiddo, "Computational Complexity and the Game Theory Approach to Cost Allocation for a Tree", *Mathematics of Operations Research*, vol. 3, pp. 189–196, 1978. http://dx.doi.org/10.1287/moor.3.3.189

[26] U. Faigle *et al.*, "On the Complexity of Testing Membership in the Core of Min-Cost Spanning Tree Games", *International Journal of Game Theory*, vol. 26, pp. 361–366, 1997. http://dx.doi.org/10.1007/BF01263277

[27] G. Bergantinos and J. Vidal-Puga, "Characterization of Monotonic Rules in Minimum Cost Spanning Trees", *International Journal of Game Theory*, pp. 1–34, 2014.

[28] L. Lorenzo and S. Lorenzo-Freire, "A characterization of Kruskal sharing rules for minimum cost spanning tree problems", *International Journal of Game Theory*, vol. 38, no. 1, pp. 107–126, 2009. http://dx.doi.org/10.1007/s00182-008-0147-0

[29] B. Dutta and A. Kar, "Cost Monotonicity, Consistency and Minimum Cost Spanning Tree Games", *Games and Economic Behavior*, vol. 48, pp. 223–248, 2004. http://dx.doi.org/10.1016/j.geb.2003.09.008

[30] H. Norde *et al.*, "Minimum Cost Spanning Tree Games and Population Monotonic Cost Allocation Schemes", *European Journal of Operational Research*, vol. 154, pp. 84–97, 2004. http://dx.doi.org/10.1016/S0377-2217(02)00714-2

[31] C.A.S. Oliveira *et al.*, "Optimization problems in multicast tree construction", in *Handbook of Optimization in Telecommunications*, Dordrecht: Kluwer, 2005, pp. 701–733.

[32] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP Completeness*, San Francisco: Freeman, 1979.

[33] Y. Sprumont, "Population Monotonic Ailocation Schemes for Cooperative Games with Transferable Utility", *Games and Economic Behavior*, vol. 2, 1990, pp. 378–394.

[34] D. R. Fulkerson, "Packing Rooted Directed Cuts in a Weighted Directed Graph", *Mathematical Programming*, vol. 6, pp. 1–13, 1974. http://dx.doi.org/10.1007/BF01580218

Received: November, 2015
Revised: February, 2016
Accepted: March, 2016

Contact address:
Darko Skorin-Kapov
Robert B. Willumstad School of Business
Adelphi University
Garden City
New York 11530
USA e-mail: skorin@adelphi.edu

DARKO SKORIN-KAPOV received his Ph.D. in Management Science from the University of British Columbia, Vancouver, Canada, and M.Sc. in Information Science and B.Sc. in Mathematics from Zagreb University, Croatia. He has extensive international academic and industrial experience. He is currently a Professor of Information Technology and Operations Management at the R.B. Willumstad School of Business, Adelphi University, New York, USA. Previously he worked at Financial Technologies International, New York; Stony Brook University, New York; Zagreb University, Croatia; Croatian Hydro-Meteorological Institute and Zagreb Bank, Croatia. His research interests are focusing on optimization and cost allocation in communication networks. He authored over 50 research papers published in journals in management science and related areas. He serves as an associate editor of several journals and is a recipient of a number of awards and grants.