Sharing the cost of a network

Koster, M.A.L.; Molina, E.; Sprumont, Y.; Tijs, S.H.

Published in:
International Journal of Game Theory

Publication date:
2001

Int J Game Theory (2001) 30:567–599

International Journal of Game Theory

Sharing the cost of a network: core and core allocations*

M. Koster[†], E. Molina[‡], Y. Sprumont[§], S. H. Tijs[¶]

[†] Faculty of Economics and Econometrics, Department of Quantitative Economics, University of Amsterdam, The Netherlands (E-mail: mkoster@fee.uva.nl)
[‡] Centro de Investigación Operativa, Universidad Miguel Hernández de Elche, Elche, Spain (E-mail: e.molina@umh.es)
[§] C.R.D.E. et Département de sciences économiques, Université de Montréal, Montréal, Canada (E-mail: sprumony@crde.umontreal.ca)
[¶] CentER and Department of Econometrics, Tilburg University, Tilburg, The Netherlands (E-mail: s.h.tijs@kub.nl)

Final version November 2001

Abstract. This paper discusses the core of the game corresponding to the standard fixed tree problem. We consider the weighted adaptation of the constrained egalitarian solution of Dutta and Ray (1989). The core of the standard fixed tree game equals the set of all weighted constrained egalitarian solutions. Each weighted constrained egalitarian solution is determined (in polynomial time) as a *home-down* allocation, which creates further insight in the local behaviour of the weighted constrained egalitarian solution. The constrained egalitarian solution is characterized in terms of a *cost sharing mechanism*.

JEL classification: C71

Key words: Cooperative game theory, cost sharing, tree games, core, constrained egalitarianism.

1. Sharing the cost of a tree network

In this paper the focus will be on the class of cooperative cost games that arises from *standard fixed tree enterprises* (cf. Granot *et al.* (1996)). A standard fixed tree enterprise is the mathematical model for the following situation. There is a fixed and finite set of agents who are connected via a fixed tree network to a special location that is called the *root*. We seek to allocate the cost of this tree for cases where the connections within the network are costly. Many real-life situations can be modelled to fit in this general setting. For instance, consider

* We thank the anonymous referee in charge for useful suggestions on the paper.

the problem of allocating the maintenance cost of an irrigation network or a cable vision network, setting airport taxes for planes or setting dredging fees for ships. In a natural way each standard fixed tree problem gives rise to a cooperative cost game, the *standard fixed tree game*, such that the agents are the players and the cost of a coalition of players is related to the minimal expenses for connecting all its members to the root. There is an extensive list of papers which use techniques from cooperative game theory to investigate essentially the same type of situations. In this respect we mention Bird (1976), Megiddo (1978), Granot and Huberman (1981), (1984), Granot *et al.* (1996), Granot and Maschler (1998), Bjørndal *et al.* (1999), and van Gellekom and Potters (1999). The special case when the underlying structure of the game is a chain, is also known as the airport problem and considered by several authors such as Littlechild (1974), Littlechild and Thompson (1977), Dubey (1982), Potters and Sudhölter (1999), Aadland and Kolpin (1998) and Bergantino and Coppejans (1997). Moulin and Shenker (1996), Young (1998) and Koster *et al.* (2000) discuss noncooperative models on structures with standard fixed trees as a special case.

It is a well known fact that the standard fixed tree games constitute a well-behaved class of cost games. A standard fixed tree game is known to be *concave*, which encompasses a uniform incentive for cooperation. Concave cost games are known to have a large core, i.e. the set that consists of all cost allocations that are stable with respect to separating coalitions (cf. Shapley (1971)). This paper studies the core of the standard fixed tree games and maintenance games in particular. In Section 3 we make use of the tree structure to characterize the core of maintenance games in several ways, and we investigate its geometrical structure. In a natural way, the results can be generalized to the class of standard fixed tree games. In Section 4 we show that the core of maintenance games consists of all weighted constrained egalitarian allocations (see, e.g., Koster (2002) and Ebert (1999)). These solutions are a weighted adaptation of the egalitarian concept by Dutta and Ray (1989). This result relates to Monderer *et al.* (1992) who show that each core element of a concave cost game is obtained as a weighted Shapley value for some weight system. Bjørndal *et al.* (1999) provides a constructive proof of this fact for standard fixed tree games.

Dutta and Ray (1989) propose an algorithm for computing the constrained egalitarian solution, which is easily generalized for computing weighted adaptations (Koster (2002)). Still the algorithms are of exponential complexity. However, in Section 4 it is shown that in case of maintenance games the weighted constrained egalitarian solutions can be calculated in polynomial time. In a way that is very natural, but particular to the case of standard fixed tree games, dual weighted Shapley values and weighted constrained egalitarian allocations are duals of each other: both types of allocations can be seen as the result of a dynamical process of *locally* distributing the costs of the arcs forming the tree. Where in Bjørndal (1999) it is shown that a weighted Shapley value is a *down-home* allocation in the sense that it is determined by splitting incremental costs from the root to the leafs of the tree, a weighted constrained egalitarian allocation is of *home-down* type, splitting the incremental costs from the leafs to the root (Section 4). Monotonicity properties for both above mentioned classes of solutions can be obtained easily from this dynamic approach.

In Section 5 we focus on fixed tree networks that are equal up to the cost function. A cost sharing mechanism is a device that relates each cost structure to a vector of cost shares. The egalitarian cost sharing mechanism relates each cost function to the constrained egalitarian solution for the corresponding maintenance game. The main theorem is that the egalitarian cost sharing mechanism is the unique cost sharing mechanism that is monotonically depending on the cost structure, while it specifies core elements in the corresponding maintenance game and simultaneously minimizes the range of cost shares. Analogously, within the bounds of the core and cost monotonicity, it uniquely minimizes the maximal cost share.

2. The standard fixed tree game

Granot *et al.* (1996) study the cost games arising from *fixed tree networks* $\Gamma = (V, E, b, c, N)$. Here (V, E) is a tree, i.e. a directed graph[1] without circuits with vertex set V and arc set E. One vertex, r, has a special meaning and is called the *root* of the tree. There is a cost function $c : E \to \mathbb{R}$ on the arcs of the tree, and for an arc e the number $c(e)$ is interpreted as the total of corresponding construction costs. Similarly, there is a cost function $b : V \to \mathbb{R}$ on the vertices of the tree. Notice that costs can be negative, with respect to arcs as well as vertices. N is the society of players, each of its members i is located at some vertex $v(i) \in V$. A vertex v is *occupied* if it specifies the location of one or more players. The coalition of players located at some vertex in $T \subseteq V$ is denoted N_T.

A fixed tree network $\Gamma = (V, E, b, c, N)$ is *standard* if the following additional properties hold:

(a) The cost function assigns nonnegative numbers to arcs.
(b) The costs on vertices are zero, i.e. $b(v) = 0$ for all $v \in V$.
(c) The root is not occupied.
(d) If v is a *leaf*, i.e. a vertex from where no other vertex can be reached, then v is occupied.
(e) If v is not occupied, then there are (at least) two vertices $v_1 \neq v_2$ such that $(v, v_1), (v, v_2) \in E$.
(f) There is exactly one vertex $v \in V$ such that $(r, v) \in E$.

For the sake of transparency we will leave out the cost function on the vertices, $b : V \to \mathbb{R}$, from the definition of a standard fixed tree network. In addition, we will frequently write G for the graph defined by (V, E). In this way $\Gamma = (G, c, N)$ stands for the standard fixed tree network (V, E, b, c, N) in the above terminology.

The objective of the players is to connect themselves through the network to the root. For instance, think of players being households that want to hook up the cable vision network, or water supply system. It is assumed that all arcs in the network are directed away from the root. Then in order to establish a connection with the root, a player needs to finance the *path* from the root to his vertex v, i.e. the subgraph $P_v = (\tilde{V}, \tilde{E})$ of (V, E) with $\tilde{V} = \{r, v_1, v_2, \dots, v_p, v\}$ and $\tilde{E} = \{(r, v_1), (v_1, v_2), \dots, (v_{p-1}, v_p), (v_p, v)\}$. Similarly, we say that a coali-

[1] Here we slightly deviate from the original setting of an undirected graph; nevertheless the interpretation and results do not change.

tion of players is connected to the root, if all the individual members are connected. It is assumed that the players may share the use of arcs and vertices, such that a coalition of players needs to pay only once for an arc or vertex. We will model the above situation using the notion of a cooperative cost game.

Definition. A *cooperative cost game* is an ordered pair (M, c) where M is a finite set of players and $c : \mathscr{P}(M) \to \mathbb{R}$ is the *characteristic function* that assigns a real number to each coalition of players S in the powerset of M, $\mathscr{P}(M)$. Moreover $c(\emptyset) = 0$. Then for each $S \subseteq M$, $c(S)$ is interpreted as the minimal cost of serving the members of S, independently from the players $M \backslash S$. The class of all cost games is denoted $\mathscr{G}$.

For fixed tree networks, by the *cost* of a coalition S, $c_\Gamma(S)$, we mean the minimal cost needed to join all members of S to the root. By putting $c_\Gamma(\emptyset) = 0$, the ordered pair (N, c_Γ) defines a cooperative cost game that we will refer to as the *fixed tree game* for Γ. If Γ is standard then we refer to (N, c_Γ) as the corresponding *standard fixed tree game*.

In order to establish the necessary connections with the root, a coalition S needs at least the union of all paths from the root to the vertices that the individual players occupy. This means, that if a player in S is located at v, then all vertices *preceding* v, i.e. the vertices on the path P_v, are indispensible. If $\tilde{v}$ is on the path P_v, then this is denoted $\tilde{v} \preccurlyeq v$. In this way $(V, \preccurlyeq)$ defines a *precedence relation* on V. By a *trunk* of $G = (V, E)$ is meant a set of vertices $T \subseteq V$ that is closed under $\preccurlyeq$, i.e. $v \in T$ and $\tilde{v} \preccurlyeq v$ imply $\tilde{v} \in T$. For $S \subseteq N$, T_S denotes the smallest trunk containing all vertices at which the players in S are located. The cost associated to a coalition S is easily restated in terms of the cost of arcs connecting the vertices in T_S. First we will introduce some additional notation. For $v \in V \backslash \{r\}$, let $\pi(v)$ be its *direct* predecessor, i.e. the unique vertex such that $(\pi(v), v) \in E$. The arc $(\pi(v), v)$ will be denoted by e_v. The set of vertices in V with predecessor v is denoted $\pi^{-1}(v)$.

Lemma 2.1. *For a standard fixed tree problem* $\Gamma = (G, c, N)$ *it holds*

$$c_\Gamma(S) = \sum_{v \in T_S} c(e_v) \quad \textit{for all } S \subseteq N. \tag{1}$$

Proof: Each of the players in S needs at least the path from the root to his location. In particular, S needs at least the union of all these paths. But then by nonnegativity of costs this union is the optimal way to connect S to the root. □

An alternative way to express the costs of a coalition is in terms of (*dual*) *unanimity games*. For each cooperative cost game (N, g) the dual cost game (N, g^*) is defined by putting $g^*(S) := g(N) - g(N \backslash S)$ for all $S \subseteq N$. Then $* : g \mapsto g^*$ defines a linear operator with the property that for all games (N, g) we have $(g^*)^* = g$. Formally, for $S \subseteq N \backslash \{\emptyset\}$ the *unanimity game* $(N, u_S) \in \mathscr{G}$ is defined by

$$u_S(T) = \begin{cases} 1 & \text{if } S \subseteq T, \\ 0 & \text{otherwise,} \end{cases}$$

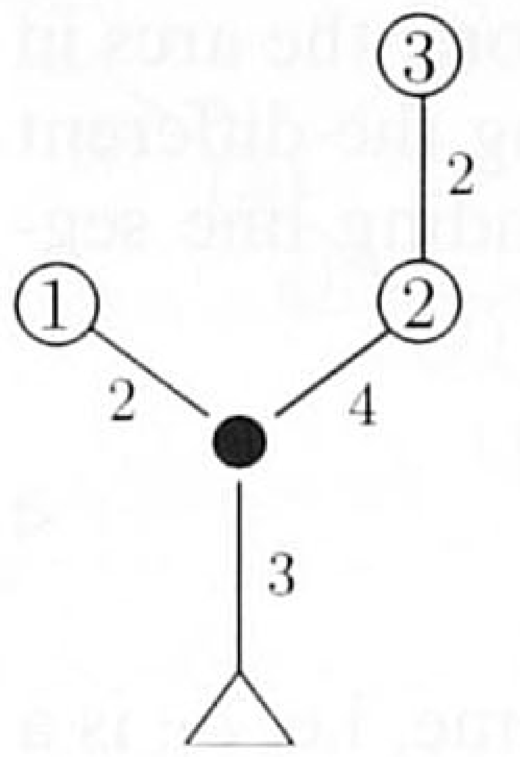

Fig. 1. A standard fixed tree problem.

for all $T \subseteq N$. Then the *dual unanimity game* for $S \subseteq N$ is given by (N, u_S^*), where

$$u_S^*(T) = \begin{cases} 1 & \text{if } T \cap S \neq \varnothing, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

for all $T \subseteq N$. The set of *dual unanimity games* forms a basis for $\mathscr{G}$, since for all cost games $(N, g) \in \mathscr{G}$ we have

$$g = \sum_{T \subseteq N \setminus \{\varnothing\}} \Delta_{g^*}(T) u_T^*, \tag{3}$$

where Δ_{g^*} is the *dividend* given by $\sum_{S:S \subseteq T} (-1)^{|T|-|S|} g^*(S)$, and $|S|$ denotes the cardinality of the player set $S \subseteq N$. The next proposition shows that a standard fixed tree game (N, c_Γ) is easily expressed using the basis of dual unanimity games.

A player $i \in N$ is considered as a *follower* of $v \in V$ if $v \preccurlyeq v(i)$, i.e. v is on the path from the root to player i's location. Then the set of all followers of v is denoted $F(v)$.[2] Then (N, c_Γ) is a linear combination of the dual unanimity games induced by the sets of followers for each of the vertices except the root.

Proposition 2.2. *Let $\Gamma = (G, c, N)$ be a standard fixed tree problem. Then the associated cost game (N, c_Γ) can be represented as*

$$c_\Gamma = \sum_{v \in V \setminus \{r\}} c(e_v) u^*_{N_{F(v)}}. \tag{4}$$

Proof: Let S be a nonempty coalition. It follows from Lemma 2.1 that S has to pay the cost of arc e_v in E if and only if there is a player j in S such that $j \in N_{F(v)}$. □

Example 2.3. Consider the three player standard fixed tree problem as is graphically depicted in Figure 1.

The different vertices are depicted as circles. Each of the encircled numbers corresponds to the location of the corresponding player. The black vertex has

[2] Note in this respect that $F(r) = V$, and that $i \in F(v(i))$ for all $i \in N$. Moreover, v is a follower of $\tilde{v}$ if $\tilde{v}$ precedes v.

no residing players. The root is depicted as a triangle. Furthermore, the arcs in the tree network are represented by the line segments connecting the different vertices. The cost of a specific link is put next to the corresponding line segment. For this situation we have

$$c_\Gamma = 2u^*_{\{1\}} + 2u^*_{\{3\}} + 4u^*_{\{2,3\}} + 3u^*_{\{1,2,3\}}. \qquad \triangleleft$$

So each standard fixed tree game (N, c_Γ) is a *positive* cost game, i.e. c_Γ is a positive linear combination of dual unanimity games. In this way we obtain the concavity of standard fixed tree games (cf. Granot *et al.* (1996)) in an immediate and alternative way, by the fact that positive cost games are concave. Koster and Tijs (2000) characterize the class of standard fixed tree games by the representation as a linear combination of dual unanimity games.

3. The tree maintenance game and its core

To further fix ideas, for the next sections on standard fixed tree games, we will focus on the special cases, where at each vertex, except the root, there is exactly one player. Standard fixed tree problems with this property will be referred to as *tree maintenance problems* and the corresponding games are *tree tree maintenance games*.

Assumption. Below we will assume that $\Gamma = (G, c, N)$ is a fixed tree maintenance problem. The player set N will be identified with the set of vertices $V \backslash \{r\}$. In this terminology, for instance, an arc $(\pi(i), i) \in E$ for some $i \in N$ will be denoted e_i. In addition we will denote $\sum_{i \in S} c(e_i)$ by $c(S)$ for all $S \subseteq N$.

The problem under consideration is to divide the the cost $c_\Gamma(N)$ among the players in N. A *vector of cost shares* is by definition a vector $x \in \mathbb{R}^N$ such that $\sum_{i \in N} x_i = c_\Gamma(N)$. Here x_i represents the amount player i has to pay according to x. Special vectors of cost shares are those that are in the core of the game, i.e. the set of vectors that satisfy a coalitional rationality constraint. Formally, the core of a cost game $(N, g) \in \mathscr{G}$, $core(N, g)$, consists of all vectors $x \in \mathbb{R}^N$ with the following two properties:

$$\text{(i)} \quad \sum_{i \in S} x_i \leq g(S) \quad \text{for all } S \subseteq N, \quad \text{(ii)} \quad \sum_{i \in N} x_i \geq g(N).$$

Property (i) expresses the idea that no coalition should contribute more than the cost they are charged for if they do not cooperate. The property (ii) states that x is a *feasible* vector of cost shares, so that the cost of the grand coalition can be covered. Then, trivially, (i) and (ii) together imply that a core element x is a vector of cost shares.

The first part of the section deals with alternative expressions of the core, while the last part is devoted to its geometric properties. There are easy ways to characterize the core of the game (N, c_Γ). We show that the core consists of those allocations according to which each agent has to make at least a zero contribution and for which the core inequalities are met for those coalitions being trunks.

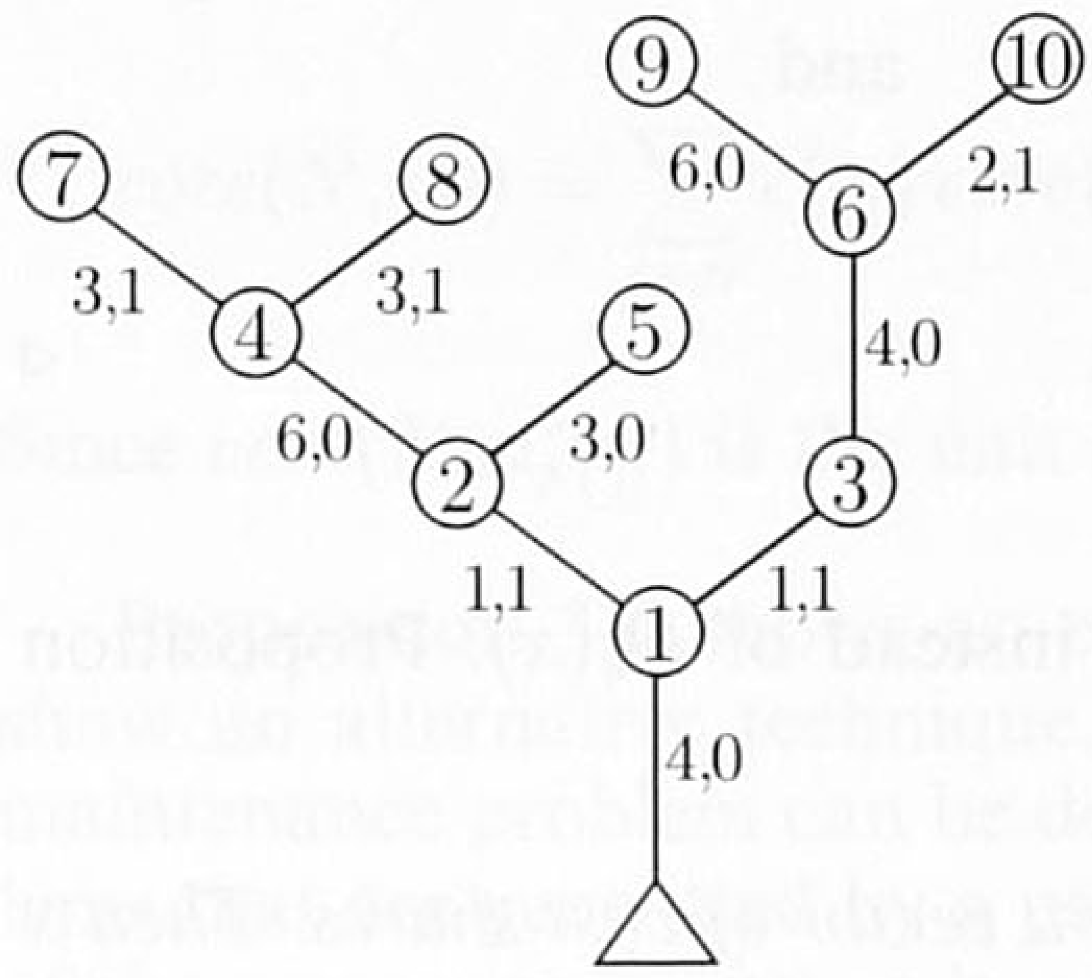

Fig. 2. A tree maintenance problem.

Proposition 3.1. *The vector* x *is a core element of* (N, c_Γ) *if and only if* $x \geq 0$ *and* $x(T) \leq c(T)$ *for each trunk* T.[3]

Proof: Trivially, if $x \in core(N, c_\Gamma)$, then $x \geq 0$ and $x(T) \leq c(T)$ for each trunk T. Conversely, let x be a nonnegative vector of cost shares such that $x(T) \leq c(T)$ for each trunk T. Let $S \subseteq N$ be a nonempty coalition. Then $S \subseteq T_S$ and therefore it holds that $c_\Gamma(S) = c(T_S) \geq x(T_S) \geq x(S)$. □

Let $e = (i, j)$ be an arc of G, and define $B_e := (V_e, E_e)$ by $V_e := F(j)$ and $E_e := \{(k, \ell) \in E \mid \ell \in F(j)\}$. B_e will be referred to as the *branch rooted at* e.[4] Given some vector of cost shares x, the *overflow over the arc* $e \in E$ is given by

$$O_e(x) = \sum_{\ell \in V_e} (x_\ell - c(e_\ell)).$$

The overflow over some arc $e \in E$ is interpreted as the amount that the inhabitants of V_e, i.e. the players located at some vertex in the branch B_e, pay in excess of the cost of the set of arcs E_e.

Example 3.2. Consider the tree maintenance problem as is depicted in Figure 2. Here the first number at an arc indicates the corresponding total costs, and the second the related overflow for the core element $x = (2, 2, 2, 4, 3, 3, 4, 4, 6, 3)^T$. For instance, the costs of e_4 and e_5 are 6 and 3, respectively. The branches rooted at e_2 and e_6 are given by

$$B_{e_2} = (\{2, 5, 4, 7, 8\}, \{e_2, e_4, e_5, e_7, e_8\}) \quad \text{and}$$

$$B_{e_6} = (\{6, 9, 10\}, \{e_6, e_9, e_{10}\}), \quad \text{respectively.}$$

Then we calculate the overflows at e_2 and e_6,

---

[3] In fact a strengthening of this result appeared in Granot and Maschler (1998): one needs only to consider trunks with one so called *outgoing arc*.

[4] According to the terminology introduced in Granot *et al.* (1996), B_e is the branch at i in the direction of j if $e = (i, j)$.

$$O_{e_2}(x) = \sum_{i \in \{2,4,5,7,8\}} (x_i - c(e_i)) = 17 - 16 = 1, \quad \text{and}$$

$$O_{e_6}(x) = \sum_{i \in \{6,9,10\}} (x_i - c(e_i)) = 12 - 12 = 0. \qquad \triangleleft$$

If $e = (i, j)$, we will sometimes write $O_j(x)$ instead of $O_e(x)$. Proposition 3.1 can be easily restated in terms of overflows.

Proposition 3.3 (Bjørndal *et al.* (1999)). *Let x be a vector of cost shares. Then x is a core element if and only if* $x \geq 0$ *and* $O_e(x) \geq 0$ *for all* $e \in E$.

Proof: Let x be a vector of cost shares. We claim that x is a core element if and only if $x \geq 0$ and for each arc $e = (i, j) \in E$,

$$\sum_{\ell \in V_e} x_\ell \geq \sum_{e' \in E_e} c(e') \tag{5}$$

where $B_e = (V_e, E_e)$ is the branch rooted at e.[5] The claim is proved as follows. The complement in V of $V_e \backslash \{i\}$ is a trunk. Therefore the result follows from budget balance and the application of Proposition 3.1. □

The next proposition shows that every core element is obtained by means of splitting, arbitrarily, the cost of each arc among its users.

Proposition 3.4. *The vector x is a core element if and only if there exist* $y^1, \ldots, y^n$ *such that* y^j *is a point in the unit simplex in* $\mathbb{R}^{F(j)}$ *for all* $j = 1, \ldots, n$ *and*

$$x_i = \sum_{j \in N(P_i)} y_i^j c(e_j) \quad \text{for all } i \in N. \tag{6}$$

Proof: In Dragan *et al.* (1989) it is proved that the core is additive on the cone of concave games. In short, a proof of this reads as follows. First, for all cost games (N, v) and (N, w) it holds that $core(N, v) \oplus core(N, w) \subseteq core(N, v + w)$. Here $\oplus$ denotes the direct sum operator. The Weber set $W(N, v)$ corresponding to a game (N, v) is the convex hull of the $|N|!$ marginal vectors of (N, v). W is subadditive as a multifunction with respect to the characteristic function, i.e. $W(N, v + w) \subseteq W(N, v) \oplus W(N, w)$ for all (N, v), $(N, w) \in \mathcal{G}$. Furthermore on the class of concave games the Weber set and the core coincide (see Driessen (1988) or Ichiishi (1981)). So if (N, v), $(N, w) \in \mathcal{G}$ are concave we also have the reversed inclusion, $core(N, v + w) = W(N, v + w) \subseteq W(N, v) \oplus W(N, w) = core(N, v) \oplus core(N, w)$, and consequently $core(N, v + w) = core(N, v) \oplus core(N, w)$.

All elements of the basis $\{(N, u_S^*)\}_{S \subseteq N \backslash \{\emptyset\}}$ are concave, and consequently all nonnegative linear combinations. Then the additivity of the core on the cone of concave cost games together with Proposition 2.2 gives

[5] A proof of this claim is in Granot *et al.* (1996).

$$core(N, c_\Gamma) = \sum_{j \in N} c(e_j)\, core(N, u^*_{F(j)}). \tag{7}$$

Since $core(N, u^*_{F(j)})$ is the unit simplex in $\mathbb{R}^{F(j)}$, we are done. □

Proposition 3.4 shows an easy way to generate core elements. Below we show an alternative technique, by the notion of a *pseudo subtree*. Each tree maintenance problem can be decomposed into smaller tree maintenance problems that are generated by a pseudo subtree. If we take core elements for each of these components, then in a natural way these can be combined into a core element of the original problem. Moreover, we show that each core element for the tree maintenance game can be obtained following such a procedure, by choosing the right *partition into pseudo subtrees*. Conversely, at each core element x there is a unique finest partition $\mathscr{T}(x)$ into pseudo subtrees such that the restriction of x to each related subproblem is a core element of the corresponding subgame. First we will formalize the notion of a pseudo subtree.

Definition. A *pseudo subtree* of a tree $G = (V, E)$ is a connected subgraph $G' = (V', E')$ such that there exists an $r' \in V'$ such that

(i) r' is the minimal element in V' with respect to $\preccurlyeq$,
(ii) there is exactly one vertex in V' that has r' as predecessor.

A pseudo subtree $G' = (V', E')$ of G rooted at r' yields a *restricted tree maintenance problem* $\Gamma' = (G', c', N')$ where c' is the restriction of c to E' and $N' = V' \backslash \{r'\}$.

Let $(S_k \cup \{r_k\}, E_k)$ with root r_k be a pseudo subtree of G for $k = 1, 2 \cdots p$. Then the collection $\{(S_k \cup \{r_k\}, E_k) \mid k = 1, 2, \ldots, p\}$ is a *partition of G into pseudo subtrees* if $\{S_1, S_2, \ldots, S_p\}$ is a partition of the vertex set N.

Example 3.5. Consider the tree $G = (\{r, 1, 2, 3, 4\}, \{(r, 1), (1, 2), (2, 3), (1, 4)\})$ and the corresponding tree maintenance problem (see also Figure 3). Define

$$G_1 = (\{r, 1, 2\}, \{(r, 1), (1, 2)\}),$$

$$G_2 = (\{2, 3\}, \{(2, 3)\}), \quad \text{and}$$

$$G_3 = (\{1, 4\}, \{(1, 4)\})$$

It is easily checked that G_1, G_2, and G_3 are pseudo subtrees, and together they constitute a partition of G. ◁

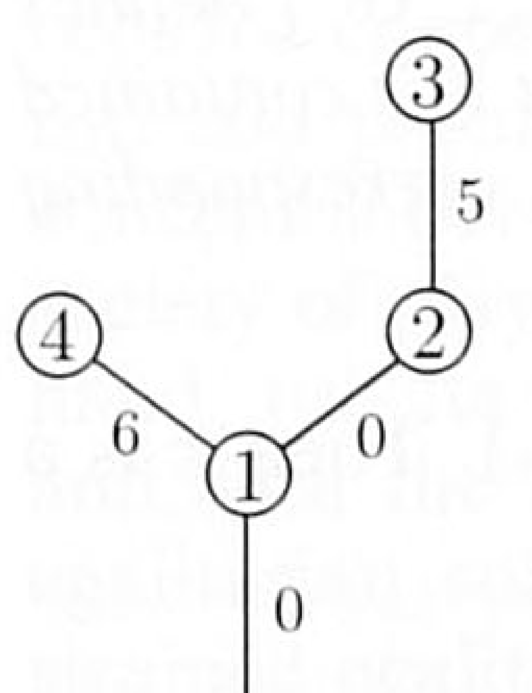

Fig. 3. Partitioning a tree by considering zero overflows.

Each vector of cost shares x defines a partition into pseudo subtrees. To see this, let $E(x)$ be the set of arcs corresponding to positive overflows, i.e. $E(x) = \{e \in E \mid O_e(x) > 0\}$. The graph $(V, E(x))$ contains p connected subgraphs G_k, where $1 \le p \le n$. For each of these subgraphs G_k, for $1 \le k \le p$, we construct a pseudo subtree G^k with player set $N(G_k)$. Let $r_k \in V \backslash N(G_k)$ be such that $r_k \in V(P_i)$ for every $i \in N(G_k)$, and $r_k = \pi(i)$ for exactly one $i \in N(G_k)$. Let $V(G_k) := N(G_k) \cup \{r_k\}$ and $E(G_k) := \{e = (i,j) \mid e \in E, i, j \in V(G_k)\}$. Then $G^k := (V(G_k), E(G_k))$ is a pseudo subtree with root r_k, and $\mathscr{T}(x) = \{G^1, G^2, \dots, G^p\}$ is a partition of G into pseudo subtrees. We refer to $\mathscr{T}(x)$ as the partition of G *induced by* x.

Example 3.6. Recall the tree maintenance problem $\Gamma = (G, c, N)$ as in Example 3.5, where the arc costs are given by $c(e) = 10$ for all $e \in E$. The allocation $x = (4, 5, 15, 16)^T$ is a core element, and the corresponding overflows are indicated next to the arcs in Figure 3. By removing all the arcs with zero overflows, we obtain the partition of G into the pseudo subtrees G^1 and G^2, where $N(G^1) = \{1, 4\}$, $N(G^2) = \{2, 3\}$, $r_1 = r$, and $r_2 = 1$. ◁

In the obvious way we will refer to branches, paths and followers restricted to some pseudo subtree. More specifically for a pseudo subtree G^* and $i \in N(G^*)$, $e \in E(G^*)$ we define

- $\tilde{F}(i)$ the set of followers of i in G^*, i.e. $F(i) \cap N(G^*)$,
- $\tilde{P}_i$ the path from the local root r^* of G^* to i,
- $\tilde{B}_e$ the branch in G^* rooted at e,
- $\tilde{O}_e(x)$ the overflow at e in G^* with respect to x, i.e. the sum $\sum_{\ell \in N(\tilde{B}_e)} (x_\ell - c(e_\ell))$.

Example 3.7. Consider the tree maintenance problem as in Example 3.6. Two pseudo subtrees are determined, $G^1 = (\{r, 1, 4\}, \{e_1, e_4\})$, and $G^2 = (\{r_1, 2, 3\}, \{e_2, e_3\})$. Then, for instance, we have $\tilde{P}_3 = G^2$, $\tilde{B}_{e_2} = (\{2, 3\}, \{e_2, e_3\})$, $\tilde{F}_1 = \{1, 4\}$, and $\tilde{O}_{e_2} = O_{e_2} = 0$. ◁

Proposition 3.8.
(i) *Let $\mathscr{T} = \{G^1, \dots, G^p\}$ be a partition of G into pseudo subtrees. Then*

$$\prod_{k=1}^{p} core(S_k, c_{\Gamma^k}) \subseteq core(N, c_\Gamma), \tag{8}$$

where (S_k, c_{Γ^k}) is the cost game corresponding to the restricted maintenance problem $\Gamma^k = (G^k, c^k, S_k)$.[6]
(ii) *Let x be a core element for (N, c_Γ). Then $\mathscr{T}(x) = \{G^1, G^2, \dots, G^p\}$ defines the unique finest partition of G into pseudo subtrees, such that x is contained in the set $\prod_{k=1}^{p} core(S_k, c_{\Gamma^k})$, where $\Gamma^1, \Gamma^2, \dots, \Gamma^p$ are the corresponding restricted maintenance problems.*

Proof: Let $x = (y^1, \dots, y^p)$ be an element of $\prod_{k=1}^{p} core(S_k, c_{\Gamma^k})$. Then x is a vector of cost shares since

[6] Here $\prod$ denotes the Cartesian product operator.

$$\sum_{i \in N} x_i = \sum_{k=1}^{p} \sum_{i \in S_k} y_i^k = \sum_{k=1}^{p} \sum_{i \in S_k} c^k(e_i) = \sum_{i \in N} c(e_i).$$

Moreover, $x \geq 0$, since $x_{S_j} \in core(S_j, c_{\Gamma^j})$ implies $x_{S_j} \geq 0$ for all j. According to Proposition 3.1 we only need to prove that $x(T) \leq c(T)$ for each trunk T. Let T be a trunk of G. For any $k = 1, \dots, p$, let T^k be the set of vertices $T \cap S_k$. Then $T^k \cup \{r_k\}$ is a trunk of $G^k = (S_k \cup \{r_k\}, E_k)$ for all $k \in \{1, \dots, p\}$ for which $T^k \neq \emptyset$. Therefore

$$x(T) = \sum_{\substack{1 \leq k \leq p \\ T^k \neq \emptyset}} \sum_{i \in T^k} y_i^k \leq \sum_{\substack{1 \leq k \leq p \\ T^k \neq \emptyset}} \sum_{i \in T^k} c^k(e_i) = \sum_{i \in T} c(e_i).$$

This proves part (i).

(ii) First we will show that $\mathscr{T}(x)$ is the finest partition into pseudo subtrees such that x is an element of the Cartesian product of the cores of the corresponding tree maintenance games. Suppose that $\{G_1', G_2', \dots, G_t'\}$ defines a partition into pseudo subtrees such that $x \in \prod_{k=1}^{t} core(N(G_k'), c_{\Gamma_k'})$, where Γ_k' is the tree maintenance problem corresponding to G_k'. Then for all $k \in \{1, 2, \dots, t\}$, $x_{N(G_k')}$ is a vector of cost shares for $(N(G_k'), c_{\Gamma_k'})$, and thus $\tilde{O}_{\pi^{-1}(r_k')}(x) = 0$ for the root r_k' of G_k'. But by construction this means that for all $k \in \{1, 2, \dots, t\}$ we have $r_k' \in \{r_1, r_2, \dots, r_p\}$, the set of roots corresponding to the partition into pseudo subtrees $\mathscr{T}(x)$. But this means that $\mathscr{T}(x)$ is a refinement of $\{G_1', G_2', \dots, G_t'\}$.

Next, we use the core characterization in Proposition 3.3. Let $1 \leq k \leq p$. Because $x \in core(N, c_\Gamma)$ is a vector of cost shares with respect to the game (N, c_Γ), and since $\mathscr{T}$ has been constructed by removing only arcs with zero overflows, it is clear that $x_{N(G^k)}$ is a vector of cost shares with respect to the game $(N(G^k), c_{\Gamma^k})$. Also, $x_{N(G^k)} \geq 0$ follows from $x \in core(N, c_\Gamma)$ and Proposition 3.3. We will complete the proof by showing that $\tilde{O}_i(x) = O_i(x) \geq 0$ for all $i \in N(G^k)$, where the inequality follows from $x \in core(N, c_\Gamma)$ and Proposition 3.3. Note that, by the construction of $\mathscr{T}(x)$, $\tilde{O}_i(x) = x_i - c(e_i) = O_i(x)$ for any $i \in N(G^k)$ such that i is a leaf in G^k, since i must either be a leaf in G, or we must have $O_j(x) = 0$ for every $j \in \pi^{-1}(i)$. Then, for every $i \in N(G^k)$ such that i is not a leaf in G^k, $\tilde{O}_i(x) = (x_i - c(e_i)) + \sum_{j \in \pi^{-1}(i) \cap \tilde{F}(i)} \tilde{O}_j(x) = (x_i - c(e_i)) + \sum_{j \in \pi^{-1}(i)} O_j(x) = O_i(x)$. □

4. Egalitarianism in tree maintenance games

The *constrained egalitarian solution* of Dutta and Ray (1989) is a solution concept for cooperative cost games which combines commitment for egalitarianism and promotion of individual interests in a consistent way. This solution concept is developed in a framework where, on one hand, each member in the society of players believes in egalitarianism as a social value, and on the other hand, private preferences dictate selfish behaviour. For concave cost games, and with the Lorenz ordering as the measure of inequality, the constrained egalitarian solution is the unique Lorenz maximal core element. The constrained egalitarian solution, however, deals with completely symmetric players. In many situations this seems an overly strong assumption. For a discussion on examples where a lack of symmetry is present, the reader is refered to Kalai and Samet (1987, 1988) and Shapley (1981). It is assumed that the asymmetries between the players are reflected by an exogeneously given vector of positive weights, which is based on considerations not captured by the parameters of the game itself. Ebert (1999) studies a weighted variant of the Lorenz ordering, by application of the Lorenz ordering on the weighted allocations. Koster (2002) generalizes this idea, by the introduction of an hierarchical structure as in Kalai and Samet (1988). Then, given such a *weighted Lorenz ordering*, Koster (2002) shows that, despite the partial nature of the weighted (hierarchical) Lorenz orderings, there is at most one maximal stable allocation; by stable we mean that a coalition may block only if they can propose a weighted egalitarian solution for the society of players it constitutes, for the induced subgame. For concave cost games existence is proved. Moreover, in general an (exponential) algorithm is proposed for calculating the solution. In this paper we will discuss the algorithm, and define the weighted constrained egalitarian solution as the resulting allocation. First we need to define the weighted cost of a coalition.

Definition. The *weighted average cost* of a coalition S in a cost game $(N, c) \in \mathscr{G}$ for a given weight vector $\omega \in \mathbb{R}^N_{++}$ is defined by

$$\alpha_\omega(c, S) = \frac{c(S)}{\omega(S)}.$$

Now the weighted constrained egalitarian solution is calculated by consecutively determining the maximal coalitions that minimize the weighted marginal cost.

Definition. The *weighted constrained egalitarian solution* for a concave cost game $(N, c) \in \mathscr{G}$ and vector of positive weights ω equals $\mathrm{ALG}(N, c, \omega)$, the vector of cost shares obtained at termination of the following algorithm:

1 *Input*: $(N, c) \in \mathscr{G}$, $\omega \in \mathbb{R}^N_{++}$.
2 *Set* $c_1 = c$, $N_1 := N$.
3 *Repeat, as long as* $N_i \neq \emptyset$, *the following step.*
Determine the unique maximal coalition $S_i \subseteq N_i$, *that minimizes the weighted average cost in* c_i.[7] Put for all $j \in S_i$, $\mathrm{ALG}_j(N, c, \omega) := \omega_j \alpha_\omega(S_i, c_i)$. *Let* $N_{i+1} := N_i \backslash S_i$, *and define* $(N_{i+1}, c_{i+1}) \in \mathscr{G}$ *by*

$$c_{i+1}(S) = c_i(S \cup S_i) - c_i(S_i) \quad \textit{for all } S \subseteq N_{i+1}.$$

4 *Output is* $\mathrm{ALG}(N, c, \omega)$.

Dutta and Ray (1989) show that in case of a concave game (N, c) and with equally weighted players, $\mathrm{ALG}(N, c, \omega)$ is the weighted constrained egalitarian

[7] For concave games the set of coalitions that minimize the average weighted value is closed under union.

solution, the Lorenz-dominant element in $core(N,c)$. Koster (2002) shows that this result carries over to various positive weights and corresponding Lorenz ordering as in Ebert (1999).

Clearly, the above framework still lacks the possibility of dealing with completely asymmetric situations, where it is socially desirable to enforce the largest transfer of all the economic opportunities from a certain group of players S to those in $N\backslash S$, such that it is not in the interest of any subcoalition of S to separate. S should enjoy as less as possible from the beneficial cooperation of the grand coalition, apart from the cost savings they are able to generate themselves through internal cooperation. Also within S there may be a set S' of players that should have *zero impact* compared to those in $S\backslash S'$, expressing that society demands the highest possible transfer of the economic prosperity of this group to the higher rewarded players in $N\backslash S$ and $S\backslash S'$. We will focus on the situation where the society can enforce these transfers, without needing the consent of players of S'. Still we will allow a proposed allocation to be attacked by S' if some of its members are able to do better without the support of other players. So society may enforce cooperation on a large scale, but it has to be sensible to the possible disagreements raised by the selfishness of the subgroups of players. A way to model fully asymmetric situations is by *hierarchical systems*. The concept is, in mathematical terms, equivalent to the weight systems in Kalai and Samet (1987, 1988), Monderer *et al.* (1992). Using hierarchical systems, we will be able to generalize weighted constrained egalitarian ideas.

Definition. A *hierarchical system* for N consists of an ordered pair $\Sigma = (\mathcal{S}, \omega)$, where $\mathcal{S} = (S_1, S_2, \ldots, S_k)$ is an ordered partition of N, and $\omega \in \mathbb{R}^N_{++}$ is a vector of positive weights.

In order to determine the set of weighted constrained egalitarian solutions, the following algorithm may be useful. It is based on the former algorithm for cost games (N,c) with trivial hierarchical systems, according to which $\mathcal{S} = \{N\}$. Firstly, given the hierarchical system for N, $((S_1, \ldots, S_k), \omega)$, we define the game $(S_k, c_k) \in \mathcal{G}$ by $c_k(S) = c(S)$ for all $S \subseteq S_k$. Then we proceed by defining inductively the reduced games $(S_j, c_j) \in \mathcal{G}$ for $j = k-1, k-2, \ldots, 1$ by

$$c_j(S) := c\left(\bigcup_{\ell=j+1}^{k} S_\ell \cup S\right) - c\left(\bigcup_{\ell=j+1}^{k} S_\ell\right) \quad \text{for all } S \subseteq S_j.$$

Then, accordingly we first apply ALG to the game (S_k, c_k) with the vector of weights ω_{S_k}. Then we proceed with determining the allocation for each of the players in S_{k-1}, by applying ALG on the reduced cost game (S_{k-1}, c_{k-1}) with corresponding weight vector $\omega_{S_{k-1}}$. Next, we continue with the players in S_{k-2} and the game (S_{k-2}, c_{k-2}), etc. ...

Definition. The *weighted constrained egalitarian solution* for a concave cost game (N,c) and hierarchical system Σ is given by $\text{ALG}^*(N,c,\Sigma)$, i.e. the profile of cost shares that is obtained by performing the following algorithm.

1. *Input: game* (N, c), *hierarchical system* $\Sigma = (\mathscr{S}, \omega)$.
2. *For* $i = k$ *to*
   do $\mathrm{ALG}(S_i, c_i, \omega_{S_i})$
   Put $\mathrm{ALG}^*_{S_i}(N, c, \Sigma) := \mathrm{ALG}(S_i, c_i, \omega_{S_i})$.
3. *Output:* $\mathrm{ALG}^*(N, c, \Sigma)$.

The weighted constrained egalitarian solution for $(N, c) \in \mathscr{G}$ and hierarchical system $\Sigma = (\mathscr{S}, \omega)$ is denoted $\mathrm{CES}^{\Sigma}(N, c)$, and $\mathrm{CES}^{\omega}(N, c)$ if Σ is trivial or $\mathscr{S} = \{N\}$.

Firstly, notice that Step 3 of ALG is of exponential complexity, and consequently this complexity is inherited by ALG^*. In this section we will reduce the number of necessary operations for tree maintenance games, such that the upperbound equals a multiple of $|N|^2$ operations at most. The polynomial algorithm to calculate $\mathrm{CES}^{\Sigma}(N, c_{\Gamma})$ hinges on a *dynamic* approach similar to the calculation of the nucleolus for standard fixed tree games in Maschler *et al.* (1995).

Interpret the vertices in V as the villages of the different players and the arcs in E as the roads to the capital city of the region (root). The roads are deteriorated and before usage they need tarring. The tarring costs are assumed 1 per unit of length, so that the cost of an arc is identified with its length. Fix a hierarchical system $\Sigma = (\mathscr{S}, \omega)$ with $\mathscr{S} = (S_1, S_2, \ldots, S_k)$ and $\omega \in \mathbb{R}^N_{++}$. The corresponding weighted constrained egalitarian solution is determined as the total individual tarring length provided that

(i) every worker keeps working as long as the road from the capital to his residence has not been completed,[8]
(ii) every worker does his job on an unfinished segment between the capital and his home village that is closest to his home village,
(iii) the workers in layer S_k start, and the starting times for the other workers are specified as the finishing time of the preceding lower layer in the hierarchy (i.e. layer S_t starts when S_{t+1} is finished), and, finally,
(iv) the tarring speed of the individual worker i in layer t is set to ω_i.

The individual cost shares are determined by the distance that the individual agents cover until their corresponding paths are entirely tarred. In this way, once players get to work in some group at one and the same road, each of them is charged for the fraction of the incurred cost corresponding to tarring the unfinished part that is proportional to his weight. Due to the way of distributing costs we will interpret the weights as *contribution rates*. We will see that this dynamic approach amounts to calculating the individual cost shares in a finite number of stages; each of the different stages corresponds to the actual status of the work procedings at the very moments that tarring of a specific road is realized. First we formally describe the algorithm and we prove its validity for calculating weighted constrained egalitarian allocations.

Algorithm 4.1. Given the maintenance problem $\Gamma = ((V, E), c, N)$ and weight system $\Sigma = ((S_1, S_2, \ldots, S_k), \omega)$ for N, the *home-down allocation* $h^{\Sigma}(\Gamma)$ is the output obtained below at Step 4.

[8] Notice that the direction of the procedure is opposed to the direction of the arcs in the network, who are all directed away from the root.

- Step 1: *Initialization*
  - Set stage counter $p = 1$.
  - Set of connected players is $N_C = \emptyset$.
  - The set of active players is $N^A(1) := S_k$.
  - The set of active arcs is $E^A(1) = \{e_i \mid i \in S_k\}$.
  - The active players at e in stage 1 are given by

$$N^A(e, 1) = \begin{cases} \{i\} & \text{if } e = e_i \in E^A(1), \\ \emptyset & \text{otherwise.} \end{cases}$$

  - The set of finished arcs is $E^f_{\_}(1) = \emptyset$.
  - Put $x^f_{\_}(e, 1) = 0$ for all $e \in E$, i.e., the proceedings for each arc $e \in E$ is 0 at the start.
- Step 2:
  (a) For $e \in E^A(p)$, calculate $t(e, p) := \dfrac{c(e) - x^f_{\_}(e, p)}{\omega(N^A(e, p))}$,
  i.e. the time that the active players at e need to finish e.
  (b) Determine $t(p) = \min\{t(e, p) \mid e \in E^A(p)\}$
  i.e. the time that the active players work at stage p.
  (c) Determine $E^f(p) := \{e \in E^A(p) \mid t(e, p) = t(p)\}$
  i.e. the set of arcs that are finished at stage p.
  (d) Determine $E^f_{\_}(p + 1) := E^f_{\_}(p) \cup E^f(p)$
  i.e. the set of arcs that are finished at stage p and earlier.
  (e) For $e \in E^A(p)$
  determine $x^f(e, p) := \omega(N^A(e, p)) \cdot t(p)$,
  (i.e. the part of e that is completed at stage p),
  put $x^f_{\_}(e, p + 1) := x^f_{\_}(e, p) + x^f(e, p)$,
  (i.e. the part of e that has been completed so far),
  $h_{N^A(e,p)} := h_{N^A(e,p)} + \omega_{N^A(e,p)} \cdot t(p)$
  i.e. the individual efforts of the active players are accumulated
- Step 3: *Determine newly connected players, and add them to N_C. By setting $N^A(p + 1) = N^A(p) \backslash N_C$, these players become inactive in the next stage.* Check if $N^A(p) \backslash N_C = \emptyset$. If true then do (1), and (2) otherwise.
  (1) Determine the highest index j with $S_j \backslash N_C \neq \emptyset$, put $N^A(p + 1) = S_j \backslash N_C$, $p := p + 1$, and go to Step 2. If there is no such j, terminate by going to Step 4.
  (2) Assign nonconnected active players to (new) arcs for the next stage, as follows:
    (i) Nonfinished active arcs remain active, as well as the corresponding players located there
    (ii) For each finished arc e_i such that $\pi(i) \notin N_C$ determine the closest nonfinished arc $\tilde{e}$ on the path P_i. Then, at the next stage, $\tilde{e}$ is active, and together with the players $N^A(\tilde{e}, p)$, the players $N^A(e_i, p)$ are assigned to $\tilde{e}$. Increase the stage number, $p = p + 1$, and go to Step 2.
- Step 4: The output is h.

Clearly, the algorithm is well-defined, i.e. it stops after at most $P \leq |N|$ stages, since at each stage at least one arc is completed and $|E| = |N|$. The way the algorithm works is clarified in the next example.

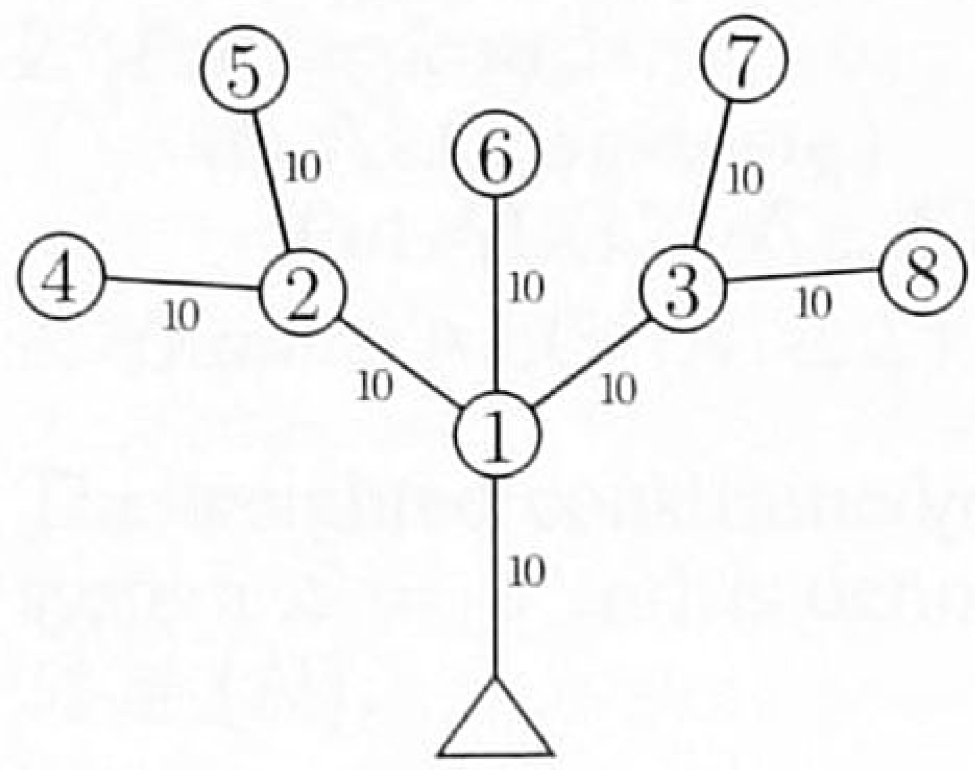

Fig. 4. The situation at stage 0.

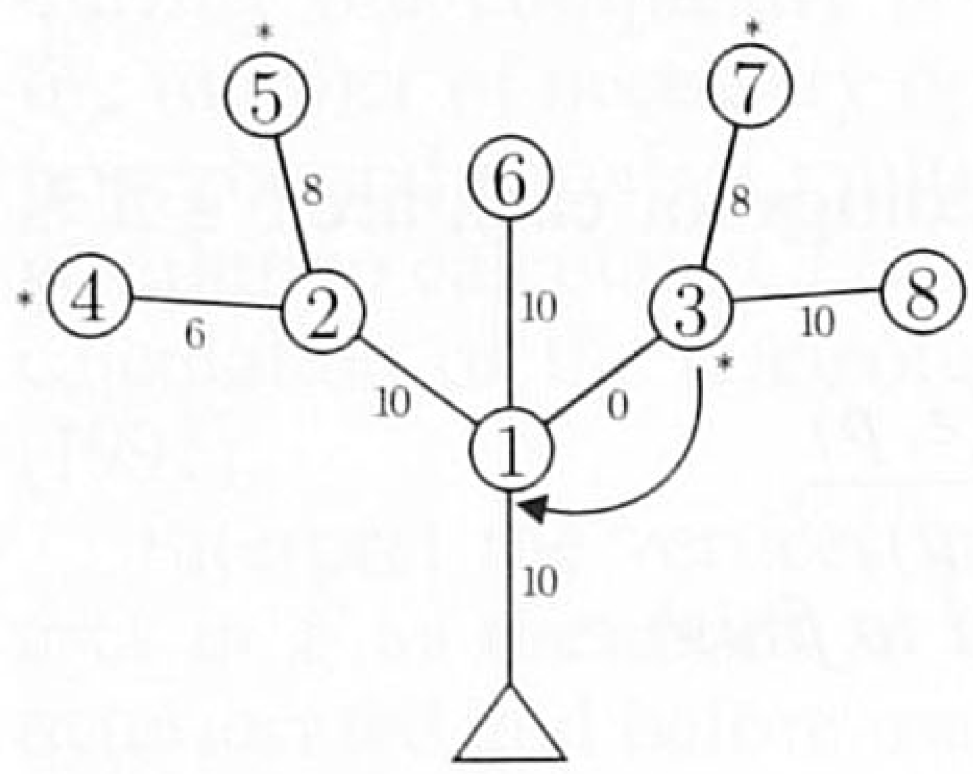

Fig. 5. Cost reduction through stage 1.

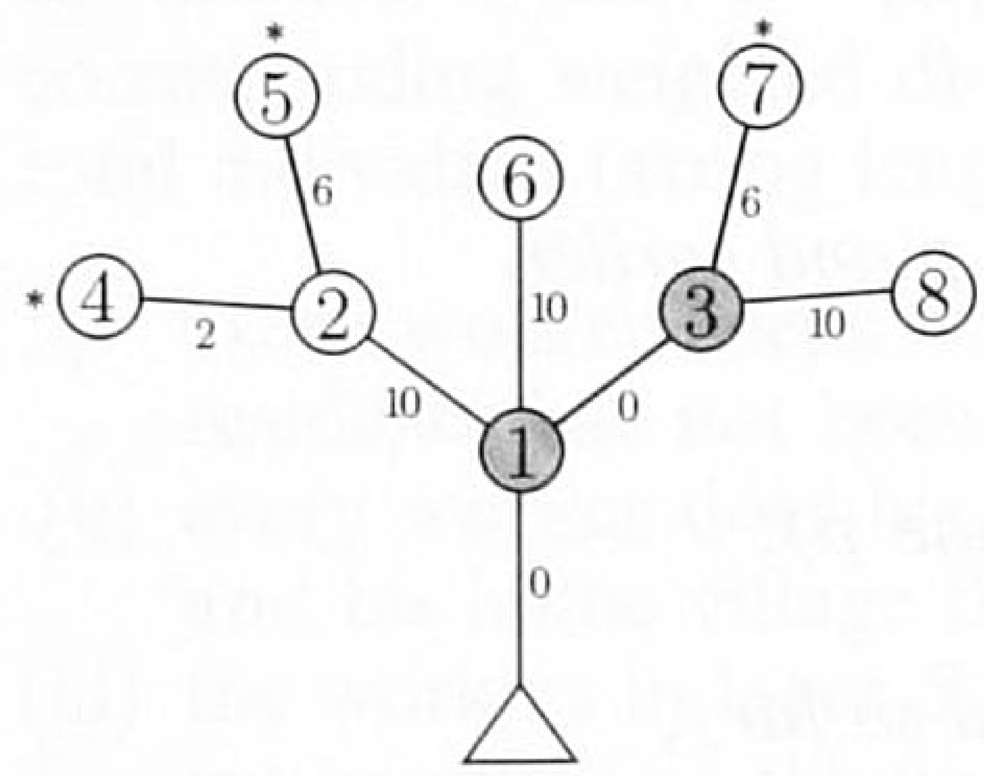

Fig. 6. Stage 2 of Algorithm 4.1.

Example 4.2. Consider the tree network as in Figure 4. We show how the Algorithm 4.1 determines the home-down allocation corresponding to the weight system Σ with partition of the player set $(\{1,6,8\},\{2\},\{3,4,5,7\})$ and weight vector $\omega = (1,1,5,2,1,2,1,2)^T$.

At the first stage the active players are $N^A(1) = \{3,4,5,7\}$, which is graphically expressed by the stars in Figure 5. The set of activated arcs is given by $E^A(1) = \{e_3, e_4, e_5, e_7\}$. We calculate the time needed to finish the different active arcs:

$$t(e_3, 1) = \tfrac{1}{5}(10-0) = 2,$$

$$t(e_4, 1) = \tfrac{1}{2}(10-0) = 5,$$

$$t(e_5, 1) = \tfrac{1}{1}(10-0) = 10, \quad \text{and}$$

$$t(e_7, 1) = \tfrac{1}{1}(10-0) = 10.$$

Hence $t(1) = \min\{2,5,10,10\} = 2$. So the arc e_3 is finished first, and $E^f(1) = E^f_{-}(2) = \{e_3\}$. Consequently we determine the proceedings per active arc at the first stage by

$$x^f(e_3, 1) = t(1) \cdot \omega_3 = 10,$$

$$x^f(e_4, 1) = t(1) \cdot \omega_4 = 4,$$

$$x^f(e_5, 1) = t(1) \cdot \omega_5 = 2, \quad \text{and}$$

$$x^f(e_7, 1) = t(1) \cdot \omega_7 = 2.$$

These amounts also specify the total contributions so far, thus $x^f_-(e_3, 1) = 10$, $x^f_-(e_4, 1) = 4$, and $x^f_-(e_5, 1) = x^f_-(e_7, 1) = 2$. The individual contributions at the first stage equal the procedings of the corresponding arcs, i.e.

$$h_3 = x^f(e_3, 1) = 10,$$

$$h_4 = x^f(e_4, 1) = 4,$$

$$h_5 = x^f(e_5, 1) = 2, \quad \text{and}$$

$$h_7 = x^f(e_7, 1) = 2.$$

Now we turn to Step 3. Since $N^A(1)\backslash N_C \neq \emptyset$, we continue with (2). No path is finished, $N^A(e_3, 1) = \{3\}$ is transfered to arc e_1, by putting $N^A(e_1, 2) = \{3\}$ and afterwards $N^A(e_3, 2) = \emptyset$.

Now the stage number is increased, and we turn to stage 2. The set of active players is $N^A(2) = \{3, 4, 5, 7\}$. The players 4, 5 and 7 are assigned to their own arc, and 3 is assigned to e_1. We calculate the different finishing times:

$$t(e_1, 2) = \omega_3^{-1}(10 - 0) = 2,$$

$$t(e_4, 2) = \omega_4^{-1}(10 - 4) = 3,$$

$$t(e_5, 2) = \omega_5^{-1}(10 - 2) = 8, \quad \text{and}$$

$$t(e_7, 2) = \omega_7^{-1}(10 - 2) = 8.$$

Thus the minimal finishing time is $t(2) = \min\{2, 3, 8\} = 2$. So player 3 is again the only player finishing with the arc that he is assigned to. So $E^f(2) = \{e_1\}$ and $E^f_-(3) = \{e_1, e_3\}$. Moreover, the proceedings per arc at stage 2 are given by

$$x^f(e_1, 2) = t(2) \cdot \omega_3 = 10,$$

$$x^f(e_4, 2) = t(2) \cdot \omega_4 = 4,$$

$$x^f(e_5, 2) = t(2) \cdot \omega_5 = 2, \quad \text{and}$$

$$x^f(e_7, 2) = t(2) \cdot \omega_7 = 2.$$

Then the total proceedings per arc are

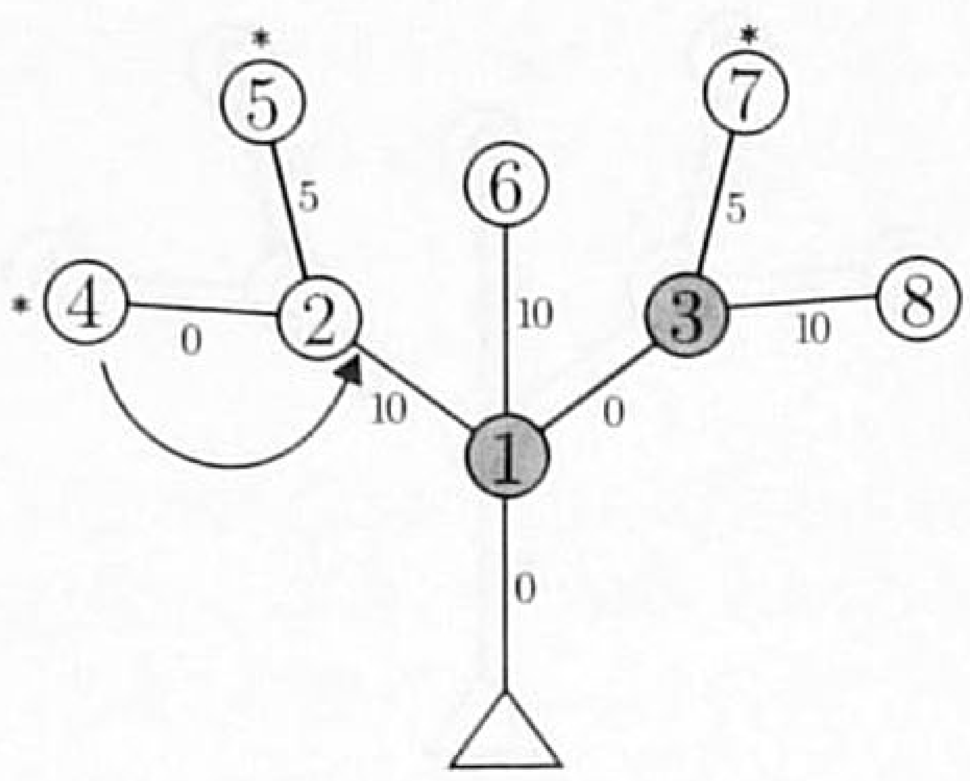

Fig. 7. Arcs e_5 and e_7 are completed at stage 4.

$$x^f_-(e_1, 2) = 10,$$

$$x^f_-(e_4, 2) = 4 + 4 = 8,$$

$$x^f_-(e_5, 2) = 2 + 2 = 4, \quad \text{and}$$

$$x^f_-(e_7, 2) = 2 + 2 = 4.$$

The newly accumulated contributions of the active players include the efforts $x(e_i, 2)$:

$$h_3 = h_3 + x^f(e_1, 2) = 10 + 10 = 20,$$

$$h_4 = h_4 + x^f(e_4, 2) = 4 + 4 = 8,$$

$$h_5 = h_5 + x^f(e_5, 2) = 2 + 2 = 4, \quad \text{and}$$

$$h_7 = h_7 + x^f(e_7, 2) = 2 + 2 = 4.$$

We move to step 3 (see Figure 7). At step 3 (1) we put $N_C = \{1, 3\}$, $N^A(3) = \{4, 5, 7\}$, and $E^A(3) = \{e_4, e_5, e_6\}$. In particular, $N^A(2)\backslash N_C \neq \emptyset$, so go to step 3 (2). Moreover, the active player sets are specified by (2): $N^A(e_4, 3) = \{4\}$, $N^A(e_5, 3) = \{5\}$, and $N^A(e_7, 3) = \{7\}$. Now go to Step 2 (3). At this stage player 4 completes e_4 in 1 additional time unit. The remaining costs for arcs e_5 and e_7 are at the same time lowered by 1 each. This follows from the following calculations:

$$\left.\begin{array}{l} t(e_4, 3) = \frac{1}{2}(10 - 8) = 1 \\ t(e_5, 3) = 10 - 4 = 6 \\ t(e_7, 3) = 10 - 4 = 6 \end{array}\right\} \Rightarrow t(3) = \min\{1, 6\} = 1.$$

Then the (total) contributions per active arc are

$$x^f(e_4, 3) = 2, \quad x^f_-(e_4, 3) = 10,$$

$$x^f(e_5, 3) = 1, \quad x^f_-(e_5, 3) = 5,$$

$$x^f(e_7, 3) = 1, \quad x^f_-(e_7, 3) = 5.$$

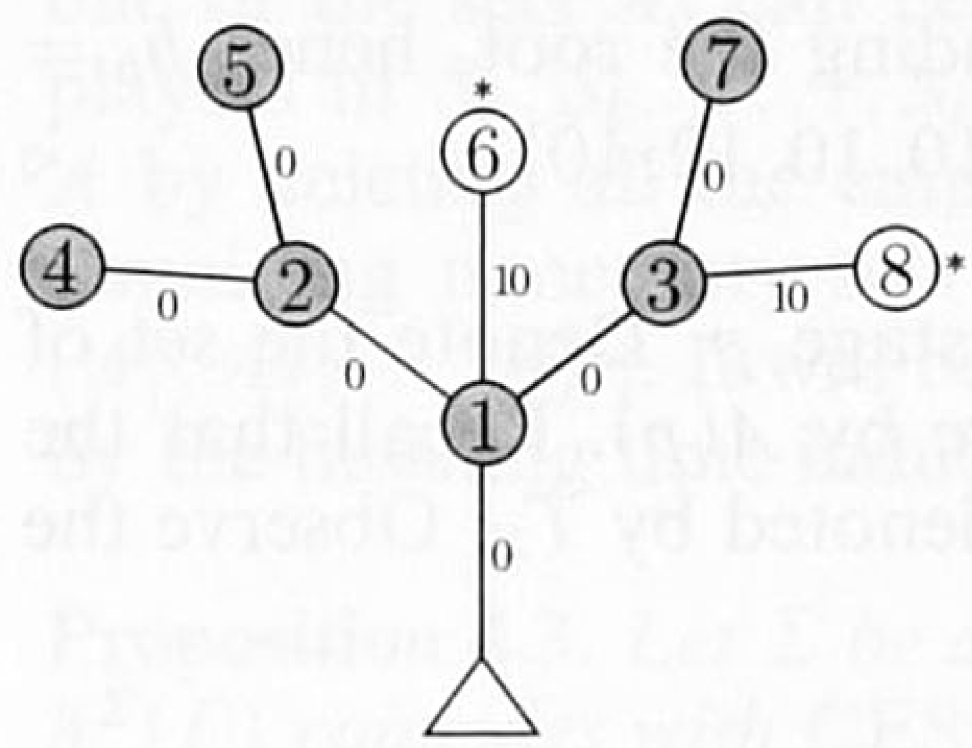

Fig. 8. The final stage: players 6 and 8 get connected.

The total contributions of the players 4, 5, and 7 are

$$h_4 = h_4 + x^f(e_4, 3) = 8 + 2 = 10,$$

$$h_5 = h_5 + x^f(e_5, 2) = 4 + 1 = 5, \quad \text{and}$$

$$h_7 = h_7 + x^f(e_7, 2) = 4 + 1 = 5.$$

Now in step 3 not much changes, except that 4 is assigned to arc e_2. So $N^A(4) = \{4, 5, 7\}$, and $E^A(4) = \{e_2, e_5, e_7\}$. Increase the stage counter to 4, go to step 2. Then at this stage, the active players simultaneously finish their arc since:

$$\left.\begin{array}{l} t(e_2, 4) = \frac{1}{2}(10 - 0) = 5, \\ t(e_5, 4) = \frac{1}{1}(10 - 5) = 5, \quad \text{and} \\ t(e_7, 4) = \frac{1}{1}(10 - 5) = 5. \end{array}\right\} \Rightarrow t(4) = 5.$$

Hence

$$x^f(e_2, 4) = 10, \quad x^f_-(e_4, 4) = 10,$$

$$x^f(e_5, 4) = 5, \quad x^f_-(e_5, 4) = 10,$$

$$x^f(e_7, 4) = 5, \quad x^f_-(e_7, 4) = 10.$$

The total contributions of the players 4, 5, and 7 are

$$h_4 = h_4 + x^f(e_2, 4) = 10 + 10 = 20,$$

$$h_5 = h_5 + x^f(e_5, 4) = 5 + 5 = 10, \quad \text{and}$$

$$h_7 = h_7 + x^f(e_7, 4) = 5 + 5 = 10.$$

At step 3 we conclude that for all active arcs e_i the paths P_i are completed; the players are connected. Hence $N_C = \{1, 2, 3, 4, 5, 7\}$. Then there is just one layer j left such that $S_j \backslash N_C \neq \emptyset$ and that is layer 1. Consequently, the set of active players becomes $N^A(4) = \{6, 8\}$, the set of active arcs is $E^A(4) = \{e_6, e_8\}$, and $N^A(e_6, 4) = \{6\}$, $N^A(e_8, 4) = \{8\}$. At the final stage (see Figure 8), players 6 and 8 both need 5 time units each to complete the corresponding arcs, and get connection to the completed component including the root, hence $h_6 = h_8 = 10$. So the final allocation is $h = (0, 0, 20, 20, 10, 10, 10, 10)^T$. ◁

Let A_p be the set of active players finished at stage p. Denote the set of active players that are finished at stage p or before by $A(p)$. Recall that the minimal trunk containing the players in $S \subseteq N$ is denoted by T_S. Observe the following facts:

(P_1) The players in some set A_p all belong to the same layer in the hierarchy.
(P_2) For all stages p, the active players that have finished here or before pay exactly for the full cost of the minimal trunk that is needed to connect them to the root, i.e.

$$c(T_{A(p)}) = \sum_{i \in A(p)} h_i^{\Sigma}(\Gamma). \tag{9}$$

Moreover, for all active players i within layer S_t that got connected with the root in stage p pay proportionally for the incremental cost of completing the arcs related to trunk T_{A_p} in addition to what has been established so far as part of $T_{A(p-1)}$, or

$$h_i^{\Sigma}(\Gamma) = \omega_i \frac{c(T_{A_p} \backslash T_{A(p-1)})}{\omega(A_p)}. \tag{10}$$

(P_3) Within each layer S_t, the weighted cost shares increase with the finishing time induced by the home-down allocation for the corresponding player, i.e. for all $i, j \in S_t$,

$$\frac{h_i^{\Sigma}(\Gamma)}{\omega_i} \leq \frac{h_j^{\Sigma}(\Gamma)}{\omega_j} \Leftrightarrow P(i) \leq P(j), \tag{11}$$

where for each player ℓ the stage at which he is first connected is denoted $P(\ell)$,
(P_4) Players that do not occur in any set A_p contribute 0.

The algorithm induces a partition of the player set as follows. Consider the tuple $(A_0^t, A_1^t, \ldots, A_{p_t}^t)$, where $p_t \in \mathbb{N}$ and $A_i^t \subseteq S_t$, $i \in \{1, 2, \ldots, p_t\}$ are such that

(i) A_0^t is the set of agents in S_t that did not become active at any stage during the algorithm,
(ii) $(A_1^t, \ldots, A_{p_t}^t) = (A_\ell, \ldots, A_{\ell+p_t-1})$ for some $\ell \leq P$.

Now consider the ordered tuple

$$A = (A_0^k, \ldots, A_{p_k}^k, A_0^{k-1}, \ldots, A_{p_{k-1}}^{k-1}, \ldots, A_0^1, \ldots, A_{p_1}^1).$$

Observe that some sets in this tuple may be empty. Especially, this is the case if at a certain stage an arc was finished and no player got connected. Also one of the sets A_0^t can be empty as no player in S_t got connected while the players in $S_k, S_{k-1}, \ldots, S_{t+1}$ were active. Create a new ordered tuple out of A by deleting all the empty sets A_i^t and by re-indexing correspondingly the remaining nonempty sets of players. Denote the resulting ordered tuple by $(A_1^*, A_2^*, \ldots, A_m^*)$. It will be referred to as the partition of the player set defined by the finishing time induced by $h^\Sigma(\Gamma)$.

Proposition 4.3. *Let Σ be a hierarchical system. Then the home-down allocation $h^\Sigma(\Gamma)$ coincides with* $\mathrm{CES}^\Sigma(N, c_\Gamma)$.

Proof: Let $\mathscr{B} = (B_1, B_2, \ldots, B_q)$ be the ordered partition of the player set induced by ALG^* calculating $\mathrm{CES}^\Sigma(N, c_\Gamma)$. Then, we claim that $\mathscr{B}$ coincides with the ordered partition $\mathscr{A} = (A_1^*, A_2^*, \ldots, A_m^*)$ defined by the finishing times induced by the home-down allocation $h^\Sigma(\Gamma)$.

Recall that B_1 is the maximal set that minimizes the weighted average cost of a coalition in the lowest layer S_k. In particular, this means that the weighted average cost of the corresponding trunk is minimal. If these costs are 0, then B_1 corresponds to the set of players that are already connected to the root in the sense of Algorithm 4.1. By construction of the ordered partition it must hold $B_1 = A_1^*$. Now consider the case that the average cost of connecting B_1 to the root is not 0. Then by (P_3) the coalition B_1 also constitutes A_1^*, the set of all players that are the first to complete all necessary arcs to the root. So we have by property (P_2) that for all $i \in B_1 = A_1^*$,

$$h_i^\Sigma(\Gamma) = \frac{\omega_i}{\omega(A_1^*)} c(T_{A_1^*}) = \frac{\omega_i}{\omega(B_1)} c(T_{B_1}) = \frac{\omega_i}{\omega(B_1)} c_\Gamma(B_1) = \mathrm{CES}_i^\Sigma(N, c_\Gamma).$$

Now suppose that $A_\ell^* = B_\ell$ for $\ell = 1, 2, \ldots, i-1$. We show that $A_i^* = B_i$. Put $B(i) := \bigcup_{t \le i} B_t$ for all $i \in \{1, 2, \ldots, q\}$ and $B(0) = \emptyset$. It is easily seen from the definition of the weighted constrained egalitarian solution that for all $j \in B_i$ it holds

$$\mathrm{CES}_j^\Sigma(N, c_\Gamma) = \frac{\omega_j}{\omega(B_i)} (c_\Gamma(B(i)) - c_\Gamma(B(i-1))).$$

Thus we have

$$\mathrm{CES}_j^\Sigma(N, c_\Gamma) = \frac{\omega_j}{\omega(B_i)} (c(T_{B(i)}) - c(T_{B(i-1)})) = \frac{\omega_j}{\omega(B_i)} c(T_{B_i} \backslash T_{B(i-1)}). \quad (12)$$

Let p be such that $A_i^* = A_p$, i.e. the set of players finished at stage p. By (P_1) these players all belong to the same layer in the hierarchy, say S_t. Then by (11) and (10) we have

$$A_i^* \in \underset{S \subseteq S_t \backslash A(p-1)}{\operatorname{argmin}} \frac{c(T_S \backslash T_{A(p-1)})}{\omega(S)}. \quad (13)$$

Furthermore, since the right hand set is closed under union it holds that A_i^* is the unique largest set with this property. But then by our induction hypothesis A_i^* equals the unique largest set in

$$\underset{S \subseteq S_i \setminus B(i-1)}{\text{argmin}} \frac{c(T_S \setminus T_{B(i-1)})}{\omega(S)}.$$

But then $A_i^* = B_i$ by (12). □

Monderer *et al.* (1992) prove that by varying the hierarchical systems the corresponding weighted Shapley values constitute the core of a concave cost game. More specifically, Bjørndal *et al.* (1999) give a *constructive* proof of this fact for tree maintenance games. As we are about to show, the class of weighted constrained egalitarian solutions is just as flexible, in the sense that each core element for a tree maintenance game is a weighted constrained egalitarian solution for an appropriate hierarchical system.

Theorem 4.4. *The core of the game* (N, c_Γ) *equals the set of all weighted constrained egalitarian allocations* $\{\text{CES}^\Sigma(N, c_\Gamma) \mid \Sigma \text{ is a hierarchical system}\}$.

Proof: Firstly, according to Algorithm 4.1 and Proposition 3.4 we have

$$\{h^\Sigma(\Gamma) \mid \Sigma \text{ is a hierarchical system}\} \subseteq core(N, c_\Gamma).$$

Next, we will show the inverse inclusion. Let x be a core element of $core(N, c_\Gamma)$. We construct a hierarchical system such that the corresponding home down allocation $h^\Sigma(\Gamma)$ equals x. Let $\mathcal{T}(x) = \{G^1, \dots, G^p\}$ be the partition into pseudo subtrees induced by x (Proposition 3.8), such that for each $k = 1, 2, \dots, p$ we have $G^k = (S_k \cup \{r_k\}, E_k)$. Let Γ^j be the maintenance problem out of Γ that is induced by G^j, for all $j \in \{1, 2, \dots, p\}$. In addition, without loss of generality, we assume that the sets $S_1, \dots, S_p$ are indexed such that $j > j^*$ if there is a player $i \in S_j$ such that $S_{j^*} \subseteq F(i)$.

Partition each S_j into $S_j^+ := \{i \in S_j \mid x_i > 0\}$ and $S_j^0 := S_j \setminus S_j^+$. For the moment assume that the derived sets S_j^+ and S_j^0 are all nonempty. Then let $\mathcal{S}^+ = (S_1^0, S_1^+, S_2^0, S_2^+, \dots, S_p^0, S_p^+)$ be the new ordered partition of the player set N. Define $\omega \in \mathbb{R}_{++}^N$ by

$$\omega_i = \begin{cases} x_i & \text{if } x_i > 0, \\ 1 & \text{otherwise.} \end{cases}$$

We will show that for $\Sigma = (\mathcal{S}^+, \omega)$ we have $h^\Sigma(\Gamma) = x$.

First of all, we claim that for all $j \in \{1, 2, \dots, p\}$ the zero contributors in layer j are connected to the root via the players in S_j^+. More specifically, the minimal trunk in G^j containing all the players S_j^+, $\tilde{T}_{S_j^+}$, contains all the players in S_j^0. Suppose that this is not the case. Then $S_j^0 \setminus \tilde{T}_{S_j^+} \neq \emptyset$, and for all $i \in S_j^0 \setminus \tilde{T}_{S_j^+}$ it holds that $c(e_i) = 0$. Suppose, on the contrary, that $c(e_i) > 0$ for some $i \in S_j^0 \setminus \tilde{T}_{S_j^+}$. Then by the combination of the fact that $x_{S_j} \in core(S_j, c_{\Gamma^j})$ (Proposition 3.8 (ii)) and nonnegativity of costs we obtain the desired contradiction:

$$x(S_j^+) = x(S_j) = c(E_j) = c(\tilde{T}_{S_j^+}) + \sum_{i \in S_j^0 \setminus \tilde{T}_{S_j^+}} c(e_i) > c(\tilde{T}_{S_j^+}) \geq x(S_j^+).$$

So we may assume that $c(e_i) = 0$ for all $i \in S_j^0 \setminus \tilde{T}_{S_j^+}$. Define the pseudo subtree G_0^j that is induced by the set of players $T_{S_j^+}$, and for $i \in S_j^0 \setminus \tilde{T}_{S_j^+}$ let G_i^j be the pseudo subtree $(\{\pi(i), i\}, \{e_i\})$. Then $\bigcup_{i \in S_j^0 \setminus \tilde{S}_j^+} G_i^j \cup \{G^1, G^2, \ldots, G^{j-1}, G_0^j, G^{j+1}, \ldots, G^p\}$ defines a partition of G into pseudo subtrees that is induced by x. Moreover, the restriction of x to the corresponding player sets of each of these pseudo subtrees specifies a core element for the corresponding tree maintenance game. But this contradicts the fact that $\mathscr{T}(x)$ is the finest partition with this property (see Proposition 3.8 (ii)). This proves our claim.

The remaining part of the proof depends on an induction argument. According to Algorithm 4.1 the first active layer becomes S_p^+. Then the individual players get connected to the root simultaneously. Since, if this were not the case, the first set of players to finish induces an autonomous trunk that is smaller than that induced by the players in S_p. Using an argument that is similar to that above we construct a finer partition into pseudo subtrees than $\mathscr{T}(x)$, such that x is in the core of each of the underlying tree maintenance games. This leads to contradiction with the fact that $\mathscr{T}(x)$ is actually *the* finest partition into subtrees.

By the above claim we see that S_t^0 gets connected at the same time as the players in S_t^+, which means that the latter set of players are responsible for completing all the arcs in $E_p = \bigcup_{i \in S_p} e_i$, so $x(S_p^+) = c(S_p) = x(S_p)$. But then by definition it holds that

$$h_i^{\Sigma}(\Gamma) = \frac{x_i}{x(S_p^+)} c(T_{S_p^+}) = \frac{x_i}{x(S_p)} c(S_p) = x_i \quad \text{for all } i \in S_p^+.$$

Now we distinguish between two cases:

(i) Suppose that $h_i^{\Sigma}(\Gamma) = x_i$ for all $i \in \{S_t^+, S_{t+1}, S_{t+2}, \ldots, S_p\}$. Then $h_i^{\Sigma}(\Gamma) = 0$ for all $i \in S_t^0$, since by the above claim we find the players of S_t^0 connected if the former layer, S_t^+, establishes their necessary connections. Then of course $h_i^{\Sigma}(\Gamma) = 0$.

(ii) Suppose that $h^{\Sigma}(\Gamma) = x_i$ for all $i \in \{S_{t+1}, S_{t+2}, \ldots, S_p\}$. We will show that $h_i^{\Sigma}(\Gamma) = x_i$ for all $i \in S_t^+$. By our indexation we have that the players in $\bigcup_{j \geq t+1} S_j$ constitute a trunk. Moreover, by our induction hypothesis this trunk is autonomous, which means that all arcs in this component are constructed after completion of S_{t+1}^+. By our indexation and the fact that the players in S_j^0 get connected by the players in S_j^+, we have that the players in S_t^+ need to construct precisely the arcs in E_t. Claim: the players in S_t^+ finalize the necessary connecting arcs simultaneously. If this is not the case, then consider the first set $S \subseteq S_t^+$ that gets connected. Then S induces a decomposition of G^t into pseudo subtrees. As a result we obtain a partition into pseudo subtrees induced by x that is finer than $\{G^1, G^2, \ldots, G^p\}$, which gives the desired contradiction. So by our above claim we obtain for all $i \in S_t^+$,

$$h_i^{\Sigma}(\Gamma) = \frac{x_i}{x(S_t^+)} c(E_t) = \frac{x_i}{x(S_t)} c(S_t) = x_i,$$

where the second equality follows by the fact that $x(S_t^+) = x(S_t)$. Thus also $h^{\Sigma}_{S_t}(\Gamma) = x_{S_t}$.

Suppose that some S_j^+'s or S_j^0's are empty. Then remove these sets from the ordered partition. For the remaining partition essentially the above reasoning applies. □

Suppose $x \in core(N, c_\Gamma)$, and $\Sigma = (\mathscr{S}, \omega)$ is a hierarchical system such that $\mathrm{CES}^{\Sigma}(N, c_\Gamma) = x$. Then it is easy to find other hierarchical systems such that the corresponding weighted constrained egalitarian solution equals x. To see this, consider the following. Let $(A_1^*, A_2^*, \ldots, A_m^*)$ be the (ordered) partition of the player set defined by the finishing times induced by $h^{\Sigma}(\Gamma)$, and choose a nondecreasing row of positive numbers $\lambda_1, \lambda_2, \ldots, \lambda_m$, i.e. $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m > 0$. Define a new 'scaled' hierarchical system Σ^λ by the partition $\mathscr{S}$ corresponding to Σ and putting $\omega^\lambda_{A_j^*} := \lambda_j \omega_{A_j^*}$ for all $j = 1, 2, \ldots, m$. This means that for each $j, k \in \{1, 2, \ldots, m\}$ the painting speeds of members in A_j^* relative to those in A_k^* increases if $j \leq k$. An easy induction argument can be given to show that the new hierarchical system Σ^λ induces the same ordered partition of the player set defined by the finishing times. Then by (P_2) it follows that $h^{\Sigma}(\Gamma) = h^{\Sigma^\lambda}(\Gamma)$, thus $\mathrm{CES}^{\Sigma}(N, c_\Gamma) = \mathrm{CES}^{\Gamma^\lambda}(N, c_\Gamma)$. This indicates that for the weighted constrained egalitarian solution essentially only the relative weights are important.

5. The constrained egalitarian maintenance cost sharing mechanism

In this section we consider the class of maintenance problems corresponding to a fixed set of agents N and a fixed tree network $G = (V, E)$. In contrast with the previous sections we will focus on the *class of* tree maintenance problems induced by the *class* $\mathscr{C}$ of all cost functions $c : E \to \mathbb{R}_+$, rather than the single instances of tree maintenance problems. Suppose that for each single instance of a tree maintenance problem a vector of cost shares is determined as the solution of the cost sharing problem. Then a maintenance cost sharing mechanism is the device that summarizes this information by relating each cost structure $c \in \mathscr{C}$ to the solution of the corresponding tree maintenance problem. An example of a maintenance cost sharing mechanism is the *constrained egalitarian maintenance cost sharing mechanism*, denoted μ^{E}, which relates each cost function $c \in \mathscr{C}$ to the constrained egalitarian solution for the game (N, c_Γ), where Γ is given by (G, c, N). More formally, the notion of a maintenance cost sharing mechanism reads as follows.

Definition. A *maintenance cost sharing mechanism* is a mapping $\mu : \mathscr{C} \to \mathbb{R}^N_+$, relating each cost function $c \in \mathscr{C}$ to a vector of cost shares $\mu(c) \in \mathbb{R}^N_+$.

Below, the focus is on maintenance cost sharing mechanisms that are monotonic, i.e. basically those mechanisms that promote the idea that a cost reduction for the different arcs in E should result in a lowering of all the individual cost shares. More formally, a maintenance cost sharing mechanism $\mu : \mathscr{C} \to \mathbb{R}^N_+$ is *cost monotonic* if for $c, \bar{c} \in \mathscr{C}$ such that $c(e) \leq \bar{c}(e)$ for all $e \in E$ it holds $\mu(c) \leq \mu(\bar{c})$. Cost monotonicity can be seen as a most intuitive and compelling solidarity requirement.

Moreover, we consider those maintenance cost sharing mechanisms μ that are consistent with the promotion of individual interests in the sense that each vector of cost shares satisfies the core constraints for the corresponding cooperative tree maintenance game. A maintenance cost sharing mechanism μ such that $\mu(c) \in core(N, c_\Gamma)$ for all $c \in \mathscr{C}$, and $\Gamma = (G, c, N)$ is said to have the *core property*.

In our setting of tree maintenance problems, the combination of cost monotonicity and the core property is not demanding at all; there is a continuum of maintenance cost sharing mechanisms with the two properties. In this respect we refer to the continuum of maintenance cost sharing mechanisms that is induced, for instance, by the class of weighted Shapley values for tree maintenance games (see Kalai and Samet (1988) or Bjørndal *et al.* (1999)). Also the egalitarian maintenance cost sharing mechanism satisfies both properties. For the mentioned induced maintenance cost sharing mechanism the core property holds by the concavity of the tree maintenance game. Earlier we showed that the constrained egalitarian solution for any tree maintenance game specifies a core element, which implies that μ^{E} satisfies the core property. Moreover, by the dynamic approach of the former section we easily obtain μ^{E} as a cost monotonic maintenance cost sharing mechanism. As we are about to show, within this class of maintenance cost sharing mechanism, μ^{E} is the unique mechanism that minimizes the range of cost shares.

Let $(T_1, \ldots, T_p)$ be the *ordered* partition of the player set N associated to the core element $\mu^{\mathrm{E}}(c)$ induced by Algorithm 4.1. Let $t(i)$ be the number such that $i \in T_{A(t(i))}$, i.e. the minimal trunk containing the players that finish at stage $t(i)$ of the algorithm. Then conditions (P_1), (P_2) and (P_3) imply the following statements:

$$\mu_i^{\mathrm{E}}(c) \le \mu_j^{\mathrm{E}}(c) \Leftrightarrow t(i) \le t(j) \tag{14}$$

$$\sum_{i \in T(k)} \mu_i^{\mathrm{E}}(c) = \sum_{i \in T(k)} c(e_i) \quad \text{for all } k = 1, \ldots, p \tag{15}$$

where $T(k)$ is the trunk of G defined as $\bigcup_{\ell=1}^{k} T_\ell$.

The *range* of a vector $x \in \mathbb{R}_+^N$, denoted $range(x)$, is given by the interval

$$\left[\min_{i \in N} x_i, \max_{i \in N} x_i\right].$$

The next lemma shows that for any maintenance cost sharing mechanism for a tree maintenance problem satisfying cost monotonicity and the core property the range of the cost shares is at least the one according to the constrained egalitarian maintenance cost sharing mechanism.

Proposition 5.1. *If μ is a maintenance cost sharing mechanism satisfying cost monotonicity and the core property, then $range(\mu(c)) \supseteq range(\mu^{\mathrm{E}}(c))$ for all $c \in \mathscr{C}$.*

Proof: The constrained egalitarian maintenance cost sharing mechanism minimizes the range of cost shares among the mechanisms satisfying cost monotonicity and the core property. To prove this, let μ be any cost share mechanism satisfying these properties. We assert that for all $c \in \mathscr{C}$ the following inequalities are satisfied:

$$\max\{\mu_i(c) \mid i \in N\} \geq \max\{\mu_i^{\mathrm{E}}(c) \mid i \in N\}, \tag{16}$$

$$\min\{\mu_i(c) \mid i \in N\} \leq \min\{\mu_i^{\mathrm{E}}(c) \mid i \in N\}. \tag{17}$$

Suppose that inequality (16) is not satisfied. Then condition (14) implies

$$\max\{\mu_j(c) \mid j \in T_p\} < \mu_i^{\mathrm{E}}(c) \quad \text{for all } i \in T_p.$$

Therefore, it follows from efficiency and condition (15) that

$$\sum_{i \in T(p-1)} c(e_i) = \sum_{i \in T(p-1)} \mu_i^{\mathrm{E}}(c) < \sum_{i \in T(p-1)} \mu_i(c), \tag{18}$$

where $T(p-1)$ is the trunk $\bigcup_{\ell=1}^{p-1} T_\ell$, contradicting the core property. A similar reasoning gives inequality (17). □

The rest of this section is devoted to explore the full power of cost monotonicity and the core property if they are combined with the egalitarian approach of minimizing the range of cost shares.

Firstly, we will prove the theorem that if a cost sharing mechanism minimizes the range of cost shares given the restrictions imposed by cost monotonicity and the core property, then it has to be the egalitarian maintenance cost sharing mechanism. Therefore, it can be considered to strengthen Proposition 5.1.

Assume that the individual players are endowed with preferences over the set of possible cost allocations, such that the utility of an agent equals minus his cost share. We will show that within the context of cost monotonicity and the core property the egalitarian maintenance cost sharing mechanism uniquely maximizes Rawlsian welfare, that is measured by the opposite of the highest cost share. The proof will be similar to that of Theorem 5.4.

Before getting more formal, we will sketch the proof of Theorem 5.4. Consider a tree maintenance problem $\Gamma = (G, c, N)$, and let $(T_1, T_2, \ldots, T_p)$ be the partition of the player set induced by $\mu^{\mathrm{E}}(c)$. First we show that if μ is a maintenance cost sharing mechanism that minimizes the range of cost shares subject to cost monotonicity and the core property, then $\mu_i(c) = \mu_i^{\mathrm{E}}(c)$ for all $i \in T_p$. The proof of Theorem 5.4 is completed using a backward induction argument: we show that if $\mu_i(c) = \mu_i^{\mathrm{E}}(c)$ for all $i \in T_t, T_{t+1}, \ldots, T_p$, then $\mu_i(c) = \mu_i^{\mathrm{E}}(c)$ for all $i \in T_{t-1}$ as well. We will need to construct a particular tree maintenance problem $\Gamma^{t-1} = (G, c^{t-1}, N)$ such that (i) $c^{t-1} \leq c$, (ii) $\mu_i(c) = \mu_i^{\mathrm{E}}(c^{t-1})$ for all $i \in \bigcup_{k \leq t-1} T_k$, and (iii) $\mu_i(c^{t-1}) = \mu_i^{\mathrm{E}}(c^{t-1})$ for all $i \in \bigcup_{k \geq t-1} T_k$. Whereas (i) and (ii) will turn out to be obvious, (iii) is more demanding and results from Lemma 5.2 and Lemma 5.3. Finally, recalling properties (i), (ii), and (iii), cost monotonicity is invoked to finish the proof of the necessary induction step.

First we will introduce the new cost functions c^k and corresponding tree maintenance problems Γ^k. Given the partition of the player set $(T_1, T_2, \ldots, T_p)$ and $k \in \{1, 2, \ldots, p\}$ denote the *per capita* cost of T_k by $\varepsilon_k = \dfrac{c(T_k)}{|T_k|}$. Note that by definition of the partition $(T_1, T_2, \ldots, T_p)$ we have

$$\varepsilon_1 < \varepsilon_2 < \cdots < \varepsilon_p. \tag{19}$$

Nextly, for each $k \in \{1, 2, \ldots, p\}$ we define a new cost function $c^k \in \mathscr{C}$ by

$$c^k(e_i) = \begin{cases} c(e_i) & \text{for } i \in T_\ell,\ \ell = 1, 2, \ldots, k, \\ \dfrac{\varepsilon_k}{\varepsilon_\ell} c(e_i) & \text{for } i \in T_\ell,\ \ell \geq k+1. \end{cases} \tag{20}$$

Under the new cost structure c^k, by (19), the arcs identified with players in T_ℓ for $\ell \geq k+1$ are (weakly) lowered, such that for each set T_j for $j \geq k$ the corresponding average cost becomes ε_k. The other arcs are as costly as before, by definition. The new cost structure c^k induces a new tree maintenance problem which is denoted Γ^k.

Lemma 5.2. *For $k \in \{1, 2, \ldots, p\}$, $c \in \mathscr{C}$ define $h \in \mathbb{R}^N$ by*

$$h_i = \begin{cases} \mu_i^E(c) & \text{for } i \in T_\ell,\ \ell = 1, 2, \ldots, k, \\ \varepsilon_k & \text{for } i \in T_\ell,\ \ell \geq k+1. \end{cases} \tag{21}$$

Then $h \in core(N, c_{\Gamma^k})$.

Proof: Firstly, note that $h_i \geq 0$ for all $i \in N$. Moreover, h specifies a vector of cost shares. In order to see this note that for $\ell > k$ we have

$$\sum_{i \in T_\ell} \frac{\varepsilon_k}{\varepsilon_\ell} c(e_i) = \sum_{i \in T_\ell} \varepsilon_k \frac{|T_\ell|}{c(T_\ell)} c(e_i) = \varepsilon_k |T_\ell|.$$

Therefore

$$\begin{aligned} c_{\Gamma^k}(N) &= \sum_{i \in N} c^k(e_i) \\ &= \sum_{\ell \leq k} \sum_{i \in T_\ell} c(e_i) + \sum_{\ell > k} \sum_{i \in T_\ell} \frac{\varepsilon_k}{\varepsilon_\ell} c(e_i) \\ &= \sum_{\ell \leq k} \sum_{i \in T_\ell} c(e_i) + \sum_{\ell > k} |T_\ell| \varepsilon_k \\ &= \sum_{\ell \leq k} \sum_{i \in T_\ell} \mu_i^{\mathrm{E}}(c) + \sum_{\ell > k} \sum_{i \in T_\ell} h_i = \sum_{i \in N} h_i, \end{aligned}$$

where the second equality follows from condition (15). In addition, as the result of Proposition 3.1, it is enough to prove that $\sum_{i \in T} h_i \leq \sum_{i \in T} c^k(e_i)$ for all trunks T. Let T be any trunk of G, then we are left two cases:

Case (i): If $T \subseteq \bigcup_{\ell \leq k} T_\ell$, then by definition $c(e_i) = c^k(e_i)$ for all $i \in T$, hence $\sum_{i \in T} h_i = \sum_{i \in T} \mu_i^{\mathrm{E}}(c) \leq \sum_{i \in T} c(e_i) = \sum_{i \in T} c^k(e_i)$. Here the inequality is due to the fact that $\mu^{\mathrm{E}}(c) \in core(N, c_\Gamma)$. Case (ii): $T \cap (\bigcup_{\ell > k} T_\ell) \neq \varnothing$. Let $T = T' \cup T''$, where $T' = T \cap (\bigcup_{\ell \leq k} T_\ell)$ and $T'' = T \backslash T'$. Then due to $\mu^{\mathrm{E}}(c) \in core(N, c_\Gamma)$ we get

$$h(T') = \sum_{i \in T'} \mu_i^{\mathrm{E}}(c) \le c_\Gamma(T') = c(T') = c^k(T'). \tag{22}$$

Next, for all T_ℓ with $\ell > k$ we have, by construction of the partition induced by $\mu^{\mathrm{E}}(c)$, that

$$c(T'' \cap T_\ell) \ge |T'' \cap T_\ell| \varepsilon_\ell. \tag{23}$$

Then by combining (21) and (23) we obtain

$$h(T'') = |T''|\varepsilon_k = \sum_{\ell > k} |T'' \cap T_\ell| \varepsilon_k$$

$$\le \sum_{\ell > k} \frac{\varepsilon_k}{\varepsilon_\ell} c(T'' \cap T_\ell) = \sum_{\ell > k} \sum_{i \in T'' \cap T_\ell} c^k(e_i). \tag{24}$$

Hence $h(T'') \le c^k(T'')$. Then use this inequality together with (22) to see that $h(T) = h(T') + h(T'') \le c^k(T') + c^k(T'') = c^k(T)$, as desired. □

Lemma 5.3. *The partition of the player set induced by h (see (21)) equals* $(T_1, T_2, \ldots, T_{k-1}, \bigcup_{\ell \ge k} T_\ell)$, *and thus* $h = \mu^{\mathrm{E}}(c^k)$.

Proof: We will show that the partition generated by Algorithm 4.1 for calculating the constrained egalitarian solution of Γ^k coincides with the ordered partition $(T_1, T_2, \ldots, T_{k-1}, \bigcup_{\ell \ge k} T_\ell)$. Let T be a trunk of G, then

$$c^k(T) \ge h(T) \ge \sum_{i \in T} \frac{c(T_1)}{|T_1|} = |T| \cdot \frac{c^k(T_1)}{|T_1|},$$

where the first inequality follows from the fact of h being a core element of (N, c_{Γ^k}). The second inequality holds again by the fact that the average costs $\frac{c(T_j)}{|T_j|}$ are non-decreasing in j. Therefore, the average cost (with respect to c^k) of T_1 is minimum. The trunk T_1 induces a partition into pseudo subtrees $G^1, \ldots, G^\ell$, one for each outgoing arc of T_1. In other words, for each $i \in \{1, 2, \ldots, \ell\}$, there is an outgoing arc $e = (j, j')$ of T_1 such that $G^i = (V_e \cup \{j\}, E_e)$ where $(V_e, E_e) = B_e$, the branch rooted at e, and where j stands for the root. Now repeated application of the previous reasoning to each of the above pseudo subtrees yields the claim (observe that $h^1 = (h_i)_{i \in N \setminus T_1}$ is a core element for each of the induced problems as was shown in the proof of Proposition 3.8). This proves our claim. □

Theorem 5.4. *For tree maintenance problems, the constrained egalitarian cost sharing mechanism is the unique maintenance cost sharing mechanism which minimizes the range of the cost shares among those mechanisms satisfying cost monotonicity and the core property.*

Proof: By Proposition 5.1 we need only to establish uniqueness. Let μ be any maintenance cost sharing mechanism satisfying cost monotonicity and the core property such that $range(\mu(c)) = range(\mu^{\mathrm{E}}(c))$ for all $c \in \mathscr{C}$. We will prove that

$\mu_i(c) = \mu_i^{\mathrm{E}}(c)$ for all $i \in T_k$ and for all $k = 1, \ldots, p$ by backward induction on the index k.

If $range(\mu(c)) = range(\mu^{\mathrm{E}}(c))$, then in view of inequalities (16) and (17) it holds $\max\{\mu_i(c) \mid i \in N\} = \max\{\mu_i^{\mathrm{E}}(c) \mid i \in N\}$. Therefore, $\mu_j(c) \le \mu_j^{\mathrm{E}}(c)$ for all $j \in T_p$, the set of players that contribute up to the last stage of the algorithm. Now, suppose that $\mu_i(c) < \mu_i^{\mathrm{E}}(c)$ for some $i \in T_p$. Then

$$\sum_{j \in T(p-1)} c(e_j) < \sum_{j \in T(p-1)} \mu_j(c), \tag{25}$$

which contradicts the core property. Thus, $\mu_i(c) = \mu_i^{\mathrm{E}}(c)$ for all $i \in T_p$. So, μ and μ^{E} coincide on the last element of the partition.

Suppose that $\mu_i(c) = \mu_i^{\mathrm{E}}(c)$ for all $i \in T_\ell$, for all $\ell = t, t+1, \ldots, p$. We show that $\mu_i(c) = \mu_i^{\mathrm{E}}(c)$ for all $i \in T_{t-1}$.

Recall the definition of the cost structure c^{t-1} and corresponding maintenance problem Γ^{t-1} as in (20). It holds

(i) $c^{t-1} \le c$,
(ii) $\mu_i^{\mathrm{E}}(c) = \mu_i^{\mathrm{E}}(c^{t-1})$ for $i \in T_\ell$ and $\ell \le t-1$,
(iii) $\mu_i(c^{t-1}) = \mu_i^{\mathrm{E}}(c^{t-1})$ for $i \in T_\ell$ and $\ell \ge t-1$.

Observe that condition (iii) follows from Lemma 5.3, stating that the partition of the player set associated to $\mu^{\mathrm{E}}(c^{t-1})$ generated by Algorithm 4.1 is given by $(T_1, \ldots, T_{t-2}, \bigcup_{\ell \ge t-1} T_\ell)$. At an earlier stage we concluded that on the last element of the partition of the player set induced by μ^{E}, both mechanisms μ and μ^{E} coincide, which means in this situation that $\mu_i(c^{t-1}) = \mu_i^{\mathrm{E}}(c^{t-1})$ for all $i \in \bigcup_{\ell \ge t-1} T_\ell$. Thus we derive

$$\sum_{\ell \ge t} \sum_{i \in T_\ell} \mu_i(c^{t-1}) = \sum_{\ell \ge t} \sum_{i \in T_\ell} \mu_i^{\mathrm{E}}(c^{t-1}), \quad \text{and therefore}$$

$$\sum_{\ell=1}^{t-1} \sum_{i \in T_\ell} \mu_i(c^{t-1}) = \sum_{\ell=1}^{t-1} \sum_{i \in T_\ell} \mu_i^{\mathrm{E}}(c^{t-1}) = \sum_{\ell=1}^{t-1} \sum_{i \in T_\ell} \mu_i^{\mathrm{E}}(c). \tag{26}$$

The induction hypothesis gives

$$\sum_{\ell \ge t} \sum_{i \in T_\ell} \mu_i(c) = \sum_{\ell \ge t} \sum_{i \in T_\ell} \mu_i^{\mathrm{E}}(c).$$

So it holds that

$$\sum_{i \in T(t-1)} \mu_i(c) = \sum_{i \in T(t-1)} \mu_i^{\mathrm{E}}(c). \tag{27}$$

By cost monotonicity we have $\mu_i(c) \ge \mu_i(c^{t-1})$ for all i, and for $i \in T(t-1)$ in particular. Moreover, on $T(t-1)$ we even have equality. Suppose, on the contrary, that $\mu_i(c) > \mu_i(c^{t-1})$ for some $i \in T(t-1)$. Then (26) implies $\sum_{i \in T(t-1)} \mu_i^{\mathrm{E}}(c) < \sum_{i \in T(t-1)} \mu_i(c)$, which contradicts (27). Therefore, we have

$\mu_i(c) = \mu_i(c^{t-1})$ for all $i \in T(t-1)$. Then conditions (i) and (ii) imply $\mu_i(c) = \mu_i^{\mathrm{E}}(c)$ for all $i \in T_{t-1}$, which completes the necessary induction step. □

Suppose that the individual players are endowed with preferences over the set of possible cost allocations, such that the utility of an agent equals minus his cost share. Taking these utilities into account, among all maintenance cost sharing mechanisms satisfying cost monotonicity and the core property the constrained egalitarian cost sharing mechanism can be considered as the most egalitarian in the following sense. Within this class of maintenance problems it is the mechanism that uniquely minimizes the highest cost share. So the constrained egalitarian cost sharing mechanism uniquely maximizes *Rawlsian welfare* that is measured by the lowest utility level.

Theorem 5.5. *The constrained egalitarian maintenance cost sharing mechanism is the unique maintenance cost sharing mechanism that minimizes the maximal cost share among those mechanisms that satisfy cost monotonicity and which satisfy the core property.*

Proof: The proof resembles that of Theorem 5.4 up to a high degree. First, the constrained egalitarian maintenance cost sharing mechanism selects the Lorenz maximal element in the core of a tree maintenance game (Dutta and Ray (1989)), which implies Rawlsian maximality. Suppose μ satisfies also cost monotonicity and the core property thereby maximizing Rawlsian welfare. Then of course by assumption for any $c \in \mathscr{C}$, $\max\{\mu_i(c) \mid i \in N\} = \max\{\mu_i^{\mathrm{E}}(c) \mid i \in N\}$. Now proceed along the same lines as in the second part of the proof of Theorem 5.4 in order to see that μ equals μ^{E}. □

One can trace easily the following independencies between the characterizing properties in Theorem 5.5. Splitting the total costs equally between the players gives a cost monotonic mechanism that minimizes both the range of the weighted cost shares and the maximal weighted cost share. But the allocation need not be a core element. Furthermore there are mechanisms that minimize the range of the cost shares subject to the core property but are not cost monotonic. A legitimate candidate would be the mechanism μ that coincides with the constrained egalitarian solution for all problems except for the following 4-player problem corresponding to the tree $G = (V, E)$ with $V = \{r, 1, 2, 3, 4\}$ and $E = \{(r, 1), (1, 2), (2, 3), (3, 4)\}$ such that $c_1 = 1$, $c_2 = c_3 = 2$, $c_4 = 3$ and $\mu(c)$ is given by $(1, 1\frac{1}{2}, 2\frac{1}{2}, 3)$. The mechanism that relates each cost function $c \in \mathscr{C}$ to the corresponding Shapley value for (N, c_Γ) with $\Gamma = (G, c, N)$, defines a cost monotonic mechanism for which the core property is satisfied, however it does not always minimize the range of cost shares or minimize the maximal cost share.

In fact, Theorem 5.4 and 5.5 are similar to results of Aadland and Kolpin (1998). They look at airport games and propose the *restrictive average mechanism* that coincides with our egalitarian maintenance cost sharing mechanism. Besides the fact that our results hold for a more general setting, Aadland and Kolpin needed an additional characterizing property which is satisfied by the constrained egalitarian maintenance cost sharing mechanism. The property in question is *ranking*, which requires that an agent with higher stand alone costs, should contribute (weakly) more.

6. Concluding remarks

Starting with fixed tree problems we firstly restricted our attention to standard fixed tree problems and secondly to maintenance problems. Most of the imposed restrictions are not severe, as we will explain now.

Firstly, consider the restrictions in the definition of the standard fixed tree network. The assumption (c) that the root is not occupied, can be relaxed. As Granot *et al.* (1996) pointed out, we can always add a zero-cost arc from a new unoccupied root to the original root without changing the associated cost game. Similarly, the cost game is not changed if we just leave out the leafs at which none of the players is located, so (d) is not indispensable. Also the requirement (f) that there is only one arc leaving the source is not essential for any of our results. Megiddo (1978), Granot and Huberman (1981), and Granot and Maschler (1998) show how to *decompose* the problem in that case such that the core of the game (N, c_Γ) is obtained by taking the Cartesian product of the cores of the games corresponding to the different components.

In the paper we restricted the class of standard fixed tree problems to those that are maintenance problems, i.e. the problems such that there is exactly one player per vertex. This assumption is not crucial for most of our results, due to the following observations. First of all, consider a standard fixed tree problem with empty vertices, i.e., say, for the standard fixed tree network $\Gamma = (V, E, c, N)$ it holds that there is at most one player in each vertex and $|N| < |V \backslash \{r\}|$. We construct a new network Γ^* with all occupied vertices (outside the root) as follows. Define $V^* := \{v \in V \mid N_v = \emptyset\}$, the nonempty set of empty vertices in V (outside the root). For each $v \in V^*$ we create an additional artificial player i_v located at v, and let N^* be the set of all the newly introduced players, i.e. $N^* := \{i_v \mid v \in V^*\}$. Consider the corresponding standard fixed tree network for the new situation, that is $\Gamma^* := (V, E, c, N \cup N^*)$. The relation between the cores of the corresponding standard fixed tree games (N, c_Γ) and $(N \cup N^*, c_{\Gamma^*})$ is shown in the next proposition.

Proposition 6.1. *Let x be a vector of cost shares for Γ^*. Then the following two statements are equivalent:*

(i) $x \in core(N \cup N^*, c_{\Gamma^*})$ *and* $x_{N^*} = 0$,

(ii) $x_N \in core(N, c_\Gamma)$.

Proof: Straightforward and left to the reader. □

We may invoke Proposition 6.1 to obtain results regarding the core of the initial standard fixed tree game (N, c_Γ) by restricting the core of the larger game to the elements that specify zero payments for each of the artificial players.

Now assume that each vertex outside the root is occupied with at least one player. We will show how to construct a related maintenance problem $\tilde{\Gamma}$ inducing the same cost game. Let $v \in V \backslash \{r\}$ be a vertex with $|N_v| = k > 1$. Create $k - 1$ new vertices $v_1, v_2, \ldots, v_{k-1}$ and redistribute the k players in N_v such that in the new situation there is exactly one player in each of the vertices $v, v_1, v_2, \ldots, v_{k-1}$. Call the new set of vertices $\tilde{V}$. Next, delete the arc $(\pi(v), v)$ from E and add the following $k - 1$ new ones, $(\pi(v), v_1), (v_1, v_2), \ldots,$

(v_{k-2}, v_{k-1}), (v_{k-1}, v), resulting in a new set of arcs $\tilde{E}$. Basically, we keep the cost structure intact by defining $\tilde{c} : \tilde{E} \to \mathbb{R}_+$ through

$$\tilde{c}(e) = \begin{cases} c(e) & \text{if } e \in E \cap \tilde{E}, \\ c((\pi(v), v) & \text{if } e = (\pi(v), v_1), \\ 0 & \text{otherwise.} \end{cases}$$

Then $\tilde{\Gamma} = (\tilde{V}, \tilde{E}, \tilde{c}, N)$ defines a maintenance problem with the property that $c_\Gamma(S) = c_{\tilde{\Gamma}}(S)$ for all $S \subseteq N$. This procedure may be repeated as long as there are still vertices in the new network with more than one player. Finally we obtain a network $\Gamma^* = (V^*, E^*, c^*, N)$ with the property that $|N_v| = 1$ for all $v \in V^* \backslash \{r\}$ and $c_{\Gamma^*}(S) = c_\Gamma(S)$ for all $S \subseteq N$. Then, trivially, by studying the related cost games we can not distinguish between the two cases, Γ and Γ^*.

References

Aadland D, Kolpin V (1998) Shared irrigation cost: an empirical and axiomatical analysis. Mathematical Social Sciences 849:203–218

Bergantino A, Coppejans L (1997) A game theoretic approach to the allocation of joint costs in a maritime environment: a case study. Occasional Papers 44, Department of Maritime Studies and International Transport, University of Wales, Cardiff, Wales, U.K.

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

Bjørndal E, Koster M, Tijs SH (1999) Weighted allocation rules for standard fixed tree games. Mimeo, Tilburg University, Tilburg, The Netherlands

Dragan I, Potters J, Tijs SH (1989) Superadditivity for solutions of coalitional games. Libertas Matematica 9:101–110

Driessen T (1988) Cooperative Games, Solutions and Applications. Series C: Game Theory, Mathematical Programming and Operations Research. Kluwer Academic Publishers, Dordrecht, The Netherlands

Dubey P (1982) The shapley value as aircraft landing fees revisited. Management Science 28:869–874

Dutta B, Ray D (1989) A concept of egalitarianism under participation constraints. Econometrica 57:615–635

Ebert U (1999) Using equivalent income of equivalent adults to rank income distributions. Social Choice and Welfare 16:233–258

Gellekom A, Potters J (1999) Consistent rules for standard tree enterprises. Report 9919, Department of Mathematics, University of Nijmegen, Nijmegen, The Netherlands

Granot D, Huberman G (1981) Minimum cost spanning tree games. Mathematical Programming 21:1–18

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

Granot D, Maschler M (1998) Spanning network games. International Journal of Game Theory 27:467–500

Granot D, Maschler M, Owen G, Zhu WR (1996) The kernel /nucleolus of a standard fixed tree game. International Journal of Game Theory 25:219–244

Ichiishi T (1981) Super-modularity: applications to convex games and to the greedy algorithm for lp. Journal of Economic Theory 25:283–286

Kalai E, Samet D (1987) On weighted shapley values. International Journal of Game Theory 16:205–222

Kalai E, Samet D (1988) Weighted shapley values. In Roth AE (ed.) The Shapley Value: Essays in Honour of Lloyd S. Shapley. Cambridge University Press, New York, U.S.A.

Koster M (2002) Hierarchical constrained egalitarianism in TU-games. Mathematical Social Sciences 43:251–265

Koster M, Reijnierse H, Voorneveld M (2000) Voluntary contribution to multiple public projects. Forthcoming in Journal of Public Economic Theory
Littlechild S (1974) A simple expression for the nucleolus in a special case. International Journal of Game Theory 3:21–30
Littlechild S, Thompson G (1977) Aircraft landing fees: a game theory approach. The Bell Journal of Economics 8:186–204
Maschler M, Potters J, Reijnierse H (1995) Monotonicity properties of the nucleolus of standard tree games. Report 9556, Department of Mathematics, University of Nijmegen, Nijmegen, The Netherlands
Megiddo N (1978) Computational complexity of the game theory approach to cost allocation for a tree. Mathematics of Operations Research 3:189–196
Monderer D, Samet D, Shapley LS (1992) Weighted values and the core. International Journal of Game Theory 21:27–39
Moulin H, Shenker S (1996) Strategyproof sharing of submodular access costs: budget balance versus efficiency. Mimeo, Duke University, Durham, North Carolina, U.S.A.
Potters J, Sudhölter P (1999) Airport problems and consistent allocation rules. Mathematical Social Sciences 38:83–102
Shapley L (1981) Discussant's comments. In Moriarity S (ed.) Joint Cost Allocation, Tulsa, Oklahoma, U.S.A. Oklahoma Press
Shapley LS (1971) Cores of convex games. International Journal of Game Theory 1:11–26
Young HP (1998) Cost allocation, demand revelation, and core implementation. Mathematical Social Sciences 36:213–229