DEPARTMENT OF MATHEMATICS
UNIVERSITY OF NIJMEGEN The Netherlands

Consistent solution rules for standard tree enterprises

J.R.G. van Gellekom and J.A.M. Potters

Report No. 9919 (April 1999)

DEPARTMENT OF MATHEMATICS
UNIVERSITY OF NIJMEGEN
Toernooiveld
6525 ED NIJMEGEN
The Netherlands

Consistent solution rules for standard tree enterprises

J.R.G. van Gellekom and J.A.M. Potters

Abstract

This paper studies solution concepts for the problem of cost sharing on a fixed tree with costs on the edges. A list of properties of solution rules is introduced of which the most important one is 'ν-consistency'. A one-parameter family σ^α ($\alpha \in [0,1]$) of solution rules satisfying 'ν-consistency' is characterized. For every tree, a related TU-game is defined. It appears that the reduced situation w.r.t. 'ν-consistency' coincides with the reduced game introduced by Davis-Maschler (1965). The solution rule σ^α coincides with the nucleolus if $\alpha = 1$ and with the constrained egalitarian solution introduced by Dutta and Ray (1989) if $\alpha = 0$. The rules σ^α are core selectors for all $\alpha \in [0,1]$ and they satisfy population monotonicity.

1 Introduction

Consider the following situation: a network of cables connects a number of villages with a central supplier. Some villages are directly connected to the supplier, other via other villages. The positions of the cables are fixed and the cables have maintenance costs which have to be divided among the inhabitants of the villages. The question is how to divide them in a 'fair' way. This situation can be modeled with a 'standard tree enterprise': the villages are represented by the nodes of the tree, the cables by the edges and the central supplier is situated in the root.

Several solution concepts for the problem of cost allocation on a fixed tree have been studied in the literature (Megiddo (1978), Galil (1980), Granot et al. (1996)) and also the special case that the tree is a chain (airport problem, Littlechild (1974), Littlechild and Thompson (1977), Dubey (1982)). In most papers solution rules are not defined on a tree, but on the related 'tree game' (N, C), a concave game, in which $C(S)$ is equal to the minimal total costs of the edges necessary to connect all members of S with the root.

Potters and Sudhölter (1995) have studied a consistency property of solution rules, 'ν-consistency', on the class of airport problems. The idea of ν-consistency is the following: suppose that inhabitant i leaves after paying his amount σ_i according to a solution rule σ. The costs of the tree are decreased by subtracting this amount from the costs on the edges of the (unique) path from the village where i lives towards the root of the tree. First the costs of the edge closest to the village where i lives are decreased until σ_i is subtracted or the costs have become 0. Then the costs of the next edge on the path to the root are decreased and so on. The solution rule σ

satisfies ν-consistency if the remaining players have to pay the same amount before and after the departure of i. In this paper we study a one-parameter family σ^{α} ($\alpha \in [0,1]$) of solution rules defined on the class of standard tree enterprises, which all satisfy 'ν-consistency'. It appears that the standard tree game related to the reduced standard tree enterprise coincides with the Davis-Maschler reduced game (1965) of the standard tree game related to the original tree.

The one-parameter family σ^{α} coincides, for some α, with well-known solution concepts for standard tree games. For $\alpha = 1$ the solution rule σ^{α} coincides with the nucleolus of the associated standard tree game (Megiddo (1978)) and for $\alpha = 0$ it coincides with the constrained egalitarian solution introduced by Dutta and Ray (1989). For all $\alpha \in [0,1]$, σ^{α} is an element of the core of the associated standard tree game. We show by examples that the Shapley value and the τ-value do not satisfy ν-consistency, so they do not belong to this one-parameter family. Sönmez (1994) has shown that the nucleolus is population monotonic on the class of airport games, but this is not the case on the whole class of concave games. We show that the rules σ^{α} are population monotonic on the class of standard tree games for all $\alpha \in [0,1]$. In particular the nucleolus is population monotonic on the whole class of standard tree games (cf. Maschler et al. (1995)).

Section 2 repeats the concept of standard tree enterprise. Section 3 introduces properties of solution rules and in Section 4 the solution rules σ^{α} are defined. Section 5 studies the solution rules σ^{α}, both w.r.t. the properties of Section 3 and w.r.t. solution rules on standard tree games. Section 6 shows that the solution rules σ^{α} are the only solution rules on the class of standard tree enterprises satisfying all properties defined in Section 3.

2 Preliminaries

Definition 2.1: A *standard tree enterprise* $\Gamma := ((V,E), \boldsymbol{r}, c, (N_{\boldsymbol{p}})_{\boldsymbol{p} \in V})$ consists of the following components:

- (V,E) is an undirected tree (a *tree* is a connected graph without cycles) with node set (or vertex set) V and edge set E. The tree describes the network of cables. V is non-empty and finite,
- $\boldsymbol{r} \in V$ is a special node called the *root* (the central supplier is situated here),
- $c : E \to \mathbb{R}_+$ is a cost function on the edges of the tree (maintenance costs),
- for every $\boldsymbol{p} \in V$ there is a finite (possibly empty) set $N_{\boldsymbol{p}}$ (the inhabitants of village $\boldsymbol{p}$) with the property $N_{\boldsymbol{p}} \cap N_{\boldsymbol{q}} = \emptyset \ \forall \boldsymbol{p}, \boldsymbol{q} \in V, \boldsymbol{p} \neq \boldsymbol{q}$.

◇

Figure 2.1 gives an example of a standard tree enterprise. Since (V,E) is a tree there is, for every $\boldsymbol{p} \in V$, exactly one path $\boldsymbol{p}_0 = \boldsymbol{r}, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_t = \boldsymbol{p}$ from the root to $\boldsymbol{p}$, such that $\{\boldsymbol{p}_\ell, \boldsymbol{p}_{\ell+1}\} \in E$ for all ℓ, with the convention that the path consists of $\boldsymbol{r}$ only if $\boldsymbol{p}$=$\boldsymbol{r}$. Define the *predecessor* of $\boldsymbol{p} \in V \backslash \{\boldsymbol{r}\}$ by $\pi(\boldsymbol{p}) := \boldsymbol{p}_{t-1}$. The edge $\{\pi(\boldsymbol{p}), \boldsymbol{p}\}$ is denoted by $e_{\boldsymbol{p}}$ and the costs of this edge by $c_{\boldsymbol{p}}$, hence $c_{\boldsymbol{p}} := \mathrm{c}(\{\pi(\boldsymbol{p}),\boldsymbol{p}\})$. For practical use we define $c_{\boldsymbol{r}} := 0$. Note that the map π and the set of nodes V together determine the structure of the tree.

We define a partial ordering on the set of nodes V by:

$$\boldsymbol{p} \preceq \boldsymbol{q} \Longleftrightarrow \text{ the (unique) path from the root } \boldsymbol{r} \text{ to } \boldsymbol{q} \text{ contains } \boldsymbol{p}.$$

A *trunk* of (V, E) is a non-empty set of nodes $T \subseteq V$ with the property: if $\boldsymbol{p} \in T$ and $\boldsymbol{q} \preceq \boldsymbol{p}$ then also $\boldsymbol{q} \in T$. In particular every trunk contains the root $\boldsymbol{r}$. Let $\boldsymbol{p} \in V$. The *branch of* $\boldsymbol{p}$, $B_{\boldsymbol{p}}$, is the following subset of V: $B_{\boldsymbol{p}} := \{\boldsymbol{q} \in V \mid \boldsymbol{p} \preceq \boldsymbol{q}\}$.

For $U \subseteq V$ define $N(U) := \bigcup_{\boldsymbol{p} \in U} N_{\boldsymbol{p}}$, the set of *players* in U (we use this term instead of 'inhabitants') and $n(U) := |N(U)|$, the number of players in U. Further let $N := N(V)$ and $n := |N|$, the set and the number of players in the tree respectively. If there is no node $\boldsymbol{q} \in V$ such that $\pi(\boldsymbol{q}) = \boldsymbol{p}$ then $\boldsymbol{p}$ is called a *leaf.* If, in addition, $n_{\boldsymbol{p}} = 1$ then $\boldsymbol{p}$ is called a *lonely leaf.* (We write $n_{\boldsymbol{p}}$ instead of $n(\{\boldsymbol{p}\})$.) The costs of an arbitrary subset U of nodes are defined by $c(U) := \sum_{\boldsymbol{p} \in U} c_{\boldsymbol{p}}$ and the costs of the standard tree enterprise by $c(\Gamma) := c(V)$. The problem we consider is how to share $c(\Gamma)$ among the players. This is in general not immediately clear, as the edges are used by different players.

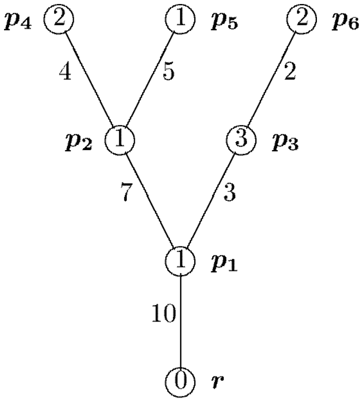


Figure 2.1: *Example of a standard tree enterprise. The numbers in the nodes denote the numbers of inhabitants. Some nodes can be ordered w.r.t.* $\preceq$, *e.g.* $\boldsymbol{r} \preceq \boldsymbol{p_1} \preceq \boldsymbol{p_2} \preceq \boldsymbol{p_5}$, *but e.g.* $\boldsymbol{p_2}$ *and* $\boldsymbol{p_3}$ *cannot be compared. The nodes* $\boldsymbol{p_4}$, $\boldsymbol{p_5}$ *and* $\boldsymbol{p_6}$ *are leaves and* $\boldsymbol{p_5}$ *is the only lonely leaf. The tree can also be described by the map* π: $\pi(\boldsymbol{p_4}) = \pi(\boldsymbol{p_5}) = \boldsymbol{p_2}, \pi(\boldsymbol{p_6}) = \boldsymbol{p_3}, \pi(\boldsymbol{p_2}) = \pi(\boldsymbol{p_3}) = \boldsymbol{p_1}$ *and* $\pi(\boldsymbol{p_1}) = \boldsymbol{r}$. *The set of nodes* $T = \{\boldsymbol{r}, \boldsymbol{p_1}, \boldsymbol{p_2}, \boldsymbol{p_3}, \boldsymbol{p_5}\}$ *is an example of a trunk and* $B_{\boldsymbol{p_2}} = \{\boldsymbol{p_2}, \boldsymbol{p_4}, \boldsymbol{p_5}\}$ *is the branch of* $\boldsymbol{p_2}$.

We only consider the following subset of standard tree enterprises:

$$\mathcal{T} := \{\Gamma \mid c_{\boldsymbol{p}} \geq 0 \ \forall \boldsymbol{p} \in V, \ n(V) \geq 1 \text{ and } \forall \boldsymbol{p} \in V : [n(B_{\boldsymbol{p}}) = 0 \Rightarrow c(B_{\boldsymbol{p}}) = 0]\}.$$

The subset of $\mathcal{T}$ with n players is denoted by $\mathcal{T}(n)$. The set $\mathcal{T}$ contains standard tree enterprises with non-negative costs, such that the costs of every branch without players is 0. Note that empty nodes are allowed and in particular empty leaves. In addition, it is allowed to have players in the root and to have more than one outgoing edge of the root. Let $\boldsymbol{p} \in V$ with $\pi(\boldsymbol{p}) = \boldsymbol{r}$. The standard tree enterprise with node set $B_{\boldsymbol{p}} \cup \{\boldsymbol{r}\}$ is called a *component* of Γ. Usually we call elements of $\mathcal{T}$ 'trees' for short. The set $\mathcal{T}$ is rather large. The reason that we have chosen such a large set is that we stay in this set when we perform some operations on trees, like with ν-consistency (see Section 3). The following subset of $\mathcal{T}$ contains trees with strictly positive costs on the edges, no empty leaves and no 'empty villages along the road'. In fact, this is the class in which we are interested.

$$\mathcal{T}_0 := \{\Gamma \in \mathcal{T} \mid \forall \boldsymbol{p} \in V\backslash\{\boldsymbol{r}\} : c_{\boldsymbol{p}} > 0 \text{ and } [n_{\boldsymbol{p}} = 0 \Rightarrow |\{\boldsymbol{q} \in V \mid \pi(\boldsymbol{q}) = \boldsymbol{p}\}| > 1]\}.$$

A *single valued solution rule* on $\mathcal{T}$ is a map σ which assigns to every $\Gamma \in \mathcal{T}$ a vector $\sigma(\Gamma)$ in $\mathbb{R}^N$. The value $\sigma_i(\Gamma)$ denotes the amount of money that player i has to pay. The restriction of $\sigma(\Gamma)$ to the players in $N\backslash\{i\}$ is denoted by $\sigma_{-i}(\Gamma)$. Solution rules for cost allocation on a fixed tree can be obtained by considering for every $\Gamma \in \mathcal{T}$ a corresponding cost game:

Definition 2.2: Let $\Gamma := ((V, E), \boldsymbol{r}, c, (N_{\boldsymbol{p}})_{\boldsymbol{p}\in V}) \in \mathcal{T}$. The corresponding *standard tree game* is a pair (N, C) where the cost function $C : 2^N \to \mathbb{R}$ is defined as follows: $C(S)$ is equal to the minimal total costs of the edges necessary to connect all members of S with the root. $\diamond$

The proof of the following proposition can be found in Granot et al. (1996).

Proposition 2.1: *Standard tree games are concave games.*

A solution rule on standard tree games can be considered to be a solution rule on $\mathcal{T}$. This paper initially studies solution rules on standard tree enterprises apart from standard tree games, although important relations will be mentioned. The next section introduces a number of properties of solution rules on $\mathcal{T}$. The last property, ν-consistency, is the most important one for this paper.

3 Properties of solution rules on trees

This section introduces properties of solution rules on trees. It will appear that the properties are not logically independent. Let σ be a single valued solution rule on $\mathcal{T}$. Let $\Gamma := ((V, E), \boldsymbol{r}, c, (N_{\boldsymbol{p}})_{\boldsymbol{p}\in V}) \in \mathcal{T}$.

Efficiency (*Eff*) Efficiency means that the players pay together exactly the costs of the edges: $\sum_{i\in N} \sigma_i(\Gamma) = c(\Gamma) = \sum_{\boldsymbol{p}\in V} c_{\boldsymbol{p}}$.

Contraction (*Contr*) Let $\boldsymbol{p} \in V\backslash\{\boldsymbol{r}\}$ such that $c_{\boldsymbol{p}} = 0$. Contraction of edge $e_{\boldsymbol{p}}$ means identifying nodes $\boldsymbol{p}$ and $\pi(\boldsymbol{p})$. A solution rule satisfies *Contraction* if every player has to pay the same amount before and after the contraction. Note that contraction of a zero-edge does not influence the related standard tree game. Let us give an example:

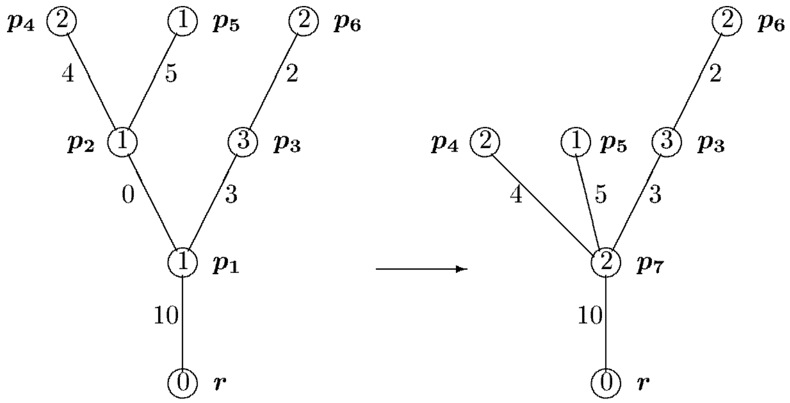


Figure 3.1: Example of a contraction: edge $e_{\boldsymbol{p_2}}$ has costs 0. Nodes $\boldsymbol{p_1}$ and $\boldsymbol{p_2}$ are contracted to a new node $\boldsymbol{p_7}$.

Deletion (*Del*) Let $\boldsymbol{p} \in V\backslash\{\boldsymbol{r}\}$ be an empty village 'along the road'. (i.e. $n_{\boldsymbol{p}} = 0$ and $|\{\boldsymbol{q} \in V \mid \pi(\boldsymbol{q}) = \boldsymbol{p}\}| = 1$). Let Γ' denote the tree obtained by deleting node $\boldsymbol{p}$, where the cost function c' is given by

$$c'_{\boldsymbol{q}} := c_{\boldsymbol{p}} + c_{\boldsymbol{q}} \text{ if } \pi(\boldsymbol{q}) = \boldsymbol{p} \text{ and } c'_{\boldsymbol{q}} := c_{\boldsymbol{q}} \text{ otherwise.}$$

The solution rule σ satisfies *Deletion* if $\sigma(\Gamma) = \sigma(\Gamma')$. Again (N, C) does not change. Every game-theoretical solution rule satisfies *Contraction* and *Deletion.* Figure 3.2 gives an example.

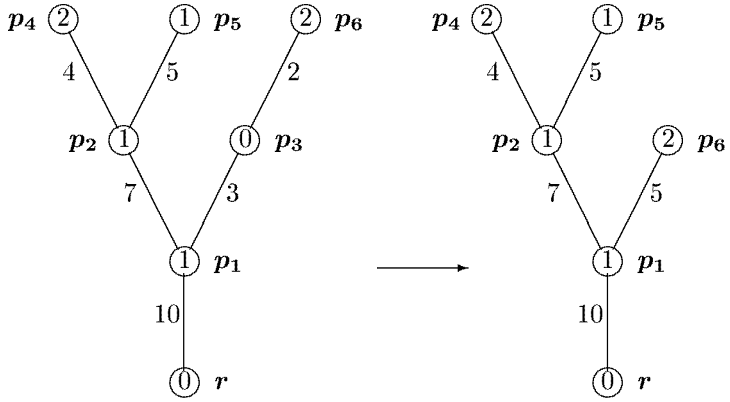


Figure 3.2: Deletion of the empty node $\boldsymbol{p_3}$.

Homogeneity (*Hom*) A solution σ is called *Homogeneous* if for every $\lambda > 0$ we have $\sigma(\bar{\Gamma}) = \lambda\sigma(\Gamma)$, where $\bar{\Gamma} := ((V, E), \boldsymbol{r}, \bar{c}, (N_{\boldsymbol{p}})_{\boldsymbol{p} \in V})$ and $\bar{c}(e) := \lambda c(e)\ \forall e \in E$. In words, if the costs of all edges are multiplied by the same constant $\lambda > 0$, then all coordinates of $\sigma(\Gamma)$ are multiplied by λ.

Reasonableness (*Reas*) The stand alone costs of player $i \in N_{\boldsymbol{p}}$ are equal to $\mathrm{sa}_i(\Gamma) := \sum_{\boldsymbol{q} \in P} c_{\boldsymbol{q}}$, where P is the path from the root $\boldsymbol{r}$ to node $\boldsymbol{p}$. Note that $\mathrm{sa}_i(\Gamma) = C(i)$. The marginal costs of i, $\mathrm{mc}_i(\Gamma)$, are equal to $\mathrm{mc}_i(\Gamma) = C(N) - C(N \backslash \{i\})$. A solution rule σ is called *Reasonable* if $\mathrm{mc}_i(\Gamma) \leq \sigma_i(\Gamma) \leq \mathrm{sa}_i(\Gamma)\ \forall i \in N$. So every player pays at least his marginal costs and at most his stand alone costs. In particular $\sigma_i(\Gamma) = 0$ if $i \in N_{\boldsymbol{r}}$ because $\mathrm{sa}_i(\Gamma) = \mathrm{mc}_i(\Gamma) = 0$ for such a player. Milnor (1952) has studied reasonable outcomes for n-person games. Figure 3.3 gives an example.

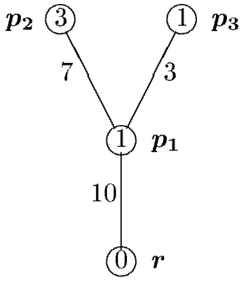


Figure 3.3: *If* $i \in N_{\boldsymbol{p_1}}$ *then* $\mathrm{mc}_i(\Gamma) = 0$ *and* $\mathrm{sa}_i(\Gamma) = 10$. *If* $i \in N_{\boldsymbol{p_2}}$ *then these amounts are equal to* 0 *and* 17 *respectively. Finally if* $i \in N_{\boldsymbol{p_3}}$ *then* $\mathrm{mc}_i(\Gamma) = 3$, $\mathrm{sa}_i(\Gamma) = 13$.

Fair ranking (*FR*) Fair ranking says that players living closer to the root pay weakly less than players who live farther away. More precisely, σ satisfies *Fair ranking* if

$$\boldsymbol{p} \preceq \boldsymbol{q}, i \in N_{\boldsymbol{p}}, j \in N_{\boldsymbol{q}} \Longrightarrow \sigma_i(\Gamma) \leq \sigma_j(\Gamma).$$

In particular two players living in the same village pay the same (take $\boldsymbol{q} = \boldsymbol{p}$). So *Fair ranking* implies 'equal treatment of equals'.

Cost-monotonicity (*CostMon*) This property says that players do not pay less, when the costs of an edge are increased. Let $\boldsymbol{p} \in V \backslash \{\boldsymbol{r}\}$ such that $n(B_{\boldsymbol{p}}) > 0$. Let $\Gamma' := (V, E, \boldsymbol{r}, c', (N_{\boldsymbol{p}})_{\boldsymbol{p} \in V}) \in \mathcal{T}$ be the standard tree enterprise obtained by increasing the costs of edge $e_{\boldsymbol{p}}$ by $\delta > 0$, i.e. $c'_{\boldsymbol{p}} := c_{\boldsymbol{p}} + \delta$ and $c'_{\boldsymbol{q}} := c_{\boldsymbol{q}}$ otherwise. A solution σ satisfies *Cost-monotonicity* if $\sigma_i(\Gamma') \geq \sigma_i(\Gamma)$ for all $i \in N$.

Population-monotonicity (*PopMon*) This property has to do with the departure of a player without paying. Nobody pays less in the reduced situation. We have to be careful with defining the reduced situation, because if e.g. the player who leaves is a lonely leaf with positive marginal costs, then just deleting this player from its village gives a tree which is not an element of $\mathcal{T}$. If a player leaves then the edges which are not used by the remaining players are no longer needed. Therefore we do not only delete a player i from node $\boldsymbol{p}$, but we also delete the part of the path from $\boldsymbol{p}$ to the root $\boldsymbol{r}$ which is used by player i only. In other words, if the marginal costs of player i are 0, then no edges are deleted, otherwise the edges who determine the marginal costs are deleted. In this way the reduced tree Γ' is an element of $\mathcal{T}$. For example if the player in $\boldsymbol{p_6}$ of Figure 3.2 leaves then both edge $e_{\boldsymbol{p_6}}$ and $e_{\boldsymbol{p_3}}$ are deleted. The solution rule σ satisfies *Population-monotonicity* if $\sigma_j(\Gamma) \leq \sigma_j(\Gamma')$ for all $i, j \in N$, where Γ' as described above and $n \geq 2$.

ν-Consistency (*ν-Cons*) In the literature several notions of consistency on the class of TU-games have been studied. The central idea is that a solution rule assigns to a 'reduced problem' with less players the restriction of the outcome of the original problem. The question is how to define the 'reduced problem'. For trees there are different possibilities. In this paper we study one of the possibilities. Let $n \geq 2$, $\boldsymbol{p} \in V, i \in N_{\boldsymbol{p}}, x := \sigma_i(\Gamma)$. Suppose that player i pays x and leaves. In the reduced situation the remaining maintenance costs $c(\Gamma) - x$ have to be shared among the other players. We want the reduced situation to be a tree in $\mathcal{T}$ which differs from the original one only w.r.t. player i and the costs. To get a tree with costs $c(\Gamma) - x$ the amount x has to be subtracted somewhere from the costs of the edges. As i only uses the edges of the path P from $\boldsymbol{p}$ to the root $\boldsymbol{r}$ it seems natural to subtract x from the costs of P; i pays only for edges he uses. In the case of ν-consistency the costs are subtracted in the following way: decrease the costs of the edges of the path P, starting in $\boldsymbol{p}$ and going towards the root $\boldsymbol{r}$. First the costs of the edge closest to the village where i lives are decreased until x is subtracted or the costs have become 0. Then the costs of the next edge on the path to the root are decreased and so on. Figure 3.4 gives an example.

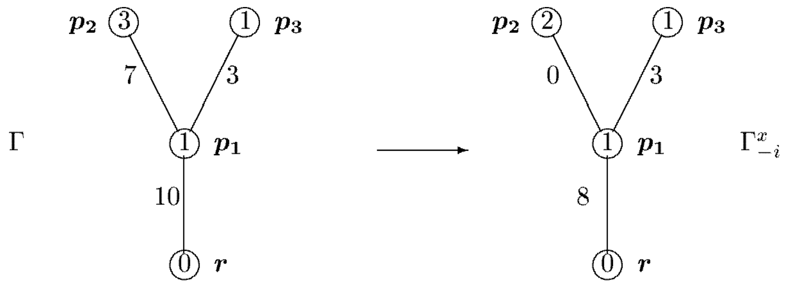


Figure 3.4: Example of ν-consistency: suppose that one of the players in $\boldsymbol{p_2}$, say i, leaves and that $x := \sigma_i^{\alpha}(\Gamma) = 9$. Then we first subtract 7 from the costs of $e_{\boldsymbol{p_2}}$, which becomes 0. The remaining 2 are subtracted from the costs of edge $e_{\boldsymbol{p_1}}$.

Subtracting costs in this way may cause problems, namely if $x < \mathrm{mc}_i(\Gamma)$ or $x > \mathrm{sa}_i(\Gamma)$. Therefore we assume for *ν-Consistency* that σ satisfies *Reasonableness.*

The reduced standard tree enterprise with respect to player i is denoted by Γ^x_{-i} where $x := \sigma_i(\Gamma)$. *Reasonableness* of σ implies that $\Gamma^x_{-i} \in \mathcal{T}$. A solution σ satisfies *ν-Consistency* if $\forall i \in N$: $\sigma(\Gamma^x_{-i}) = \sigma_{-i}(\Gamma)$, where $x := \sigma_i(\Gamma)$. Note that the reduced tree with respect to a player in the root is obtained by just deleting this player. Let (N, C) and $(\bar{N}, \bar{C})$ be the standard tree game associated with Γ and Γ^x_{-i} respectively. It is not difficult to show that

$$\overline{C}(S) = \min\{C(S), C(S \cup \{i\}) - x_i\}.$$

So the standard tree game associated with the reduced tree is exactly the Davis-Maschler reduced game of the game associated with the original tree. The term 'ν-consistency' is introduced in Potters and Sudhölter (1995). The ν has been chosen because the nucleolus satisfies this kind of consistency. They also study another kind of consistency on airport games, which they call '*μ-consistency*' and which is satisfied by the 'modified nucleolus', introduced by Sudhölter (1997). In the case of *μ-consistency*, costs are subtracted from the path from the root to the node where a player lives, starting in the root. It can be shown that there is no solution on $\mathcal{T}$ satisfying *μ-consistency*, *Reas* and *FR*.

The properties we introduced are not logically independent. We have e.g.

Proposition 3.1: *Let σ be a solution rule on $\mathcal{T}$.*

a *If σ satisfies Reas and ν-Cons then σ also satisfies Eff.*

b *If σ satisfies Contr, Reas, ν-Cons and CostMon, then σ also satisfies PopMon.*

Proof:

a The proof is by induction to the number of players. Suppose that σ satisfies *Reas* and *ν-Cons*. If $n = 1$ then $c(\Gamma)$ is equal to the costs of the path from the root $\boldsymbol{r}$ to the node where the player lives, i.e. $c(\Gamma) = \mathrm{sa}_1(\Gamma) = \mathrm{mc}_1(\Gamma)$. By *Reas* we have $\mathrm{mc}_1(\Gamma) \leq \sigma_1(\Gamma) \leq \mathrm{sa}_1(\Gamma)$. So $\sigma_1(\Gamma) = c(\Gamma)$ and efficiency follows.

Now suppose that σ is efficient if the number of players is at most $n-1$ ($n \geq 2$). Take a tree $\Gamma \in \mathcal{T}(n)$. Choose a player, say i, and let $x := \sigma_i(\Gamma)$. Applying *ν-Cons* with respect to player i gives a tree Γ^x_{-i} with $n-1$ players. The induction hypothesis together with *ν-Cons* gives

$$\sum_{j \in N \backslash i} \sigma_j(\Gamma) = \sum_{j \in N \backslash i} \sigma_j(\Gamma^x_{-i}) = c(\Gamma^x_{-i}) = c(\Gamma) - x = c(\Gamma) - \sigma_i(\Gamma),$$

from which it follows that σ is efficient.

b Suppose that σ satisfies *Contr*, *Reas*, *ν-Cons* and *CostMon*. Let $\Gamma \in \mathcal{T}$, $i \in N$, $n \geq 2$. Player i and some edges, say E', are deleted giving a tree Γ'. Note that $E' = \emptyset$ if $\mathrm{mc}_i(\Gamma) = 0$. Applying *$\nu$-Cons* w.r.t. player i gives a tree Γ^x_{-i}. Let Γ'' be the tree obtained by deleting E' from Γ^x_{-i}. Then *CostMon*, *Contr* and *ν-Cons* give that for all $j \in N \backslash \{i\}$: $\sigma_j(\Gamma') \geq \sigma_j(\Gamma'') = \sigma_j(\Gamma^x_{-i}) = \sigma_j(\Gamma)$. □

The next proposition says that a solution rule on $\mathcal{T}$ can be uniquely extended if we know it for standard tree enterprises with 1 or 2 players and if it satisfies *Reas*, *ν-Cons* and *CostMon*. This is important for the characterization in Section 6.

Proposition 3.2: *Let σ, τ be two solution rules on $\mathcal{T}$. If σ and τ both satisfy Reas, ν-Cons, CostMon and if $\sigma = \tau$ if $n \leq 2$, then $\sigma = \tau$.*

Proof: Suppose that σ and τ both satisfy *Reas*, *ν-Cons*, *CostMon*, that $\sigma = \tau$ if $n \leq 2$ and that $\sigma \neq \tau$. Let Γ be a tree with a minimal number of players for which $\sigma \neq \tau$. Then $n \geq 3$.

Let $i \in N$ such that $\sigma_i(\Gamma) > \tau_i(\Gamma)$ (i exists because σ and τ satisfy *Eff* by Lemma 3.1). Then from *ν-Cons*, *CostMon* and the definition of Γ we get

$$\sigma_{-i}(\Gamma) = \sigma(\Gamma_{-i}^{\sigma_i(\Gamma)}) \leq \sigma(\Gamma_{-i}^{\tau_i(\Gamma)}) = \tau(\Gamma_{-i}^{\tau_i(\Gamma)}) = \tau_{-i}(\Gamma). \tag{3.1}$$

Now take a $j \in N\backslash\{i\}$. Again *ν-Cons* and *CostMon* and the definition of Γ give

$$\sigma_{-j}(\Gamma) = \sigma(\Gamma_{-j}^{\sigma_j(\Gamma)}) \geq \sigma(\Gamma_{-j}^{\tau_j(\Gamma)}) = \tau(\Gamma_{-j}^{\tau_j(\Gamma)}) = \tau_{-j}(\Gamma). \tag{3.2}$$

Next let $k \in N\backslash\{i, j\}$. Then $\sigma_k(\Gamma) = \tau_k(\Gamma)$ by (3.1) and (3.2). So by *ν-Cons* and the definition of Γ we have (using $x := \sigma_k(\Gamma) = \tau_k(\Gamma)$)

$$\sigma_{-k}(\Gamma) = \sigma(\Gamma_{-k}^{x}) = \tau(\Gamma_{-k}^{x}) = \tau_{-k}(\Gamma),$$

from which it follows that $\sigma(\Gamma) = \tau(\Gamma)$, a contradiction. □

We now want to answer the question 'Can we find solution rules satisfying the properties of Section 3?'. We start with a sub-question 'Can we find solution rules satisfying *ν-Cons*?'. This is not immediately clear. The next section gives a one-parameter family σ^α, $\alpha \in [0, 1]$ of solution rules on $\mathcal{T}$ which all satisfy *ν-Cons.* In Section 5 we show that σ^α satisfies all properties introduced in Section 3 for all $\alpha \in [0, 1]$.

4 Definition of the solution rules $\boldsymbol{\sigma}^{\alpha}$

For every $\alpha \in [0, 1]$ we define a solution rule on $\mathcal{T}$. The computation of σ^α consists of computing trunks with minimal 'weights'. We start - more general - with introducing weights for connected parts of a tree.

4.1 Preliminaries

Let $\alpha \in [0, 1]$, $\Gamma \in \mathcal{T}$. We only compute weights in trees without players in the root. Therefore we assume in this subsection that $N_{\boldsymbol{r}} = \emptyset$. Let $Q \subseteq V$ be a nonempty connected set of nodes, $Q \neq \{\boldsymbol{r}\}$. The *degree* of Q, $d(Q)$, is the number of 'outgoing' edges of Q used by at least one player and not starting in the root:

$$d(Q) := |\{\boldsymbol{p} \in V\backslash Q \mid \pi(\boldsymbol{p}) \in Q\backslash\{\boldsymbol{r}\}, N(B_{\boldsymbol{p}}) \neq \emptyset\}|.$$

The number $i(Q)$ counts the 'incoming' edges of Q used by at least one player and not starting in the root. If $Q \cup \{\boldsymbol{r}\}$ is connected or Q is contained in a branch without players then $i(Q) = 0$; otherwise $i(Q) = 1$. Consider for example the left tree of Figure 3.1. For $Q_1 = \{\boldsymbol{p_2}, \boldsymbol{p_4}, \boldsymbol{p_5}\}$ we have $d(Q_1) = 0$, $i(Q_1) = 1$. For $Q_2 = \{\boldsymbol{p_1}, \boldsymbol{p_3}\}$ these numbers are 2 and 0 respectively. The *grade* of Q is defined by

$$g^\alpha(Q) := n(Q) + \alpha(d(Q) - i(Q)).$$

Then $g^\alpha(Q) \geq 0$, because assuming that $g^\alpha(Q) < 0$ implies $i(Q) = 1$, $d(Q) = 0$ and $n(Q) = 0$ which is impossible (because $n(Q) = d(Q) = 0$ implies $i(Q) = 0$). Note that, for Q_1 and Q_2 as defined above, we have $g^\alpha(Q_1 \cup Q_2) = 8 + \alpha(1 - 0) = 4 + \alpha(0 - 1) + 4 + \alpha(2 - 0) = g^\alpha(Q_1) + g^\alpha(Q_2)$. This property always holds if $Q_1 \cup Q_2$ is connected and $Q_1 \cap Q_2 \subseteq \{\boldsymbol{r}\}$, because the number of players is additive and there is at most one edge which connects Q_1 and Q_2, which cancels.

The *weight* of Q is defined by

$$w^\alpha(Q) := \begin{cases} \frac{c(Q)}{g^\alpha(Q)} & \text{if } g^\alpha(Q) > 0 \\ \infty & \text{if } g^\alpha(Q) = 0 \text{ and } c(Q) > 0 \\ 0 & \text{if } g^\alpha(Q) = 0 \text{ and } c(Q) = 0. \end{cases}$$

We are in fact interested in pairs $(c(Q), g^\alpha(Q))$, but we use the function w^α to order the pairs in $\mathbb{R}_+ \cup \{\infty\}$. Therefore it is not allowed to simplify the fractions.

The following lemma will be used often in Section 5, in particular when Q_2 is a trunk:

Lemma 4.1: *Let Q_1 and Q_2 be connected subsets of V with $Q_1 \cap Q_2 \subseteq \{\boldsymbol{r}\}$ and such that $Q_1 \cup Q_2$ is again connected. Then*

$$w^\alpha(Q_1) \leq w^\alpha(Q_2) \Longleftrightarrow w^\alpha(Q_1) \leq w^\alpha(Q_1 \cup Q_2) \leq w^\alpha(Q_2)$$

and

$$w^\alpha(Q_1) < w^\alpha(Q_2) \Longleftrightarrow w^\alpha(Q_1) < w^\alpha(Q_1 \cup Q_2) \leq w^\alpha(Q_2).$$

Proof: The proof is straightforward and mainly based on the following relation:

$$\min\left\{\frac{a}{b}, \frac{c}{d}\right\} \leq \frac{a+c}{b+d} \leq \max\left\{\frac{a}{b}, \frac{c}{d}\right\} \quad b, d > 0.$$

□

Now let T be a trunk. We have $g^\alpha(T) = n(T) + \alpha d(T) \geq 0$ and $g^\alpha(T) = 0$ implies $n(T) = 0$ and $\alpha d(T) = 0$. As there is at least one player outside the root, there is always a trunk for which $g^\alpha(T) > 0$, i.e. with finite weight. Therefore

$$w^\alpha_{\mathrm{m}}(\Gamma) := \min\{w^\alpha(T) \mid T \text{ trunk of } \Gamma, T \neq \{\boldsymbol{r}\}\}$$

is finite. If α is fixed then we omit it in the notations.

Consider the tree in Figure 2.1. For $\alpha = 1$ we have

$$w_{\mathrm{m}}(\Gamma) := \min\{\frac{10}{3}, \frac{17}{5}, \frac{20}{8}, \frac{22}{9}, \frac{21}{6}, \frac{24}{9}, \frac{26}{10}, \frac{22}{5}, \frac{25}{8}, \frac{27}{9}, \frac{26}{6}, \frac{29}{9}, \frac{31}{10}, \frac{13}{6}, \frac{15}{7}\} = \frac{15}{7}$$

and there is one trunk with minimal weight: $T^* = \{\boldsymbol{r}, \boldsymbol{p_1}, \boldsymbol{p_3}, \boldsymbol{p_6}\}$.

4.2 Definition of σ^α

Let $\alpha \in [0, 1]$. The idea of the solution rule σ^α is that villages closest to the central supplier choose some neighbouring villages with which they want to cooperate. They pay a part of the costs of the edges they use themselves and the remaining costs are paid by villages further away. The value α determines which part of the costs is 'shifted'.

We define the solution rule σ^α by an algorithm. The algorithm consists of computing repeatedly the maximal (w.r.t. the number of nodes) trunk T_{m} with minimal weight $w_{\mathrm{m}}(\Gamma)$. The players in T_{m} each have to pay $w_{\mathrm{m}}(\Gamma)$. Then these players leave and T_{m} is 'contracted to the root'. The costs of every outgoing edge of T_{m}, i.e. the edges which $d(T_{\mathrm{m}})$ counts, are increased by $\alpha w_{\mathrm{m}}(\Gamma)$. The following algorithm gives the details. We use the fact that the function π determines the tree.

Computation of σ^α
Take $\Gamma \in \mathcal{T}$, $\alpha \in [0, 1]$.

Step 1: Put $\sigma_i^\alpha(\Gamma) := 0 \quad \forall i \in N_{\boldsymbol{r}}$ and delete the players in the root.

Step 2: Compute $w_{\mathrm{m}}(\Gamma) := \min\{w(T) \mid T \text{ trunk of } \Gamma, T \neq \{\boldsymbol{r}\}\}$ and let T_{m} be the maximal trunk with minimal weight.

Step 3: Put $\sigma_i^\alpha(\Gamma) := w_{\mathrm{m}}(\Gamma)$ for all $i \in N(T_{\mathrm{m}})$;
Compute the tree where T_{m} is contracted:
$V := \boldsymbol{r'} \cup V \setminus T_{\mathrm{m}}$; $N_{\boldsymbol{r'}} = \emptyset$;
$$\pi'(\boldsymbol{p}) := \begin{cases} \boldsymbol{r'} & \text{if } \pi(\boldsymbol{p}) \in T_{\mathrm{m}}, \\ \pi(\boldsymbol{p}) & \text{otherwise;} \end{cases}$$
$c_{\boldsymbol{p}} := c_{\boldsymbol{p}} + \alpha w_{\mathrm{m}}(\Gamma)$ if $\pi(\boldsymbol{p}) \in T_{\mathrm{m}} \setminus \{\boldsymbol{r}\}$ and $N(B_{\boldsymbol{p}}) \neq \emptyset$;
$\pi := \pi'$; $\boldsymbol{r} := \boldsymbol{r'}$;
$N := N \setminus N(T_{\mathrm{m}})$;

Step 4: Repeat Steps 2 and 3 until there are no players left.

The number of trunks is finite, so there is a trunk with minimal weight, but there can be more than one trunk with minimal weight. Lemma 4.2 shows that 'the maximal trunk with minimal weight' is a correct definition. In every iteration, the number of remaining nodes strictly decreases, hence the algorithm stops after finitely many iterations.

Lemma 4.2: *Let $\alpha \in [0,1]$ be fixed.*

a *If T_1 and T_2 are both trunks with minimal weight, then the union $T_1 \cup T_2$ has also minimal weight.*

b *If T is a trunk with minimal weight and $T = T_1 \cup T_2$ with $T_1 \cap T_2 = \{\boldsymbol{r}\}$, then both T_1 and T_2 have minimal weight.*

Proof:

a Suppose that T_1 and T_2 are two trunks with minimal weight. The statement follows immediately if $T_1 \subseteq T_2$ or $T_2 \subseteq T_1$, so we suppose that this is not the case. The proof consists of applying Lemma 4.1 a number of times. Let $S := T_1 \cap T_2$. If $S = \{\boldsymbol{r}\}$ then (in particular) $w(T_1) \leq w(T_2)$, hence $w(T_1) \leq w(T_1 \cup T_2) \leq w(T_2)$ and it follows that $T_1 \cup T_2$ has also minimal weight. If $S \neq \{\boldsymbol{r}\}$ then $w(T_1 \backslash T_2) \leq w(S)$ (because if $w(S) < w(T_1 \backslash T_2)$ then $w(S) < w(S \cup (T_1 \backslash T_2)) = w(T_1)$ which gives a contradiction because S is also a trunk). So $w(T_1 \backslash T_2) \leq w((T_1 \backslash T_2) \cup S) = w(T_1) = w(T_2)$ and $w(T_1 \backslash T_2) \leq w((T_1 \backslash T_2) \cup T_2) = w(T_1 \cup T_2) \leq w(T_2)$. Therefore $w(T_1 \cup T_2) = w(T_2)$.

b Suppose that T is a trunk with minimal weight and $T = T_1 \cup T_2$ with $T_1 \cap T_2 = \{\boldsymbol{r}\}$. If $w(T_1) \neq w(T_2)$ then by Lemma 4.1 $\min\{w(T_1), w(T_2)\} < w(T)$, contradicting that T has minimal weight. So $w(T_1) = w(T_2)$ and, again by Lemma 4.1, $w(T_1) = w(T) = w(T_2)$.

□

Remark: From **b** it follows that if there is more than one edge starting in the root, then we can split the problem and compute σ^α for the components of Γ.

We shall now compute σ^α for the tree in Figure 2.1 for different values of α. We start with $\alpha = 1$. We have already seen that $T^* = \{\boldsymbol{r}, \boldsymbol{p_1}, \boldsymbol{p_3}, \boldsymbol{p_6}\}$ is the unique trunk with minimal weight $w_{\mathrm{m}}(\Gamma) = \frac{15}{7}$. So $\sigma_i^\alpha(\Gamma) = \frac{15}{7}$ for all $i \in N(T^*)$. Contraction of T^* gives the left tree of Figure 4.1. Now we find that $T^* = \{\boldsymbol{r}, \boldsymbol{p_2}\}$ is the unique trunk with minimal weight $w_{\mathrm{m}}(\Gamma) = 3\frac{1}{21}$, hence the player in $\boldsymbol{p_2}$ pays this amount. Contraction of T^* gives the right tree of Figure 4.1. Now we find immediately that $\sigma_i^\alpha(\Gamma) = 3\frac{11}{21}$ for the players in $\boldsymbol{p_4}$ and $\sigma_i^\alpha(\Gamma) = 8\frac{1}{21}$ for the player in $\boldsymbol{p_5}$. We can summarize this as $(2\frac{1}{7}, 3\frac{1}{21}, 2\frac{1}{7}, 3\frac{11}{21}, 8\frac{1}{21}, 2\frac{1}{7})$, where the i-th coordinate is the amount that a player in node $\boldsymbol{p_i}$ has to pay. In the same way we find for $\alpha = \frac{1}{2}$ the amounts $(2\frac{4}{13}, 3\frac{43}{91}, 2\frac{4}{13}, 3\frac{43}{91}, 6\frac{67}{91}, 2\frac{4}{13})$ and for $\alpha = 0$ we find $(2\frac{1}{2}, 3\frac{2}{3}, 2\frac{1}{2}, 3\frac{2}{3}, 5, 2\frac{1}{2})$.

We defined solution rules σ^α for $\alpha \in [0,1]$. In the same way we can define solution rules for $\alpha > 1$. It will appear that such rules do not satisfy all properties introduced in the previous section.

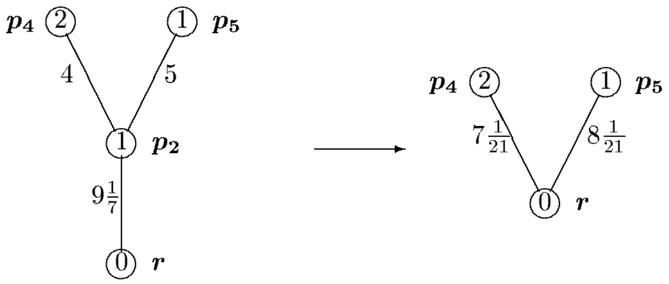


Figure 4.1: Trees in the computation of σ^1 for the tree in Figure 2.1.

5 Properties of the solution rules σ^α

This section studies two kinds of properties of the solution rules σ^α. Section 5.1 shows that σ^α satisfies the properties as introduced in Section 3. Section 5.2 studies the relationship between the solution rules σ^α ($\alpha \in [0,1]$) and well-known solution concepts for (standard tree) games.

5.1 Properties of solution rules

In the proofs we use several times two trees Γ and Γ'. The weights w.r.t. Γ (Γ') are denoted by w (w') and the maximal trunk with minimal weight is denoted by T_{m} (T'_{m}). The costs and grades w.r.t. Γ (Γ') are denoted by c and g (c' and g').

Proposition 5.1: *The solution rule σ^α satisfies Eff for all $\alpha \in [0,1]$.*

Proof: *Eff* follows from the fact that the costs of every trunk T_{m} are paid

- partly by the players in T_{m}: $\frac{n(T_{\mathrm{m}})}{n(T_{\mathrm{m}})+\alpha d(T_{\mathrm{m}})}c(T_{\mathrm{m}})$,
- partly by shifting costs: $\frac{\alpha d(T_{\mathrm{m}})}{n(T_{\mathrm{m}})+\alpha d(T_{\mathrm{m}})}c(T_{\mathrm{m}})$.

□

Proposition 5.2: *The solution rule σ^α satisfies Contr for all $\alpha \in [0,1]$.*

Proof: Let $\Gamma \in \mathcal{T}$ and let $\boldsymbol{p} \in V\backslash\{\boldsymbol{r}\}$ such that $c_{\boldsymbol{p}} = 0$. If $\boldsymbol{p}$ is the only node besides the root, then σ^α assigns 0 to all players before and after contraction. Otherwise let Γ' be the tree after contraction of edge $e_{\boldsymbol{p}}$. There is a one-to-one correspondence between trunks T' in Γ' and trunks T in Γ with $\boldsymbol{p} \in T$ or $\pi(\boldsymbol{p}) \notin T$ such that corresponding trunks have equal weights. If T is a trunk in Γ with $\pi(\boldsymbol{p}) \in T$ and $\boldsymbol{p} \notin T$, then $w(T \cup \boldsymbol{p}) \leq w(T)$ (use $w(\boldsymbol{p}) = 0$ and apply Lemma 4.1). So to find a trunk with minimal weight in Γ, it is sufficient to consider only trunks with $\boldsymbol{p} \in T$ or $\pi(\boldsymbol{p}) \notin T$. As a consequence we can contract in Γ and Γ' corresponding trunks T_{m} and T'_{m} and $w_{\mathrm{m}}(\Gamma) = w'_{\mathrm{m}}(\Gamma')$.

If $\boldsymbol{p} \in T_{\mathrm{m}}$, the trees after contraction of T_{m} and T'_{m} are the same and σ^{α} is the same for the remaining players. If $\boldsymbol{p} \notin T_{\mathrm{m}}$ then $\pi(\boldsymbol{p}) \notin T_{\mathrm{m}}$ and we have after contraction of T_{m} a tree in which edge $e_{\boldsymbol{p}}$ still has costs zero. Induction to the number of nodes completes the proof. □

Proposition 5.3: *The solution rule σ^{α} satisfies Del for all $\alpha \in [0,1]$.*

Proof: Let $\Gamma \in \mathcal{T}$ and let $\boldsymbol{p} \in V \backslash \{\boldsymbol{r}\}$ such that $n_{\boldsymbol{p}} = 0$ and $\#\{\boldsymbol{q} \in V \mid \pi(\boldsymbol{q}) = \boldsymbol{p}\} = 1$. Let $\boldsymbol{q}$ be the unique node with $\pi(\boldsymbol{q}) = \boldsymbol{p}$ and Γ' the tree obtained by deletion of $\boldsymbol{p}$. The proof is very similar to the proof of *Contr*. As σ^{α} satisfies *Contr* we may assume that the costs on the edges are strictly positive. There is a one-to-one correspondence between trunks T' in Γ' and trunks T in Γ with $\boldsymbol{q} \in T$ or $\boldsymbol{p} \notin T$ such that corresponding trunks have the same weight. If T is a trunk in Γ with $\boldsymbol{p} \in T$ and $\boldsymbol{q} \notin T$, then $w(T \backslash \{\boldsymbol{p}\}) < w(T)$ because $c_{\boldsymbol{p}} > 0$. So to find a trunk with minimal weight in Γ, it is sufficient to consider only trunks with $\boldsymbol{q} \in T$ or $\boldsymbol{p} \notin T$. As a consequence we can contract in Γ and Γ' corresponding trunks T_{m} and T'_{m} and $w_{\mathrm{m}}(\Gamma) = w'_{\mathrm{m}}(\Gamma')$.

If $\boldsymbol{q} \in T_{\mathrm{m}}$, the trees after contraction of T_{m} and T'_{m} are the same and σ^{α} is the same for the remaining players. If $\boldsymbol{q} \notin T_{\mathrm{m}}$ then $\boldsymbol{p} \notin T_{\mathrm{m}}$ and we have after contraction of T_{m} a tree in which node $\boldsymbol{p}$ can be deleted. Induction to the number of nodes completes the proof. □

Remark: as σ^{α} satisfies *Contr* and *Del* we assume from now on that $\Gamma \in \mathcal{T}_0$. In particular, costs on the edges are strictly positive and, as a consequence, weights of trunks are strictly positive.

Proposition 5.4: *The solution rule σ^{α} satisfies Hom for all $\alpha \in [0,1]$.*

Proof: If the costs of all edges are multiplied by the same $\lambda > 0$, then all weights are multiplied by λ; and multiplication by λ and taking the minimum can be interchanged, because λ is positive. So σ^{α} satisfies *Hom.* □

Proposition 5.5: *The solution rule σ^{α} satisfies FR for all $\alpha \in [0,1]$.*

Let $\boldsymbol{p}, \boldsymbol{q} \in V$, $\boldsymbol{p} \preceq \boldsymbol{q}$. The rule σ^{α} divides the set of nodes V in groups, such that σ^{α} is constant for players in nodes in the same group. These groups correspond with maximal trunks with minimal weights in subsequent iterations. If $\boldsymbol{p}$ and $\boldsymbol{q}$ are in the same group, then *FR* immediately follows. For the other case, it is sufficient to show that w_{m} increases in subsequent iterations in the computation of σ^{α}. This is done in Lemma 5.6. In addition, the lemma shows that we can, in the computation of σ^{α}, contract an arbitrary trunk with minimal weight, instead of the maximal one, without changing the output of the algorithm.

Lemma 5.6: *Let $\alpha \in [0,1]$ be fixed, $\Gamma \in \mathcal{T}_0$. Let T^* be a trunk of (V,E) with minimal weight $w_{\mathrm{m}}(\Gamma) = \frac{c(T^*)}{g(T^*)}$ and let T be an arbitrary trunk with $T^* \subset T$. Let Γ' be the tree after contraction of T^* and let $T' := T \backslash T^* \cup \{\boldsymbol{r'}\}$. Then $w'(T') \geq w_{\mathrm{m}}(\Gamma)$ and equality holds if and only if $w(T) = w_{\mathrm{m}}(\Gamma)$. As a consequence $w'_{\mathrm{m}}(\Gamma') \geq w_{\mathrm{m}}(\Gamma)$.*

Proof: Let $d := |\{\boldsymbol{p} \in T \backslash T^* \mid \pi(\boldsymbol{p}) \in T^* \backslash \{\boldsymbol{r}\}\}|$, the number of edges in T which costs are increased when T^* is contracted. The following equalities hold:

- $c'(T') = c(T) - c(T^*) + \alpha w_{\mathrm{m}}(\Gamma)d$,
- $n'(T') = n(T) - n(T^*)$,
- $d'(T') = d(T) - d(T^*) + d$.

Using

$$w_{\mathrm{m}}(\Gamma) \leq w(T) = \frac{c(T)}{g(T)} \tag{5.1}$$

we get

$$\begin{aligned} w'(T') = \frac{c(T) - c(T^*) + \alpha w_{\mathrm{m}}(\Gamma)d}{g(T) - g(T^*) + \alpha d} &\geq \frac{w_{\mathrm{m}}(\Gamma)g(T) - w_{\mathrm{m}}(\Gamma)g(T^*) + \alpha w_{\mathrm{m}}(\Gamma)d}{g(T) - g(T^*) + \alpha d} \\ &= w_{\mathrm{m}}(\Gamma). \end{aligned} \tag{5.2}$$

Equality in (5.2) holds if and only if equality in (5.1) holds, i.e. if and only if $w(T) = w_{\mathrm{m}}(\Gamma)$. An arbitrary trunk in Γ' can be seen as the result, after contraction of T^*, of a trunk in Γ which contains T^*. So $w'(T') \geq w_{\mathrm{m}}(\Gamma)$ for *all* trunks T' of Γ'. Hence $w'_{\mathrm{m}}(\Gamma') \geq w_{\mathrm{m}}(\Gamma)$. □

Proposition 5.7: *The solution rule σ^α satisfies CostMon for all $\alpha \in [0,1]$.*

Proof: Let $\alpha \in [0,1]$. We use induction to the number of nodes. The case $|V| = 2$ follows immediately. Take $\Gamma \in \mathcal{T}_0$ with $|V| > 2$. Let $\boldsymbol{p} \in V \backslash \{\boldsymbol{r}\}$ such that $n(B_{\boldsymbol{p}}) > 0$ and let $\delta > 0$. Let $\Gamma' = (V, E, \boldsymbol{r}, c', (N_{\boldsymbol{q}})_{\boldsymbol{q} \in V})$ be the tree with cost function $c'_{\boldsymbol{p}} := c_{\boldsymbol{p}} + \delta$ and $c'_{\boldsymbol{q}} := c_{\boldsymbol{q}}$ otherwise. We distinguish between two cases:

1. Suppose that there is a trunk T with $w(T) = w_{\mathrm{m}}(\Gamma)$ and $\boldsymbol{p} \notin T$. Then $w(T) \leq w(T'_{\mathrm{m}}) \leq w'(T'_{\mathrm{m}}) \leq w'(T)$. Now $\boldsymbol{p} \notin T$ implies $w(T) = w'(T)$ hence $w'(T) = w'(T'_{\mathrm{m}})$. So in both Γ and Γ' we can contract T. Then $\sigma^\alpha(\Gamma)$ and $\sigma^\alpha(\Gamma')$ coincide on $N(T)$ and the shifted amounts are equal ($\alpha w(T) = \alpha w'(T)$). Hence, by induction, we have $\sigma^\alpha(\Gamma') \geq \sigma^\alpha(\Gamma)$.

2. Suppose that $\boldsymbol{p} \in T$ for every trunk T with $w(T) = w_{\mathrm{m}}(\Gamma)$. We increase the costs of $c_{\boldsymbol{p}}$ gradually. Define for every $\varepsilon > 0$ the cost function $c^\varepsilon : V \to \mathbb{R}_+$ by: $c^\varepsilon_{\boldsymbol{p}} := c_{\boldsymbol{p}} + \varepsilon$ and $c^\varepsilon_{\boldsymbol{q}} := c_{\boldsymbol{q}}$ otherwise. Let Γ^ε be the corresponding tree and let $T^\varepsilon_{\mathrm{m}}$ be the maximal trunk of Γ^ε with minimal weight $w^\varepsilon_{\mathrm{m}}(\Gamma)$. Define $\delta_1 := \max\{t \leq \delta \mid \text{there exists a trunk } T \text{ with } w(T) = w_{\mathrm{m}}(\Gamma),\ w^t(T) = w^t_{\mathrm{m}}(\Gamma^t)\}$. There exists a trunk T_1 which has minimal weight with respect to both Γ and Γ^{δ_1}. Then $w^{\delta_1}(T_1) > w(T_1)$ and in both cases contraction of T_1 gives $\sigma^\alpha_i(\Gamma^{\delta_1}) > \sigma^\alpha_i(\Gamma)$ for all $i \in N(T_1 \backslash \{\boldsymbol{r}\})$. By induction we have $\sigma^\alpha(\Gamma^{\delta_1}) \geq \sigma^\alpha(\Gamma)$. We are done if $\delta = \delta_1$. If $\delta_1 < \delta$ then let $T^* \neq T_1$ be such that $w^{\delta_1}(T^*) = w^{\delta_1}(T_1)$. If $\boldsymbol{p} \notin T^*$ then we contract T^* in Γ^{δ_1} and we are in case 1: we can increase $c_{\boldsymbol{p}}$ arbitrarily. If $\boldsymbol{p} \in T'$ for all trunks T' with $w^{\delta_1}(T') = w^{\delta_1}(T_1)$ then we define $\delta_2 := \max\{t \leq \delta \mid \text{there exists a trunk } T \text{ with } w^{\delta_1}(T) = w^{\delta_1}_{\mathrm{m}}(\Gamma),\ w^t(T) = w^t_{\mathrm{m}}(\Gamma^t)\}$. Now we can repeat the procedure on Γ^{δ_1} and Γ^{δ_2}. This procedure ends after finitely many steps because the number of trunks is finite. □

Proposition 5.8: *The solution rule σ^α satisfies Reas for all $\alpha \in [0,1]$.*

Proof: Let $\Gamma \in \mathcal{T}_0$, $\alpha \in [0,1]$. If $i \in N_{\boldsymbol{r}}$ then, clearly, *Reas* is satisfied. First the marginal costs: take $\boldsymbol{p} \in V$ and $i \in N_{\boldsymbol{p}}$.

- If $\boldsymbol{p}$ is not a lonely leaf, then $mc_i(\Gamma) = 0 \leq \sigma_i^\alpha(\Gamma)$ because weights of trunks are nonnegative.

- If $\boldsymbol{p}$ is a lonely leaf then $mc_i(\Gamma) = c_{\boldsymbol{p}}$. If a trunk with minimal weight is contracted, then nonnegative costs are shifted. So it is sufficient to show that i pays at least his marginal costs in a tree with $\boldsymbol{p} \in T_{\text{m}}$. If $T_{\text{m}} = \{\boldsymbol{p}, \boldsymbol{r}\}$ then i pays $c_{\boldsymbol{p}}$. Now suppose that $T_{\text{m}} \neq \{\boldsymbol{p}, \boldsymbol{r}\}$ and let $T := T_{\text{m}} \backslash \{\boldsymbol{p}\}$. If $w(T_{\text{m}}) < c_{\boldsymbol{p}}$ i.e. $\frac{c(T)+c_{\boldsymbol{p}}}{n(T)+1+\alpha(d(T)-1)} < c_{\boldsymbol{p}}$ then $\frac{c(T)}{n(T)+\alpha(d(T)-1)} < c_{\boldsymbol{p}}$. So
$$w(T_{\text{m}}) = \frac{c(T)+c_{\boldsymbol{p}}}{n(T)+1+\alpha(d(T)-1)} > \frac{c(T)}{n(T)+\alpha(d(T)-1)} \geq w(T)$$
contradicting the fact that T_{m} has minimal weight. So $\sigma_i^\alpha(\Gamma) = w(T_{\text{m}}) \geq c_{\boldsymbol{p}}$.

The proof for the stand alone costs is by induction to the number of nodes. The case $|V| = 2$ follows immediately. Now suppose $|V| > 2$. We first show that $\sigma_i^\alpha(\Gamma) \leq \text{sa}_i(\Gamma)$ for all $i \in N(T_{\text{m}})$. Let $\boldsymbol{p} \in T_{\text{m}} \backslash \{\boldsymbol{r}\}, i \in N_{\boldsymbol{p}}$. Let P be the unique path from $\boldsymbol{p}$ towards the root $\boldsymbol{r}$. As P is also a trunk we have
$$\sigma_i^\alpha(\Gamma) = w(T_{\text{m}}) \leq w(P) = \frac{c(P)}{g(P)} \leq c(P) = \text{sa}_i(\Gamma). \tag{5.3}$$

Let $i \notin N(T_{\text{m}})$ and let Γ' be the tree after contraction of T_{m}. It is sufficient to show that $\text{sa}_i(\Gamma) \geq \text{sa}_i(\Gamma')$ (induction completes the proof). To prove this inequality let $\boldsymbol{q}$ be the first node in T_{m} on the path from $\boldsymbol{p}$ to $\boldsymbol{r}$ (in Γ). It is sufficient to show that $\alpha w(T_{\text{m}}) \leq c(\overline{P})$ where $\overline{P}$ is the path from the root $\boldsymbol{r}$ to node $\boldsymbol{q}$. The inequality follows immediately if $\alpha = 0$ and if $\alpha > 0$ then $g(\overline{P}) > 0$ and we can apply Formula (5.3). □

Proposition 5.9: *The solution rule σ^α satisfies ν-Cons for all $\alpha \in [0,1]$*

Let $\Gamma \in \mathcal{T}_0, \boldsymbol{p} \in V, i \in N_{\boldsymbol{p}}$ and define $x := \sigma_i^\alpha(\Gamma)$. Let Γ_{-i}^x be the reduced standard tree enterprise, i.e. i is removed and the costs of the path from $\boldsymbol{p}$ to the root $\boldsymbol{r}$ are diminished with x, starting in $\boldsymbol{p}$ (see also the definition of *ν-Cons*). We have to show that $\sigma^\alpha(\Gamma_{-i}^x) = \sigma_{-i}^\alpha(\Gamma)$. This trivially holds if $i \in N_{\boldsymbol{r}}$, so we assume that $\boldsymbol{p} \neq \boldsymbol{r}$. In addition we assume that $N_{\boldsymbol{r}} = \emptyset$ (otherwise first apply *ν-Cons* w.r.t. the players in the root), that there is exactly one node $\boldsymbol{q}$ such that $\pi(\boldsymbol{q}) = \boldsymbol{r}$ (w.l.o.g. by Lemma 4.2) and that $n \geq 2$. Let $\overline{w}$ be the weight with respect to Γ_{-i}^x, $\overline{w}_{\text{m}}$ the minimal weight and let $\overline{T}_{\text{m}}$ be the maximal trunk with minimal weight w.r.t. Γ_{-i}^x. We write w_{m} instead of $w_{\text{m}}(\Gamma)$.

We first consider the case $\boldsymbol{p} \in T_{\text{m}}$ and $g(T_{\text{m}}) = 1$. Then $\alpha = 0$ and T_{m} is a path, which costs are entirely paid by i. Then *ν-Cons* with respect to player i followed by *Contr* applied on T_{m} is exactly one iteration in the algorithm to compute $\sigma^\alpha(\Gamma)$, hence $\sigma_{-i}^\alpha(\Gamma) = \sigma^\alpha(\Gamma_{-i}^x)$.

The proof of the case $g(T_{\mathrm{m}}) > 1$ consists of a number of steps. We first show that the weight of the minimal trunk does not change: $\overline{w}_{\mathrm{m}} = w_{\mathrm{m}}$. We finish the proof that σ^{α} satisfies *ν-Cons* by induction to the number of nodes.

Lemma 5.10: $\overline{w}_{\mathrm{m}} \leq w_{\mathrm{m}}$.

Proof: We distinguish between some cases:

- Suppose that $\boldsymbol{p} \in T_{\mathrm{m}}$. Then we have $\overline{w}(T_{\mathrm{m}}) = \frac{c(T_{\mathrm{m}})-x}{g(T_{\mathrm{m}})-1}$. As $w(T_{\mathrm{m}}) = \frac{c(T_{\mathrm{m}})}{g(T_{\mathrm{m}})} = x$, we have $\overline{w}(T_{\mathrm{m}}) = w(T_{\mathrm{m}}) = w_{\mathrm{m}}$. Therefore $\overline{w}_{\mathrm{m}} \leq w_{\mathrm{m}}$.
- Suppose that $\boldsymbol{p} \notin T_{\mathrm{m}}$ and $g(P) \geq 1$, where P is the path from $\boldsymbol{p}$ to T_{m}.
  - If $x < c(P)$ then $\overline{w}_{\mathrm{m}} \leq \overline{w}(T_{\mathrm{m}}) = w(T_{\mathrm{m}}) = w_{\mathrm{m}}$.
  - If $x \geq c(P)$ then $\overline{w}_{\mathrm{m}} \leq \overline{w}(T_{\mathrm{m}} \cup P) = \frac{c(T_{\mathrm{m}})+c(P)-x}{g(T_{\mathrm{m}})+g(P)-1} \leq \frac{c(T_{\mathrm{m}})}{g(T_{\mathrm{m}})} = w(T_{\mathrm{m}}) = w_{\mathrm{m}}$.
- Suppose that $\boldsymbol{p} \notin T_{\mathrm{m}}$ and $g(P) < 1$ i.e. $n(P) + \alpha(d(P) - 1) < 1$, which is only possible if $d(P) = 0, n(P) = 1$ and $\alpha > 0$ (because $i \in N(P)$, so $n(P) \geq 1$). Then $P = \{\boldsymbol{p}\}$ and $N_{\boldsymbol{p}} = \{i\}$ because we have assumed that there are no empty leaves and no empty villages along the road. After contraction of T_{m} in Γ the costs of edge $e_{\boldsymbol{p}}$ are increased by αw_{m} and we have $x = \alpha w(T_{\mathrm{m}}) + c(P)$. Therefore $\overline{w}_{\mathrm{m}} \leq \overline{w}(T_{\mathrm{m}}) = \frac{c(T_{\mathrm{m}})-\alpha w(T_{\mathrm{m}})}{g(T_{\mathrm{m}})-\alpha} = w(T_{\mathrm{m}}) = w_{\mathrm{m}}$. (Note that there is no division by zero, because if $g(T_{\mathrm{m}}) = \alpha$, then $n(T_{\mathrm{m}}) = 0$ and $d(T_{\mathrm{m}}) = 1$. This is impossible, since $\pi(\boldsymbol{p})$ would be an empty village along the road.)

□

Let $S := T_{\mathrm{m}} \cap \overline{T}_{\mathrm{m}} \neq \{\boldsymbol{r}\}$. We write $\overline{T}_{\mathrm{m}} = S \cup \overline{Q}_1 \cup \ldots \cup \overline{Q}_l$. The sets $\overline{Q}_t$ are pairwise disjoint and each of them contains exactly one node $\overline{\boldsymbol{q}}_{\boldsymbol{t}}$ such that $\pi(\overline{\boldsymbol{q}}_{\boldsymbol{t}}) \in S$.

Lemma 5.11: *The following inequalities hold*

$$\overline{w}(\overline{Q}_t) \leq \overline{w}(\overline{T}_{\mathrm{m}}) \leq w(T_{\mathrm{m}}) < w(\overline{Q}_t).$$

Proof: Suppose that $\overline{w}(\overline{Q}_t) > \overline{w}(\overline{T}_{\mathrm{m}})$. Then $\overline{w}(\overline{Q}_t) > \overline{w}(\overline{T}_{\mathrm{m}} \backslash \overline{Q}_t)$, because if $\overline{w}(\overline{Q}_t) \leq \overline{w}(\overline{T}_{\mathrm{m}} \backslash \overline{Q}_t)$ then $\overline{w}(\overline{Q}_t) \leq \overline{w}(\overline{T}_{\mathrm{m}})$ by Lemma 4.1. Applying Lemma 4.1 again gives $\overline{w}(\overline{T}_{\mathrm{m}} \backslash \overline{Q}_t) < \overline{w}(\overline{T}_{\mathrm{m}})$, which contradicts the definition of $\overline{T}_{\mathrm{m}}$. The second inequality is exactly the previous lemma. Now suppose that $w(\overline{Q}_t) \leq w(T_{\mathrm{m}})$. Then $T_{\mathrm{m}} \cup \overline{Q}_t$ is a better candidate for T_{m} because $w(T_{\mathrm{m}} \cup \overline{Q}_t) \leq w(T_{\mathrm{m}})$ by Lemma 4.1. □

Lemma 5.12: *We have $\ell \leq 1$ and if $\ell = 1$ then $\boldsymbol{p} \in \overline{Q}_1 =: \overline{Q}$.*

Proof: If $\boldsymbol{p} \in T_{\mathrm{m}}$ we find $\overline{w}(\overline{Q}_t) = w(\overline{Q}_t)$ and therefore (by Lemma 5.11) $\ell = 0$. If $\boldsymbol{p} \notin T_{\mathrm{m}}$, then there is at most one index t with $\overline{Q}_t \cap P \neq \emptyset$, where P is the path from $\boldsymbol{p}$ to T_{m}. Therefore $\ell \leq 1$. Let us suppose that $\ell = 1$ and that $\boldsymbol{p} \notin \overline{Q}$. Let $\overline{P}$ denote the path from $\boldsymbol{p}$ to $\overline{Q}$. Note that $x > c(\overline{P})$ (otherwise $\overline{w}(\overline{Q}_t) = w(\overline{Q}_t)$). So $\overline{w}(\overline{P}) = 0$ and $\overline{T}_{\mathrm{m}} \cup \overline{P}$ is a better candidate for $\overline{T}_{\mathrm{m}}$ for the trunk is larger and $\overline{w}(\overline{T}_{\mathrm{m}} \cup \overline{P}) \leq \overline{w}(\overline{T}_{\mathrm{m}})$ by Lemma 4.1. □

We continue by distinguishing between some cases with respect to $g(\overline{Q})$. Lemma 5.13 considers three cases which can all occur. Initially the cases of Lemma 5.14 cannot be excluded, but they all give a contradiction.

Lemma 5.13:

1. *If* $\ell = 0$ *then* $w_{\mathrm{m}} = \overline{w}_{\mathrm{m}}$.
2. *If* $\ell = 1$ *and* $g(\overline{Q}) < 1$ *then* $w_{\mathrm{m}} = \overline{w}_{\mathrm{m}}$.
3. *If* $\ell = 1$, $\alpha = 0$ *and* $n(\overline{Q}) = 1$ *(i.e.* $g(\overline{Q}) = 1$*) then* $w_{\mathrm{m}} = \overline{w}_{\mathrm{m}}$.

Proof:

1. In this case $\overline{T}_{\mathrm{m}} \subseteq T_{\mathrm{m}}$.
   - If $\boldsymbol{p} \in \overline{T}_{\mathrm{m}}$ then $\boldsymbol{p} \in T_{\mathrm{m}}$ and $x = w_{\mathrm{m}}$; therefore we have
   $$\overline{w}_{\mathrm{m}} = \frac{c(\overline{T}_{\mathrm{m}}) - x}{g(\overline{T}_{\mathrm{m}}) - 1} = \frac{c(\overline{T}_{\mathrm{m}}) - w_{\mathrm{m}}}{g(\overline{T}_{\mathrm{m}}) - 1} \geq \frac{g(\overline{T}_{\mathrm{m}})w_{\mathrm{m}} - w_{\mathrm{m}}}{g(\overline{T}_{\mathrm{m}}) - 1} = w_{\mathrm{m}}.$$
   - If $\boldsymbol{p} \notin \overline{T}_{\mathrm{m}}$ then let $\overline{P}$ denote the path from $\boldsymbol{p}$ to $\overline{T}_{\mathrm{m}}$. In this case $x < c(\overline{P})$ (because if $x \geq c(\overline{P})$ then $\overline{w}(\overline{P}) = 0$ and $\overline{T}_{\mathrm{m}} \cup \overline{P}$ is a better candidate for $\overline{T}_{\mathrm{m}}$ by Lemma 4.1). So $\overline{w}_{\mathrm{m}} = \overline{w}(\overline{T}_{\mathrm{m}}) = w(\overline{T}_{\mathrm{m}}) \geq w(T_{\mathrm{m}}) = w_{\mathrm{m}}$.
2. Note that $g(\overline{Q}) < 1 \iff d(\overline{Q}) = 0, \alpha > 0$ and $n(\overline{Q}) = 1$. So if $g(\overline{Q}) < 1$ then $\overline{Q} = \{\boldsymbol{p}\}$ and $\boldsymbol{p}$ is a lonely leaf. Then $x = c_{\boldsymbol{p}} + \alpha w_{\mathrm{m}}$ and
   $$\overline{w}_{\mathrm{m}} = \frac{c(S) + c(\overline{Q}) - x}{g(\overline{T}_{\mathrm{m}}) - 1} = \frac{c(S) - \alpha w_{\mathrm{m}}}{g(S) - \alpha} \geq \frac{g(S)w_{\mathrm{m}} - \alpha w_{\mathrm{m}}}{g(S) - \alpha} = w_{\mathrm{m}}.$$
3. In this case $\overline{Q}$ is a path. By *Reas* in the tree obtained by contraction of T_{m} we have $x \leq c(\overline{Q})$. Hence $w_{\mathrm{m}} \leq w(S) = \frac{c(S)}{n(S)} \leq \frac{c(S)+c(\overline{Q})-x}{n(S)} = \overline{w}_{\mathrm{m}}$.

□

Lemma 5.14: *The remaining cases all give a contradiction:*

1. $n(\overline{Q}) = 2, d(\overline{Q}) = 0$ *and* $\alpha = 1$, *(i.e.* $g(\overline{Q}) = 1$*)*
2. $n(\overline{Q}) = 1, d(\overline{Q}) = 1$ *and* $\alpha > 0$, *(i.e.* $g(\overline{Q}) = 1$*)*
3. $g(\overline{Q}) > 1$.

Proof: *1.* First note that if $n(\overline{Q}) = 2, d(\overline{Q}) = 0$ and $\alpha = 1$ then $\overline{Q} = \{\boldsymbol{p}, \pi(\boldsymbol{p}) =: \boldsymbol{q}\}$ because the other three possibilities for $\overline{Q}$ give a contradiction immediately:

- $\overline{Q} = \{\boldsymbol{p}\}$. Then $x \geq c(\overline{Q})$ otherwise S is a better candidate for $\overline{T}_{\mathrm{m}}$. Further $\frac{c(\overline{Q})+w_{\mathrm{m}}}{2} = x \geq c(\overline{Q})$ which implies $w(\overline{Q}) = c(\overline{Q}) \leq w_{\mathrm{m}}$ and $T_{\mathrm{m}} \cup \overline{Q}$ is a better candidate for T_{m}.

- $\overline{Q} = \{\boldsymbol{p}, \boldsymbol{q}\}$ with $\pi(\boldsymbol{q}) = \boldsymbol{p}$. Now $S \cup \{\boldsymbol{p}\}$ is a better candidate for $\overline{T}_{\mathrm{m}}$ because $c_{\boldsymbol{q}} > 0$.
- $\overline{Q} = \{\boldsymbol{p}, \boldsymbol{q}, \tilde{\boldsymbol{q}}\}(\pi(\boldsymbol{q}) = \pi(\boldsymbol{p}) = \tilde{\boldsymbol{q}})$. As $c_{\boldsymbol{q}} > 0$ we find that $S \cup \{\boldsymbol{p}, \tilde{\boldsymbol{q}}\}$ is a better candidate for $\overline{T}_{\mathrm{m}}$.

Contracting T_{m} in Γ and computing σ^{α} in the component which contains $\boldsymbol{p}$ gives $x = c_{\boldsymbol{p}} + \frac{1}{2}(c_{\boldsymbol{q}} + w_{\mathrm{m}})$. As $x \geq c(\overline{Q})$ (because otherwise S is a better candidate for $\overline{T}_{\mathrm{m}}$) we have $c_{\boldsymbol{q}} \leq w_{\mathrm{m}}$, which implies $w(\{\boldsymbol{q}\}) \leq w_{\mathrm{m}}$ and $T_{\mathrm{m}} \cup \{\boldsymbol{q}\}$ is a better candidate for T_{m}. Contradiction.

2/3. We prove 2 and 3 partly simultaneously. We first show, by induction to the number of nodes, that $w_{\mathrm{m}} = \overline{w}_{\mathrm{m}}$. The case $|V| = 2$ follows immediately. Let $\Gamma \in \mathcal{T}_0$, $i \in N$, be a counterexample with $|V|$ as small as possible. Let $\boldsymbol{p}$ be such that $i \in N_{\boldsymbol{p}}$. Let $\tilde{\Gamma}$ be the tree obtained by performing one iteration in the computation of $\sigma^{\alpha}(\Gamma)$ such that T_{m} is contracted. Let $\tilde{\Gamma}^1$ be the component of $\tilde{\Gamma}$ containing $\boldsymbol{p}$. Then $\tilde{\Gamma}$ has less nodes than Γ and $\sigma_i^{\alpha}(\tilde{\Gamma}) = x$. Let $\tilde{T}_{\mathrm{m}}^1$ be the largest trunk of $\tilde{\Gamma}^1$ with minimal weight $\tilde{w}_{\mathrm{m}}^1$. We have $w_{\mathrm{m}} < \tilde{w}_{\mathrm{m}}^1$ (Lemma 5.6). We shall show that $w := w_{\mathrm{m}}((\tilde{\Gamma}^1)_{-i}^{x}) < \tilde{w}_{\mathrm{m}}^1$ which gives a tree with less nodes than Γ for which the minimal weight strictly decreases when i leaves.

2. Suppose that $n(\overline{Q}) = 1, d(\overline{Q}) = 1$ and $\alpha > 0$. In this case there are two possibilities: $\overline{Q} = \{\boldsymbol{p}\}$ or $\overline{Q} = \{\boldsymbol{p}, \pi(\boldsymbol{p})\}$. In both cases we have $x \geq c(\overline{Q})$, because otherwise S is a better candidate for $\overline{T}_{\mathrm{m}}$. Now we get a contradiction:

$$w \leq \frac{c(\overline{Q}) + \alpha w_{\mathrm{m}} - x}{\alpha} \leq w_{\mathrm{m}} < \tilde{w}_{\mathrm{m}}^1.$$

3. Now suppose that $g(\overline{Q}) > 1$. Using $c(S) \geq g(S)\overline{w}_{\mathrm{m}}$ and $\overline{w}_{\mathrm{m}} = \frac{c(S)+c(\overline{Q})-x}{g(S)+g(\overline{Q})-1}$ we find $c(\overline{Q}) - x \leq \overline{w}_{\mathrm{m}}(g(\overline{Q}) - 1) < w_{\mathrm{m}}(g(\overline{Q}) - 1)$. So

$$w \leq \frac{c(\overline{Q}) + \alpha w_{\mathrm{m}} - x}{g(\overline{Q}) - 1 + \alpha} < w_{\mathrm{m}} < \tilde{w}_{\mathrm{m}}^1.$$

Conclusion: in all cases $\overline{w}_{\mathrm{m}} = w_{\mathrm{m}}$. We shall now show that $T_{\mathrm{m}} \cup \tilde{T}_{\mathrm{m}}^1 \backslash \{\tilde{\boldsymbol{r}}\}$ is a better candidate for T_{m}, from which it follows that the last two cases of this lemma can not occur.

2. If $n(\overline{Q}) = 1, d(\overline{Q}) = 1$ and $\alpha > 0$ then we have

$$\tilde{w}(\tilde{T}_{\mathrm{m}}^1) = \tilde{w}_{\mathrm{m}}(\tilde{\Gamma}^1) = w \leq \frac{c(\overline{Q}) + \alpha w_{\mathrm{m}} - x}{\alpha} \leq w_{\mathrm{m}} \leq \tilde{w}(\tilde{T}_{\mathrm{m}}^1) = \frac{c(\tilde{T}_{\mathrm{m}}^1) + \alpha w_{\mathrm{m}}}{g(\tilde{T}_{\mathrm{m}}^1)},$$

where the last inequality follows from *FR*. Now we find $w(\tilde{T}_{\mathrm{m}}^1) = \frac{c(\tilde{T}_{\mathrm{m}}^1)}{g(\tilde{T}_{\mathrm{m}}^1)-\alpha} = w_{\mathrm{m}}$ and by Lemma 4.1: $w(T_{\mathrm{m}} \cup \tilde{T}_{\mathrm{m}}^1 \backslash \{\tilde{\boldsymbol{r}}\}) \leq w(T_{\mathrm{m}})$.

3. Finally if $g(\overline{Q}) > 1$ then we find:

$$\tilde{w}(\tilde{T}_{\mathrm{m}}^1) = w_{\mathrm{m}}(\tilde{\Gamma}^1) = w \leq \frac{c(\overline{Q}) + \alpha w_{\mathrm{m}} - x}{g(\overline{Q}) - 1 + \alpha} \leq w_{\mathrm{m}} \leq \tilde{w}(\tilde{T}_{\mathrm{m}}^1)$$

and again $T_{\mathrm{m}} \cup \tilde{T}^1_{\mathrm{m}} \backslash \{\tilde{\boldsymbol{r}}\}$ is a better candidate for T_{m}. □

Now we can finish the proof that σ^α satisfies *ν-Cons* by induction to the number of nodes. If $|V| = 2$ then *ν-Cons* follows immediately. Now suppose that $|V| > 2$. From Lemma 5.13 (and 5.14) we learn that $\overline{w}_{\mathrm{m}} = w_{\mathrm{m}}$. If we go through the proof of Lemma 5.10 we see that $\overline{w}(T_{\mathrm{m}}) = \overline{w}(\overline{T}_{\mathrm{m}})$, i.e. T_{m} has minimal weight with respect to $\overline{w}$ and therefore $T_{\mathrm{m}} \subseteq \overline{T}_{\mathrm{m}}$. We distinguish between the same cases as in Lemma 5.13:

- If $\ell = 0$ then $T_{\mathrm{m}} = \overline{T}_{\mathrm{m}}$ and $\overline{w}_{\mathrm{m}} = w_{\mathrm{m}}$ so $\sigma^\alpha_j(\Gamma) = \sigma^\alpha_j(\Gamma^x_{-i})$ for all $j \in N(T_{\mathrm{m}})$.

  If $\boldsymbol{p} \in T_{\mathrm{m}}$ then we find after one iteration in the algorithm the same tree for Γ and Γ^x_{-i}. So we also have $\sigma^\alpha_j(\Gamma) = \sigma^\alpha_j(\Gamma^x_{-i})$ if $j \notin N(T_{\mathrm{m}})$. If $\boldsymbol{p} \notin T_{\mathrm{m}}$ then $x \leq c(\overline{P})$ and one iteration in the algorithm applied on Γ and Γ^x_{-i} respectively gives again a tree $\tilde{\Gamma}^1$ and its reduced tree $(\tilde{\Gamma}^1)^x_{-i}$. We can proceed by induction.

- If $\ell = 1$ and $g(\overline{Q}) < 1$ then $\overline{T}_{\mathrm{m}} = T_{\mathrm{m}} \cup \{\boldsymbol{p}\}$ and $N_{\boldsymbol{p}} = \{i\}, d(\overline{Q}) = 0, \alpha > 0$. Contraction of T_{m} gives a tree $\tilde{\Gamma}$ for which $\{\boldsymbol{p}, \tilde{\boldsymbol{r}}\}$ is a trunk. We first contract $\{\boldsymbol{p}, \tilde{\boldsymbol{r}}\}$. Then we get the same tree as the one obtained by applying one iteration of the algorithm on Γ^x_{-i}, such that $\overline{T}_{\mathrm{m}}$ is contracted.

- If $\ell = 1, \alpha = 0$ and $n(\overline{Q}) = 1$ then $\overline{T}_{\mathrm{m}} = T_{\mathrm{m}} \cup \overline{Q}$ and $\overline{Q}$ is a path. One iteration in the algorithm applied on Γ and Γ^x_{-i}, such that in both cases T_{m} is contracted, gives a tree $\tilde{\Gamma}^1$ and its reduced tree $(\tilde{\Gamma}^1)^x_{-i}$ and induction completes the proof.

5.2 Game-theoretical properties

The first question we ask is whether there is an $\alpha \in [0, 1]$ such that σ^α is equal to well-known solution concepts for standard tree games. The answer is affirmative in case of the nucleolus and the constrained egalitarian solution as Theorem 5.15 shows:

Theorem 5.15:

- *If $\alpha = 1$ then σ^α coincides with the nucleolus (Meggido (1978)).*

- *If $\alpha = 0$ then σ^α coincides with the constrained egalitarian solution (Dutta and Ray (1989)).*

For the Shapley value (Shapley (1953)) and the τ-value (Tijs (1981)) we get negative answers, because both solution concepts do not satisfy *ν-Cons* as the following example shows:

Example 5.1: Consider the standard tree enterprise Γ as indicated in Figure 5.1.

The Shapley value of the standard tree game (N, C) related to Γ is given by $\Phi_i(C) = \frac{10}{4} + \frac{20}{3} = 9\frac{1}{6}$ if $i \in N_{\boldsymbol{q}}$, $\Phi_i(C) = \frac{10}{4} = 2\frac{1}{2}$ if $i \in N_{\boldsymbol{p}}$. Reduction with respect to a player in node $\boldsymbol{q}$ gives the tree $\overline{\Gamma}$. Now we have $\Phi_i(\overline{C}) = \frac{10}{3} \neq \Phi_i(C)$ if $i \in N_{\boldsymbol{p}}$.

The τ-value of (N, C) becomes $\tau_i(C) = 9$ if $i \in N_{\boldsymbol{q}}$ and $\tau_i(C) = 3$ if $i \in N_{\boldsymbol{p}}$. Reduction with respect to a player in node $\boldsymbol{q}$ yields the tree Γ'. Then $\tau_i(C') = 4\frac{1}{26} \neq \tau_i(C)$ if $i \in N_{\boldsymbol{p}}$. ○

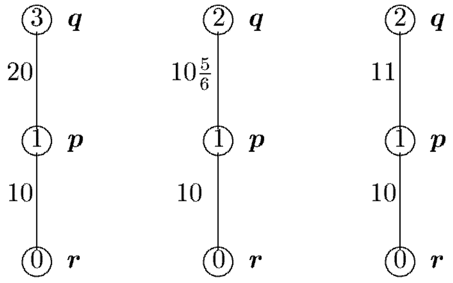


Figure 5.1: *Standard tree enterprises* Γ, $\overline{\Gamma}$ *and* Γ' *respectively.*

The next proposition shows that σ^α is a core element of the related standard tree game for all $\alpha \in [0,1]$.

Proposition 5.16: *Let σ be a solution rule on $\mathcal{T}$ satisfying Reas and ν-Cons. Then* $\sigma(\Gamma) \in \text{Core}(N,C) := \{y \in \mathbb{R}^N \mid y(N) = C(N),\ y(S) \leq C(S)\ \forall S \subset N\}$ *for all* $\Gamma \in \mathcal{T}$.

Proof: The proof is by induction to the number of players. If $n = 1$ then $\sigma(\Gamma) = c(P) = C(N)$, where P is the path from the root $\boldsymbol{r}$ to the node the player lives in.

Let $\Gamma \in \mathcal{T}$ with $n > 1$, $x := \sigma(\Gamma)$, $i \in N$. Let $(\overline{N}, \overline{C})$ be the game related to $\Gamma^{x_i}_{-i}$ i.e. $\overline{N} = N\backslash\{i\}$, $\overline{C}(S) = \min\{C(S \cup \{i\}) - x_i, C(S)\} \quad \forall S \subset N$. By *$\nu$-Cons* with respect to player i and the induction hypothesis we get

$$\sigma_{-i}(\Gamma) = \sigma(\Gamma^{x_i}_{-i}) \in \text{Core}(\overline{N}, \overline{C}).$$

So $x(\overline{N}) = \overline{C}(\overline{N})$ and $x(S) \leq \overline{C}(S)$ for all $S \subset N$. From $x_i \geq \text{mc}_i = C(N) - C(\overline{N})$ and $x(\overline{N}) = \overline{C}(\overline{N})$ it follows that $x(N) = \overline{C}(\overline{N}) + x_i = \min\{C(N) - x_i, C(\overline{N})\} + x_i = C(N)$.

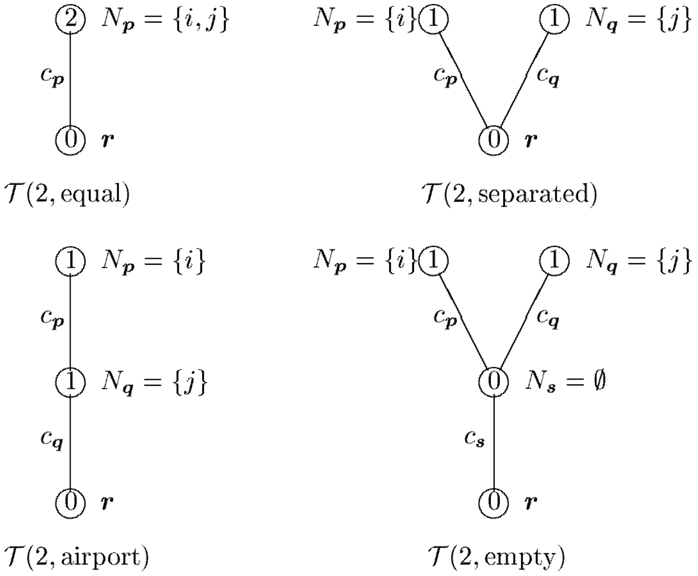
Now let $S \subset N$. If $i \in S$ then $x(S\backslash i) \leq \overline{C}(S\backslash i) \leq C(S) - x_i$ and therefore $x(S) \leq C(S)$. If $i \notin S$ then $x(S) \leq \overline{C}(S) \leq C(S)$. So $x \in \text{Core}(N, C)$. □

6 Characterization of the solution rules σ^α

We shall now prove the main theorem of this paper.

Theorem 6.1: *Let σ be a solution rule on $\mathcal{T}$. Then σ satisfies Hom, Reas, Del, Contr, FR, CostMon and ν-Cons if and only if there exists a parameter $\alpha \in [0,1]$ such that $\sigma = \sigma^\alpha$.*

We have already shown in the previous section that σ^α satisfies the seven properties. We shall now prove the 'only if' part. Suppose that σ satisfies the seven properties. Note that σ also satisfies *Eff* by Proposition 3.1. We have, according to Proposition 3.2, only to consider the cases $n = 1$ and $n = 2$. The case $n = 1$ follows from *Reas*. Now let $\Gamma \in \mathcal{T}$, $n = 2$, say $N = \{i, j\}$. It is sufficient to consider the four classes of trees as indicated in Figure 6.1: $\mathcal{T}(2, \text{equal})$, $\mathcal{T}(2, \text{separated})$, $\mathcal{T}(2, \text{airport})$ with $c_p > 0$ and $\mathcal{T}(2, \text{empty})$ with $c_p, c_q, c_s > 0$.



$\mathcal{T}(2, \text{equal})$ $\mathcal{T}(2, \text{separated})$

$\mathcal{T}(2, \text{airport})$ $\mathcal{T}(2, \text{empty})$

Figure 6.1: *Trees with two players.*

On $\mathcal{T}(2, \text{equal})$ we have $\sigma_i(\Gamma) = \sigma_j(\Gamma) = c_p/2 = \sigma_i^\alpha(\Gamma) = \sigma_j^\alpha(\Gamma)$ for all $\alpha \in [0, 1]$ by *Eff* and *FR*. On $\mathcal{T}(2, \text{separated})$ we have, by *Reas*, $\sigma_i(\Gamma) = c_p = \sigma_i^\alpha(\Gamma)$ and $\sigma_j(\Gamma) = c_q = \sigma_j^\alpha(\Gamma)$ for all $\alpha \in [0, 1]$. The next two subsections consider the other two cases.

6.1 $\mathcal{T}(2, \text{airport})$

This case is much more complicated. By *Hom* we may assume that $c_p = 1$. Now σ_i and σ_j are functions of $x := c_q$. Consider the following function $f : \mathbb{R}_+ \to \mathbb{R}, f(x) := \sigma_i(\Gamma(x))$, where $\Gamma(x)$ as in Figure 6.2. So $f(x)$ is the amount that player i pays if $c_p = 1$ and $c_q = x$. Then player i pays $x + 1 - f(x)$ by *Eff*. For the σ^α-rules, f^α becomes: $f^\alpha(x) = x + 1 - \min\{\frac{x}{1+\alpha}, \frac{x+1}{2}\} = \max\{\frac{\alpha}{\alpha+1}x + 1, \frac{x+1}{2}\}$. So we want to show that $f(x) = \max\{1 + \beta x, \frac{x+1}{2}\}$ for some $\beta \in [0, \frac{1}{2}]$.

The seven properties of σ put some restrictions on f. We mention

- $\frac{1}{2}(x + 1) \leq f(x)\ \forall x \in \mathbb{R}_+$ (*FR*).
  We write $U := \{x \in \mathbb{R}_+ \mid f(x) = \frac{1}{2}(x + 1)\}$,
- $1 \leq f(x) \leq 1 + \frac{1}{2}x\ \forall x \in \mathbb{R}_+$, in particular $f(0) = 1$.
  The first inequality follows from *Reas*. For the second one consider the tree Γ of $\mathcal{T}(2, \text{airport})$ in Figure 6.1 where $c_p := 0$ and $c_q := x$. In this tree both players

pay $\frac{1}{2}x$ by *Contr*, *Eff* and *FR*. By *CostMon* we find $x+1-f(x) \geq \frac{1}{2}x$, i.e. $f(x) \leq 1+\frac{1}{2}x$. If $f(x) = 1+\frac{1}{2}x$ on $\mathbb{R}_+$ then $f = f^\alpha$ for $\alpha = 1$ and we are done. So we assume from now on that $f(x) < 1+\frac{1}{2}x$ for some $x \in \mathbb{R}_+$.

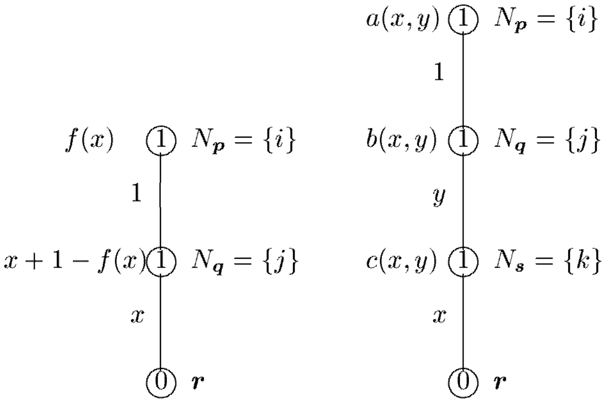
Let $V := \{x \in \mathbb{R}_+ \mid f(x) = 1+\frac{1}{2}x\}$. Note that $0 \in V$ and $U \cap V = \emptyset$.

- The function $x \mapsto \frac{f(x)-1}{x}$ $(x > 0)$ is (weakly) increasing.

  Let $x > 0,\ y > 1$ and consider the tree Γ of $\mathcal{T}(2, \text{airport})$ of Figure 6.1 where $c_{\boldsymbol{p}} := y,\ c_{\boldsymbol{q}} := x$. By *CostMon* and *Hom* we find $y(\frac{x}{y}+1-f(\frac{x}{y})) \geq x+1-f(x)$, i.e. $\frac{f(\frac{x}{y})-1}{\frac{x}{y}} \leq \frac{f(x)-1}{x}$ as was to be shown.

  Suppose that $f(x_1) = f(x_2)$, where $0 < x_1 < x_2$. Then $\frac{f(x_1)-1}{x_1} \leq \frac{f(x_2)-1}{x_2}$, which implies $f(x_1) = f(x_2) = 1$. As $f(x_2) \geq \frac{1}{2}(x_2+1)$ we have $x_2 \leq 1$. In particular f is strictly monotonic on $[1, \infty)$.

- For every $\delta > 0 : f(x+\delta) - f(x) \leq \delta\ \forall x \in \mathbb{R}_+$ (*CostMon*) and continuity of f follows. Note that $V \neq \mathbb{R}_+$, $f(x) \geq \frac{1}{2}(x+1)$ and continuity of f imply that there exists a number $x \in \mathbb{R}_+ \backslash V$ such that $f(x) > 1$. We use this later on.



$f(x)$
1
$N_{\boldsymbol{p}} = \{i\}$
1
$x+1-f(x)$
1
$N_{\boldsymbol{q}} = \{j\}$
x
0
$\boldsymbol{r}$
$a(x,y)$
1
$N_{\boldsymbol{p}} = \{i\}$
1
$b(x,y)$
1
$N_{\boldsymbol{q}} = \{j\}$
y
$c(x,y)$
1
$N_{\boldsymbol{s}} = \{k\}$
x
0
$\boldsymbol{r}$

Figure 6.2: *The trees* $\Gamma(x)$ *and* $\Gamma(x,y)$.

The property *ν-Cons* with respect to i and j does not give extra conditions on f. We get new conditions if we consider the airport problem $\Gamma(x,y)$ with 3 players (Figure 6.2). Define the functions $a, b, c : \mathbb{R}_+^2 \to \mathbb{R}$ by: $a(x,y) := \sigma_i(\Gamma(x,y)), b(x,y) := \sigma_j(\Gamma(x,y)), c(x,y) := \sigma_k(\Gamma(x,y))$. The conditions on σ put conditions on a, b, c and f. For all $(x,y) \in \mathbb{R}_+^2$ we have

- $a(x,y) + b(x,y) + c(x,y) = x+y+1$ (*Eff*).
- $1 \leq a(x,y) \leq x+y+1$ (*Reas*)
  $0 \leq b(x,y) \leq x+y$
  $0 \leq c(x,y) \leq x$.

- $0 \le c(x,y) \le b(x,y) \le a(x,y) \quad (FR)$.
- Let $\delta > 0$. By *CostMon*: $a(x,y) \le a(x+\delta, y+\delta)$ and the same inequality holds for b and c. By *Eff*: $a(x+\delta, y+\delta) - a(x,y) + b(x+\delta, y+\delta) - b(x,y) + c(x+\delta, y+\delta) - c(x,y) = 2\delta$. So $a(x+\delta, y+\delta) - a(x,y) \le 2\delta$ and a is continuous. Similarly b and c are continuous.

The conditions concerning ν-*Cons* are more complicated:

- ν-*Cons*(c) (i.e. ν-*Cons* w.r.t. player k) (*Del*):
$$a(x,y) = f(x + y - c(x,y)). \tag{6.1}$$
- ν-*Cons*(b):
  - if $b(x,y) < y$ then (*Hom*, *Del*)
$$(1 + y - b(x,y)) f(\frac{x}{1 + y - b(x,y)}) = a(x,y), \tag{6.2}$$
  - if $b(x,y) \ge y$ then (*Contr*)
$$a(x,y) = f(x + y - b(x,y)). \tag{6.3}$$
- ν-*Cons*(a):
  - if $a(x,y) < 1 + y$ then (*Hom*, *Contr*)
$$(1 + y - a(x,y)) f(\frac{x}{1 + y - a(x,y)}) = b(x,y). \tag{6.4}$$
  - if $a(x,y) \ge 1 + y$ then (*Contr*)
$$b(x,y) = c(x,y). \tag{6.5}$$

From now on, we only consider points $(x,y) \in \mathbb{R}^2_+$ with $y = x + 1 - f(x)$. Therefore we write $a(x)$ instead of $a(x,y)$ etc.

As $f(x) \ge f(0) = 1$ for all $x \in \mathbb{R}_+$, we have $U \subseteq [1, \infty)$. The following lemma shows that U is an unbounded interval.

Lemma 6.2: *There exists a number $u_0 \ge 1$ such that $U = [u_0, \infty)$.*

Proof: Take $x \in \mathbb{R}_+ \backslash V$ such that $f(x) > 1$. Let $y := x + 1 - f(x)$ and $s := x + y - c(x)$. From ν-*Cons*(c) and *Eff* we get the following equalities:

$$\begin{array}{rcl} a(x) & = & f(s), \\ b(x) & = & s + 1 - f(s), \\ c(x) & = & x + y - s. \end{array}$$

We shall first show that $s = x$. From

$$s + 1 - f(s) = b(x) \ge c(x) = 2x + 1 - f(x) - s$$

we get $s \geq x$ (*FR* and monotonicity of the function $t \mapsto 2t + 1 - f(t)$). Then $b(x) = s + 1 - f(s) \geq x + 1 - f(x) = y$. So (by ν-*Cons*(b) and ν-*Cons*(c)) $f(x + y - b(x)) = a(x) = f(x+y-c(x))$. If $s > x$ then $b(x) > c(x)$ which implies that $a(x) = 1$, because f is strictly monotonic on $[1, \infty)$. As a consequence $f(x) \leq f(s) = a(x) = 1$, contradicting $f(x) > 1$. Conclusion: $s = x$.

We further have $a(x) < y + 1$ (because if $a(x) \geq y + 1$, then $f(x) = f(s) = a(x) \geq y + 1 = x + 2 - f(x)$ gives $2f(x) \geq x + 2$, i.e. $x \in V$). Hence ν-*Cons*(a) gives $(1 + y - a(x))f(t(x)) = b(x)$, where $t(x) := \frac{x}{1+y-a(x)} = \frac{x}{2+x-2f(x)}$. So

$$f(t(x)) = \frac{x + 1 - f(x)}{x + 2 - 2f(x)} = \frac{1}{2}(t(x) + 1), \text{ i.e. } t(x) \in U.$$

So for $x \notin V$, $f(x) > 1$, we have $t(x) \in U$. If $x \in U$ then $t(x) = x$ and if $x \notin U$ then $t(x) > x$. Let u_0 be the smallest number in U (exists because U is nonempty, closed and $U \subseteq [1, \infty)$). Suppose that there exists a number x_1 such that $x_1 > u_0$ and $x_1 \notin U$. We may assume that $V \cap (u_0, x_1) = \emptyset$, because one can always choose $x_1 \in (u_0, v_0)$, where $v_0 := \inf(V \cap [u_0, \infty))$. Let x_2 be the largest number in U smaller than x_1. As $x_2 \in U$ we have $t(x_2) = x_2$. If $x_2 < z < x_1$ then $z \notin V$, $f(z) \geq \frac{1}{2}(z+1) > \frac{1}{2}(u_0 + 1) \geq 1$, so $t(z)$ is well defined, $t(z) \in U$ and $t(z) > z$. But $(x_2, x_1) \cap U = \emptyset$ from which we get $t(z) > x_1$ for all $z \in (x_2, x_1)$. So $t(x_2) \geq x_1$ (by continuity of t) contradicting $x_2 < x_1$. □

Corollary 6.3:

a) $V = \{0\}$.

b) If f is not strictly monotonic, then $f(x) = \max\{1, \frac{1}{2}(x+1)\} = f^{\alpha}(x)$ for $\alpha = 0$.

Proof: a) Suppose $v \in V \backslash \{0\}$. Take $x > v$, $x > u_0$. Then $\frac{1}{2} = \frac{f(v)-1}{v} \leq \frac{f(x)-1}{x}$, i.e. $x \in V \cap U = \emptyset$.

b) Suppose that $0 < x_1 < x_2$ and $f(x_1) = f(x_2)$. We have already seen that this implies $f(x_1) = f(x_2) = 1$ and $x_2 \leq 1$. Let $x := \max\{z \in \mathbb{R}_+ \mid f(z) = 1\}$. We follow more or less the proof of Lemma 6.2. Consider the point (x, y) where $y := x + 1 - f(x) = x$ and let $s := x + y - c(x)$. Again, we have $s \geq x$ (*FR*) and $b(x, y) \geq y$ (*CostMon*). Now ν-*Cons*(b) gives $a(x, y) = f(x+y-b(x)) = f(2x - b(x))$. From $b(x) \geq y = x$ it follows that $2x - b(x) \leq x$, so $a(x) = f(2x - b(x)) = 1$. Applying ν-*Cons*(c) gives $1 = a(x) = f(2x - c(x))$, from which we get $2x - c(x) \leq x$, i.e. $c(x) \geq x$. This together with *Reas* gives $c(x) = x$ and $b(x) = x$ (*Eff*). Finally, $a(xs) = 1 < 1 + y$, so ν-*Cons*(a) gives $xf(\frac{x}{x}) = b(x) = x$, i.e. $f(1) = 1$. Hence $x = 1$ and also $u_0 = 1$. □

Lemma 6.4: *If f is strictly monotonic then for all $x \in (0, u_0)$*

$$\frac{f(x) - 1}{x} = \frac{f(u_0) - 1}{u_0} =: \beta.$$

Note that f is completely determined and $0 < \beta < \frac{1}{2}$.

Proof: Let $x \in (0, u_0)$. Then $\frac{f(x)-1}{x} \leq \frac{f(u_0)-1}{u_0}$. Consider the point $(x, y) := (x, x + 1 - f(x))$. As $f(x) > f(0) = 1$ and $x \notin V$, we get from the proof of Lemma

6.2 that $a(x) < 1 + y$ and $b(x) = y$. Applying ν-*Cons*(a) and ν-*Cons*(b) gives

$$(1 + y - a(x))f(t(x)) = b(x) \quad \text{and} \quad a(x) = f(x), \text{ where } t(x) := \frac{x}{1 + y - a(x)}.$$

As $t(x) \in U$ (see again the proof of Lemma 6.2) we have

$$\frac{f(x) - 1}{x} = \frac{f(t(x)) - 1}{t(x)} \geq \frac{f(u_0) - 1}{u_0}.$$

So

$$\frac{f(x) - 1}{x} = \frac{f(u_0) - 1}{u_0}.$$

□

6.2 $\mathcal{T}$(2, empty)

We first consider the class $\mathcal{T}(3, \text{nonempty})$, consisting of trees like in $\mathcal{T}(2, \text{empty})$ (see Figure 6.1), but now there is one player in node $\boldsymbol{s}$, say $N_{\boldsymbol{s}} = \{k\}$ and $c_{\boldsymbol{s}} = 1$, $c_{\boldsymbol{p}}, c_{\boldsymbol{q}} > 0$. According to the previous section, there exists a number $\alpha \in [0, 1]$ such that $\sigma = \sigma^\alpha$ on $\mathcal{T}(2, \text{airport})$.

Lemma 6.5: *On* $\mathcal{T}(3, \text{nonempty})$, $\sigma = \sigma^\alpha$.

Proof: Take $\Gamma \in \mathcal{T}(3, \text{nonempty})$ and define $x := c_{\boldsymbol{p}}$, $y := c_{\boldsymbol{q}}$, $a(x, y) := \sigma_i(\Gamma)$, $b(x, y) := \sigma_j(\Gamma)$ and $c(x, y) := \sigma_k(\Gamma)$.

From *Reas* we get $a(x, y) \geq x$ and therefore applying ν-*Cons*(a) and *Hom* gives $b(x, y) = y \cdot f^\alpha(\frac{x+1-a(x,y)}{y})$. Similarly ($\nu$-*Cons*$(b)$ and *Hom*) $a(x,y) = x \cdot f^\alpha(\frac{y+1-b(x,y)}{x})$. As $f^\alpha(z) = \max\{\frac{\alpha}{1+\alpha}z + 1, \frac{z+1}{2}\}$, the following equalities must hold:

$$\begin{aligned} a(x, y) &= \max\{\frac{\alpha}{1 + \alpha}(y + 1 - b(x, y)) + x, \frac{x + y + 1 - b(x, y)}{2}\} \\ b(x, y) &= \max\{\frac{\alpha}{1 + \alpha}(x + 1 - a(x, y)) + y, \frac{x + y + 1 - a(x, y)}{2}\}. \end{aligned}$$

By distinguishing between four cases, in fact corresponding with four trunks of Γ, it can be shown that $a(x, y) = \sigma_i^\alpha(\Gamma)$, $b(x, y) = \sigma_j^\alpha(\Gamma)$ and $c(x, y) = \sigma_k^\alpha(\Gamma)$.

□

Lemma 6.6: *On* $\mathcal{T}(2, \text{empty})$, $\sigma = \sigma^\alpha$.

Proof: Let $\Gamma \in \mathcal{T}$ (2, empty). Define $x := c_{\boldsymbol{p}}$, $y := c_{\boldsymbol{q}}$. We may assume that $c_{\boldsymbol{s}} = 1$ (*Hom*). We construct a tree $\Gamma' \in \mathcal{T}(3, \text{nonempty})$ with costs $c_{\boldsymbol{p}} = x'$, $c_{\boldsymbol{q}} = y'$ and $c_{\boldsymbol{s}} = 1$, where x' and y' are defined later. Then we show that $\sigma(\Gamma) = \sigma^\alpha(\Gamma)$ using ν-*Cons*(c).

We distinguish four cases, again corresponding with four trunks of Γ.

1. $x \geq \frac{1-\alpha}{2\alpha}$, $y \geq \frac{1-\alpha}{2\alpha}$. In this case $\sigma_i^\alpha(\Gamma) = \frac{1}{2} + x$, $\sigma_j^\alpha(\Gamma) = \frac{1}{2} + y$. Define $x' := \frac{2\alpha}{1+2\alpha}x$, $y' := \frac{2\alpha}{1+2\alpha}y$. Then we have

   $$\sigma_k(\Gamma') = \sigma_k^\alpha(\Gamma') = \min\{\frac{1}{1+2\alpha}, \frac{x'+1}{2+\alpha}, \frac{y'+1}{2+\alpha}, \frac{x'+y'+1}{3}\} = \frac{1}{1+2\alpha}.$$

   Then $\sigma_i(\Gamma') = \frac{\alpha}{1+2\alpha} + x', \sigma_j(\Gamma') = \frac{\alpha}{1+2\alpha} + y'$. Applying ν-*Cons*(c) gives a tree $\Gamma'' \in \mathcal{T}(2, \text{empty})$ and $\sigma_i(\Gamma'') = \sigma_i(\Gamma'), \sigma_j(\Gamma'') = \sigma_j(\Gamma')$. By *Hom* we have $\sigma_i(\Gamma) = \frac{1+2\alpha}{2\alpha}\sigma_i(\Gamma'') = \frac{1+2\alpha}{2\alpha}\sigma_i(\Gamma') = x + \frac{1}{2} = \sigma_i^\alpha(\Gamma)$. And similarly $\sigma_j(\Gamma) = \frac{1+2\alpha}{2\alpha}\sigma_j(\Gamma'') = \frac{1+2\alpha}{2\alpha}\sigma_j(\Gamma') = y + \frac{1}{2} = \sigma_j^\alpha(\Gamma)$.

   For the next three cases we only give x' and y'. The proofs are similar.

2. $x \leq \frac{1-\alpha}{2\alpha}$, $y \geq \frac{1-\alpha}{1+\alpha}(x+1)$; $x' := \frac{(1+\alpha)x}{2+\alpha+x}$, $y' := \frac{(1+\alpha)y}{2+\alpha+x}$.

3. $y \leq \frac{1-\alpha}{2\alpha}$, $x \geq \frac{1-\alpha}{1+\alpha}(y+1)$; $x' := \frac{(1+\alpha)x}{2+\alpha+y}$, $y' := \frac{(1+\alpha)y}{2+\alpha+y}$.

4. $y \leq \frac{1-\alpha}{1+\alpha}(x+1)$, $x \leq \frac{1-\alpha}{1+\alpha}(y+1)$; $x' := \frac{2x}{3+x+y}$, $y' := \frac{2y}{3+x+y}$.

□

References

[1] **Davis and M. Maschler (1965)** *The kernel of a cooperative game.* Naval Research Logistics Quarterly 12, 223-259.

[2] **Dubey, P. (1982)** *The Shapley value as aircraft landing fees - revisited.* Management Science 28, 869-874.

[3] **Dutta, B. and D. Ray (1989)** *A Concept of Egalitarianism under Participation Constraints.* Econometrica 3, 615-635.

[4] **Galil, Z. (1980)** *Applications of efficient mergeable heaps for optimization problems on trees.* Acta Informatica 13, 53-58.

[5] **Granot, D., M. Maschler, G. Owen and W. Zhu (1996)** *The Kernel/Nucleolus of a Standard Tree Game.* International journal of game theory 25, 219-244.

[6] **Littlechild, S.C. (1974)** *A Simple Expression for the Nucleolus in a Special Case.* International journal of game theory 3, 21-29.

[7] **Littlechild, S.C. and G.F. Thompson (1977)** *Aircraft landing fees: a game theory approach.* Bell Journal of Economics 8, 186-204.

[8] **Maschler M., J. A. M. Potters and J. H. Reijnierse (1995)** *Monotonicity properties of the nucleolus of standard tree games.* Report 9556, University of Nijmegen, The Netherlands.

[9] **Megiddo, N. (1978)** *Computational complexity of the game theory approach to cost allocation for a tree.* Mathematics of Operations Research 3, 189-196.

[10] **Milnor, J.W. (1952)** *Reasonable outcomes for n-person games.* The Rand Corporation 196.

[11] **Potters, J.A.M. and P. Sudhölter (1999).** *Airport Problems and Consistent Allocation Rules.* Forthcoming in Mathematical Social Sciences.

[12] **Shapley, L.S. (1953)** *A value for N-person games.* Contributions to the theory of games II, H.W. Kuhn and A.W. Tucker eds. Princeton: Princeton University Press, 307-317

[13] **Sönmez, T. (1994)** *Population monotonicity of the nucleolus on a class of public good problems.* Mimeo University of Rochester

[14] **Sudhölter P. (1997).** *The modified nucleolus: properties and axiomatizations.* International journal of game theory 26, 147-182.

[15] **Tijs, S.H. (1981)** *Bounds for the core of a game and the τ-value* Game Theory and Mathematical Economics, eds. O. Moeschlin and D. Pallaschke, Amsterdam, North Holland, 123-132