Average tree solutions for graph games

Richard Baron and Sylvain Béal and Eric Remila and Philippe Solal

31. July 2008

Average tree solutions for graph games

R. Baron [*] S. Béal [†] E. Rémila [‡] P. Solal [§¶]

July 31, 2008

Abstract

In this paper we consider cooperative graph games being TU-games in which players cooperate if they are connected in the communication graph. We focus our attention to the average tree solutions introduced by Herings, van der Laan and Talman [6] and Herings, van der Laan, Talman and Yang [7]. Each average tree solution is defined with respect to a set, say T, of admissible rooted spanning trees. Each average tree solution is characterized by efficiency, linearity and an axiom of T-hierarchy on the class of all graph games with a fixed communication graph. We also establish that the set of admissible rooted spanning trees introduced by Herings, van der Laan, Talman and Yang [7] is the largest set of rooted spanning trees such that the corresponding average tree solution is a Harsanyi solution. One the other hand, we show that this set of rooted spanning trees cannot be constructed by a distributed algorithm. Finally, we propose a larger set of spanning trees which coincides with the set of all rooted spanning trees in clique-free graphs and that can be computed by a distributed algorithm.

1 Introduction

A situation in which a finite set of agents can obtain certain payoffs by cooperation can be described by a cooperative game with transferable utility. In

[*]CREUSET, Université de Saint-Etienne, France `richard.baron@univ-st-etienne.fr`

[†]GREDEG, Université de Nice Sophia Antipolis, France `beals@gredeg.cnrs.fr`

[‡]LIP, École Normale Supérieure de Lyon France `eric.remila@ens-lyon.fr`

[§]CREUSET, Université de Saint-Etienne, France `philippe.solal@univ-st-etienne.fr`

[¶]The authors are grateful to Jean-Jacques Herings and Dolf Talman for stimulating discussion and encouragement.

standard cooperative game theory it is assumed that any coalition of agents may form. On the other hand, in many social situations the collection of possible coalitions is restricted by social, hierarchical or communicational structures. Examples are games with communication structures (see, e.g. Myerson [9]), games with permission structures (see, e.g. Gilles, Owen and van den Brink [5]), games with precedence constraints (see, e.g. Faigle and Kern [4]) and more general models of games restricted on regular systems (see, e.g. Lange and Grabisch [8]). In this paper we restrict ourselves to cooperative games with communication structures. A communication structure is represented by an undirected graph. The vertices in the graph represent the agents and the edges represent the communication links between agents. A coalition of agents can only cooperate if they are connected. This yields a so-called graph game. A (single-valued) solution is a mapping that assigns to each graph game a vector of payoffs. In Herings, van der Laan and Talman [6] and Herings, van der Laan, Talman and Yang [7] (henceforth abbreviated HLT and HLTY respectively) average tree solutions are introduced. From the undirected graph, a set of rooted spanning trees is defined. Each rooted spanning tree describes how information travels across the graph and induces a specific marginal contribution vector. A solution is called an average tree solution if it is the average of those marginal contribution vectors over a set of admissible rooted spanning trees. HLT [6] restrict the analysis to cycle-free graph games and consider the set of all rooted spanning trees. Using efficiency and component fairness, the authors characterize the corresponding average tree solution for cycle-free graph games. Component fairness means that deleting a link between two agents yields for both resulting components the same average change in payoff, where the average is taken over the agents in the component. In case the undirected graph is arbitrary, HLTY [7] construct a specific set of admissible rooted spanning trees. The induced average tree solution has several advantages. It coincides with the Shapley value when the underlying graph is complete and with the average tree solution as defined by HLT [6] when the underlying graph is cycle-free. In addition, a link-convexity condition for graph games to have their average tree solution in the core is given.

Here we present an axiomatic characterization of the average tree solutions. The domain on which we establish this characterization consists of all graph games with a fixed undirected graph. Besides linearity and efficiency, we use an axiom of hierarchy defined with respect to a set of admissible rooted spanning trees. This third axiom is stated for unanimity graph games and requires that the payoff of any agent is proportional to the number of times his position is decisive in a rooted spanning tree. These three logically independent axioms uniquely determine the corresponding average tree solution for graph games.

We also discuss the properties of the set of admissible rooted spanning trees studied in HLTY [7]. We provide a simple characterization of this set and show that it is the largest set of rooted spanning trees making the corresponding average tree solution a Harsanyi solution (see Vasil'ev [11]). A Harsanyi solution distributes the Harsanyi dividends over the agents in the corresponding coalitions according to a chosen sharing system. The latter assigns to every (connected) coalition a sharing vector which specifies for each of its members the share in the dividend associated with the coalition. The payoff to each agent is thus equal to the sum of its shares in the dividends of all coalitions she belongs to.

In order to compare different average tree solutions in terms of the distribution of the Harsanyi dividends, we introduce a new set of admissible rooted spanning trees, the set of triangle-free trees. This set is larger than the set of admissible trees introduced in HLTY [7], and so, from above, the corresponding average tree solution can not be a Harsanyi solution. A connected coalition may give up a share of its Harsanyi dividend in favor of non-members if they play a decisive role in some trees. On the other hand, this average tree solution yields the Shapley value on the class of complete graph games and coincides with the average tree solution for cycle-free graphs as defined in HLT [6]. The basic idea in HLT [6] is to consider all possible rooted spanning trees. The set of triangle-free trees is equal to the set of all rooted spanning trees in clique free-graphs. In this way, the average tree solution defined with respect to the set of triangle-free trees generalizes the average tree solution defined in HLT [6] from cycle-free graph games to clique-free graph games.

We elaborate further on the proposed set of triangle-free trees by providing additional computational properties it satisfies. Assume that the agents have the opportunity to orientate links in order to create a rooted spanning tree. They communicate with their neighbors on the graph by sending messages over communication links and have only the ability to perform local computation using information concerning their neighborhood. Computation and message transmission between agents are asynchronous. Such a message-passing computation corresponds to a distributed algorithm. We design a distributed algorithm which finds triangle-free trees such that once a link has been selected, its orientation will never be reconsidered. Finally, we prove that no such algorithm exists for finding the admissible rooted spanning trees introduced in HLTY [7].

This paper is organized as follows. Section 2 is a preliminary section containing concepts from cooperative graph games. In section 3 we prove that efficiency, linearity and hierarchy uniquely determine an average tree solution on the class of all graph games with a fixed undirected graph. In section 4, we characterize the set of admissible trees introduced in HLTY [7], and the corresponding average tree solution is discussed in terms of the distribution of the Harsanyi dividends. A comparison with the average tree solution constructed from the set of all triangle-free trees is also given and a distributed algorithm to build triangle-free trees is proposed. Section 5 concludes.

2 Preliminaries

Consider a finite set of agents $N = \{1, 2, \ldots, n\}$, $n \in \mathbb{N}$, who face restrictions on communication. Each subset S of N is called a *coalition*. The bilateral communication possibilities between the agents are represented by an *undirected graph* (N, L), where the set of nodes coincides with the set of agents N, and the set of links L is a subset of the set of unordered pairs of elements of N. For each agent $i \in N$, the set $L_i = \{j \in N \mid \{i, j\} \in L\}$ denotes the *neighborhood* of i in (N, L). For each non-empty coalition S of N, $L(S) = \{\{i, j\} \in L \mid i, j \in S\}$ is the set of links between agents in S. The graph $(S, L(S))$ is the *subgraph* of (N, L) induced by S. A sequence of distinct agents $(i_1, i_2, \ldots, i_p)$ is a *path* in (N, L) if $\{i_q, i_{q+1}\} \in L$ for $q = 1, \ldots, p-1$. Two agents i and j are *connected* in (N, L) if $i = j$ or there exists a path from i to j. A graph (N, L) is *connected* if any two agents i and j in N are connected. A coalition S is connected in (N, L) if $(S, L(S))$ is a connected graph. Denote by $C(L)$ the set of connected coalitions in (N, L). It is assumed that only connected coalitions are able to cooperate. If a coalition S belongs to $C(L)$, then its members can fully coordinate their actions. The worth obtained by cooperation between connected agents is described by the function $v : C(L) \longrightarrow \mathbb{R}$ such that $v(\emptyset) = 0$. In order to guarantee that the grand coalition N can form, we assume that (N, L) is connected. The function v defines a *graph game* with transferable utility on (N, L). We take the communication graph (N, L) to be fixed, and therefore consider the vector space $\mathcal{C}_{N,L}$ of all graph games v on (N, L). For each coalition $S \in C(L)\backslash\{\emptyset\}$, define the *unanimity graph game* u_S on $C(L)$ by $u_S(T) = 1$ if $S \subseteq T$, and $u_S(T) = 0$ otherwise. It is well known that the set of unanimity graph games forms a basis for $\mathcal{C}_{N,L}$. Therefore, for each $v \in \mathcal{C}_{N,L}$, there exist unique real numbers a_S, $S \in C(L)$, such that $v = \sum_{S\in C(L)\backslash\{\emptyset\}} a_S u_S$. These real numbers are called the *Harsanyi dividends*.

A *single-valued solution* on $\mathcal{C}_{N,L}$ is a function f that assigns to every $v \in \mathcal{C}_{N,L}$ a payoff vector $f(v) \in \mathbb{R}^n$. The Shapley value (Shapley [10]) is a solution on $\mathcal{C}_{K_N}$, where K_N denotes the complete graph on N. Let Σ_N be the set of all permutations σ on N. For a given $\sigma \in \Sigma_N$ and $i \in N$, we define $S_i^\sigma = \{\sigma(1), \sigma(2), \ldots, \sigma(i)\}$ and $S_0^\sigma = \emptyset$. Pick any $v \in \mathcal{C}_{K_N}$ and consider the marginal contribution vector $m^\sigma(v)$ on $\mathbb{R}^n$ defined by $m^\sigma_{\sigma(i)}(v) = v(S_i^\sigma) - v(S_{i-1}^\sigma)$ for every $i \in N$. The *Shapley value* is the solution $\mathrm{Sh}(v)$ that assigns to every game $v \in \mathcal{C}_{K_N}$ the average of all marginal contribution vectors $m^\sigma(v)$, i.e.

$$\mathrm{Sh}(v) = \frac{1}{n!} \sum_{\sigma\in\Sigma_N} m^\sigma(v) \tag{2.1}$$

In HLT [6] and HLTY [7] the so-called average tree solutions are proposed. A solution is an average tree solution if it is the average of specific marginal contribution vectors. Each of such vectors is determined by a rooted spanning tree. A *spanning tree* of a connected graph is a minimal set of links that connect all agents. A spanning tree is *rooted* if one agent has been designated the root, in which case the links have a natural orientation. Denote by t_i such a spanning tree rooted at agent i. For any rooted spanning tree t_i, any agent $j \in N$, $t_i(j) \in N$ denotes the unique *predecessor* of j in t_i, with the convention that $t_i(i) = i$, which amounts to say that the set of ordered pairs $\{(t_i(j), j) \mid j \in N \setminus \{i\}\}$ is the set of directed links of t_i. The inverse image of $j \in N$ under t_i, denoted by $t_i^{-1}(j)$, is the possibly empty set of *successors* of j in t_i. An agent k is a *subordinate* of j in t_i if there is a directed path from j to k, i.e. if there is a sequence of distinct agents $(i_1, i_2, \ldots, i_p)$ such that $i_1 = j$, $i_p = k$ and for each $q = 1, 2, \ldots, p-1$, $i_{q+1} \in t_i^{-1}(i_q)$. The set $S_j^{t_i}$ denotes the union of the set of all subordinates of j in t_i and $\{j\}$. So, we have $t_i^{-1}(j) \subseteq S_j^{t_i}\backslash\{j\}$. Pick any $v \in \mathcal{C}_{N,L}$ and consider the marginal contribution vector $m^{t_i}(v)$ on $\mathbb{R}^n$ defined by $m_j^{t_i}(v) = v(S_j^{t_i}) - \sum_{k\in t_i^{-1}(j)} v(S_k^{t_i})$ for every $j \in N$. The marginal contribution $m_j^{t_i}(v)$ of $j \in N$ in t_i is thus equal to the worth of the coalition consisting of agent j and all his subordinates in t_i minus the sum of the worths of the coalitions consisting of any successor of

j and all subordinates of this successor in t_i. Denote by $T_{N,L}$ a non-empty set of admissible rooted spanning trees on (N, L). The *average tree* solution $\mathrm{AT}(v)$ with respect to $T_{N,L}$ assigns to every game $v \in \mathcal{C}_{N,L}$ the average of all marginal contribution vectors $m^{t_i}(v)$, i.e.

$$\mathrm{AT}(v) = \frac{1}{|T_{N,L}|} \sum_{t_i \in T_{N,L}} m^{t_i}(v). \tag{2.2}$$

3 Axiomatic characterization

In this section we characterize the average tree solutions given by (2.2) as the unique solution on $\mathcal{C}_{N,L}$ satisfying a collection of three logically independent axioms. The first two axioms are standard axioms in cooperative game theory.

Efficiency A solution f on $\mathcal{C}_{N,L}$ is efficient if for any $v \in \mathcal{C}_{N,L}$, it holds that $\sum_{i\in N} f_i(v) = v(N)$.

Linearity A solution f on $\mathcal{C}_{N,L}$ is linear if for any $v, w \in \mathcal{C}_{N,L}$ and any $a \in \mathbb{R}$, it holds that $f(av) = af(v)$ and $f(v + w) = f(v) + f(w)$.

Before introducing the third axiom, we need a few more definitions and notations. Let $T_{N,L}$ be a non-empty set of rooted spanning trees. For each $S \in C(L)\backslash\{\emptyset\}$ and each $t_i \in T_{N,L}$ denote the smallest subtree of t_i that contains S by $\hat{t}_i(S)$ and denote its root by $r_{\hat{t}_i(S)}$. Note that if S remains a connected coalition in t_i, $\hat{t}_i(S)$ is the subtree $t_i(S)$ of t_i induced by S and so the root of $t_i(S)$ is a member of S. A *system* on $C(L)\backslash\{\emptyset\}$ is a collection of vectors $h = (h^S)_{S\in C(L)\backslash\{\emptyset\}}$ where $h^S \in \mathbb{R}^n_+$ for each $S \in C(L)$. Consider the following system on $C(L)\backslash\{\emptyset\}$: for each $S \in C(L)\backslash\{\emptyset\}$ and each $i \in N$, h^S_i is equal to the average number of times agent i is the root of such a subtree that contains S, i.e. $h^S_i = |\{t_k \in T_{N,L} \,|\, i = r_{\hat{t}_k(S)}\}|/|T_{N,L}|$. The support of h^S, consisting of all agents with a strictly positive weight in h^S, will be denoted by $B(h^S) = \{i \in N \,|\, h^S_i > 0\}$.

$T_{N,L}$**-hierarchy** A solution f on $\mathcal{C}_{N,L}$ satisfies $T_{N,L}$-hierarchy if for any unanimity graph game u_S, $S \in C(L)\backslash\{\emptyset\}$, and any pair of distinct agents $\{i, j\}$, it holds that $h^S_i f_j(u_S) = h^S_j f_i(u_S)$.

The $T_{N,L}$-hierarchy axiom is reminiscent of the hierarchical strength axiom introduced by Faigle and Kern [4] in order to characterize a Shapley value for cooperative games with precedence constraints.

Proposition 3.1 *For each non-empty set $T_{N,L}$, there is a unique solution f on $\mathcal{C}_{N,L}$ that satisfies efficiency, linearity and $T_{N,L}$-hierarchy.*

Proof. Fix a non-empty set $T_{N,L}$. Pick any $S \in C(L)\backslash\{\emptyset\}$. By definition, $u_S(N) = 1$. By $T_{N,L}$-hierarchy, $h_i^S f_j(u_S) = h_j^S f_i(u_S)$ for each distinct pair of agents $\{i, j\}$. Assume that $i \in B(h^S)$ and $j \notin B(h^S)$. Then, $0 = h_i^S f_j(u_S)$ and so $f_j(u_S) = 0$. Thus, efficiency becomes $\sum_{j \in B(h^S)} f_j(u_S) = 1$. Combining this equation with the $T_{N,L}$-hierarchy axiom, we first get for each $i \in B(h^S)$:

$$\sum_{j \in B(h^S)} f_j(u_S) = \sum_{j \in B(h^S)} f_i(u_S) \frac{h_j^S}{h_i^S} = 1,$$

which in turn gives for each $i \in B(h^S)$:

$$f_i(u_S) = h_i^S \tag{3.1}$$

One easily checks that, conversely, the proposed solution satisfies efficiency and $T_{N,L}$-hierarchy. Thus, f is uniquely determined on unanimity graph games. Because the collection of unanimity graph games is a basis for $\mathcal{C}_{N,L}$, there exist $a_S \in \mathbb{R}$, $S \in C(L)\backslash\{\emptyset\}$, such that $v = \sum_{S \in C(L)\backslash\{\emptyset\}} a_S u_S$. Therefore, $f(v) = \sum_{S \in C(L)\backslash\{\emptyset\}} a_S f(u_s)$ by linearity and so f is uniquely determined on $\mathcal{C}_{N,L}$. □

Proposition 3.2 *For each non-empty set $T_{N,L}$, the average tree solution given by* (2.2) *satisfies efficiency, linearity and $T_{N,L}$-hierarchy on $\mathcal{C}_{N,L}$.*

Proof. Pick any $v \in \mathcal{C}_{N,L}$, any non-empty set $T_{N,L}$, and consider the corresponding average tree solution AT(v) given by (2.2).

Efficiency We first show that for each $t_i \in T_{N,L}$ and each $j \in N$, $\sum_{k \in S_j^{t_i}} m_k^{t_i}(v) = v(S_j^{t_i})$. We proceed by induction on the number of agents in $S_j^{t_i}$.

Initial step: assume that $S_j^{t_i} = \{j\}$. Then, $m_j^{t_i}(v) = v(\{j\}) - v(\emptyset)$. Thus, the assertion is true when j is a leaf of t_i.

Induction step: assume that the assertion holds for each $S_j^{t_i}$ with at most p agents, and pick any $S_j^{t_i}$ with $p+1$ agents. We get,

$$v(S_j^{t_i}) = m_j^{t_i}(v) + \sum_{k \in t_i^{-1}(j)} v(S_k^{t_i}).$$

By the induction hypothesis, $v(S_k^{t_i}) = \sum_{q \in S_k^{t_i}} m_q^{t_i}(v)$. Because $S_j^{t_i}$ is the disjoint union of $\{j\}$ and the sets $S_k^{t_i}$, $k \in t_i^{-1}(j)$, we obtain $\sum_{q \in S_j^{t_i}} m_q^{t_i}(v) = v(S_j^{t_i})$, as desired.

Because for each $t_i \in T_{N,L}$, $S_i^{t_i} = N$, we get $\sum_{j \in N} m_j^{t_i}(v) = v(N)$. By (2.2), we conclude that the average tree solution satisfies efficiency.

Linearity The average tree solution is linear as the average of $|T_{N,L}|$ marginal contribution vectors.

$T_{N,L}$**-hierarchy** Pick any connected coalition $S \in C(L)\backslash\{\emptyset\}$, any agent $i \in N$ and any $t_k \in T_{N,L}$. Observe the following facts:

1. If $i \neq r_{\hat{t}_k(S)}$, then either $S_i^{t_k} \not\supseteq S$ or $r_{\hat{t}_k(S)} \in S_i^{t_k}$. In both cases $m_i^{t_k}(u_S) = 0$.

2. If $i = r_{\hat{t}_k(S)}$, then $S_i^{t_k} \supseteq S$ and so $u_S(S_i^{t_k}) = 1$. Because $\hat{t}_k(S)$ is the smallest subtree of t_k that contains S, we have $S_j^{t_k} \not\supseteq S$ and so $u_S(S_j^{t_k}) = 0$ for each $j \in t_k^{-1}(i)$. Hence, $m_i^{t_k}(u_S) = 1$.

From (1) and (2), we get for each $i \in N$,

$$\begin{aligned}
\mathrm{AT}_i(u_S) &= \frac{1}{|T_{N,L}|}\left(\sum_{\substack{t_k \in T_{N,L}: \\ i = r_{\hat{t}_k(S)}}} m_i^{t_k}(u_S) + \sum_{\substack{t_k \in T_{N,L}: \\ i \neq r_{\hat{t}_k(S)}}} m_i^{t_k}(u_S)\right) \\
&= \frac{1}{|T_{N,L}|} \sum_{\substack{t_k \in T_{N,L}: \\ i = r_{\hat{t}_k(S)}}} 1 \\
&= h_i^S \\
&= f_i(u_S) \qquad (3.2)
\end{aligned}$$

where the last equality follows from (3.1). By proposition 3.1, we conclude that AT satisfies $T_{N,L}$-hierarchy. This completes the proof of proposition 3.2. $\square$

By proposition 3.1 and proposition 3.2, the average tree solution with respect to $T_{N,L}$ can be written as

$$\mathrm{AT}_i(v) = \sum_{\substack{S\in C(L):\\ i\in B(h^S)}} h_i^S a_S \tag{3.3}$$

for each $v = \sum_{S\in C(L)\setminus\{\emptyset\}} a_S u_S$ in $\mathcal{C}_{N,L}$ and each $i \in N$.

4 Sets of admissible trees

The study of the previous section holds for any non empty set of rooted spanning trees chosen as admissible trees. Here, we examine the relative interest of $\mathrm{AT}(v)$ for various sets of admissible trees.

4.1 Cycle-free graphs

The basic idea is to consider the set $T^a_{N,L}$ of all possible rooted spanning trees of a graph (N, L). This has been done by HLT [6] for cycle-free graph games. In this context, each agent i induces exactly one rooted spanning tree t_i in the following way: $t_i(i) = i$ and for each sequence of distinct agents $(i_1, \ldots, i_p)$ such that $i_1 = i$ and $\{i_q, i_{q+1}\} \in L$ for $q = 1, \ldots, p-1$, set $t_i(i_{q+1}) = i_q$. Hence, $T^a_{N,L}$ contains exactly n elements. It has been shown (HLT [6], Theorem 5.1) that, for each $i \in N$, the corresponding average tree solution can be written as

$$\mathrm{AT}_i(v) = \frac{1}{n}\sum_{t_i\in T^a_{N,L}} m^{t_i}(v) = \sum_{\substack{S\in C(L):\\ i\in S}} \frac{1+p^L_S(i)}{|S| + \sum_{j\in S} p^L_S(j)} a_S \tag{4.1}$$

where $p^L_S(j)$, $j \in S$, is the number of agents outside S that j represents. An agent $j \in S$ represents agent k outside S if k is connected to j and on the unique path connecting j and k all agents between j and k are outside S. Denote by $P^L_S(j)$ the set of agents outside S that $j \in S$ represents. Because the graph (N, L) is cycle-free, it holds that $B(h^S) = S$, and it is not difficult to verify that for each $i \in S$,

$$h_i^S = \frac{1+p^L_S(i)}{|S| + \sum_{j\in S} p^L_S(j)} \tag{4.2}$$

Firstly, note that $\{P_S^L(j)\}_{j\in S}$ forms a partition of $N\backslash S$, i.e $P_S^L(j) \cap P_S^L(i) = \emptyset$ for each $i, j \in S$, $i \neq j$, and $\cup_{j\in S} P_S^L(j) = N\backslash S$. It follows that $|S|+\sum_{j\in S} p_S^L(j) = n$. Secondly, for each $i \in S$ there is a unique $t_i \in T^a_{N,L}$ and agent i is such that $r_{\hat{t}_i(S)} = i$. This corresponds to the unit in the numerator of (4.2). Thirdly, for each $t_k \in T^a_{N,L}$, $\hat{t}_k(S) = t_k(S)$. If i represents k, then S is a subset of the set of subordinates of i in t_k so that $r_{t_k(S)} = i$. If i does not represent k, there exists an agent $j \in S$ who represents k and so i is a subordinate of j in t_k. This implies that $i \neq r_{t_k(S)}$. Conclude that $r_{t_k(S)} = i$ if and only if i represents k. Therefore, $|\{t_k \in T^a_{N,L} \,|\, i = r_{\hat{t}_k(S)}\}| = 1 + p_S^L(i)$. Thus, (4.2) holds.

4.2 Harsanyi trees

In [7], HLTY consider the average tree solution with respect to a specific set of admissible rooted spanning trees constructed as follows. Let $B = (B_i)_{i\in N}$ be a collection of coalitions satisfying the following conditions:

1. For each $i \in N$, it holds that $i \in B_i$ and $B_i \in C(L)$;

2. For all $i, j \in N$, $i \neq j$, it holds that either $B_i \subseteq B_j\backslash\{j\}$ or $B_j \subseteq B_i\backslash\{i\}$ or both $B_i \cap B_j = \emptyset$ and $B_i \cup B_j \notin C(L)$;

3. For each $i \in N$ and each connected component C of the subgraph of (N, L) induced by the set of agents $B_i\backslash\{i\}$, it holds that $C = B_j$ for some $j \in N$ such that $\{i, j\} \in L$.

Any collection $B = (B_i)_{i\in N}$ satisfying conditions 1, 2 and 3 induces a unique rooted spanning tree, say t_i^B, in such a way that (j, k) is a directed edge of t_i^B if and only if B_k is a component of $B_j\backslash\{j\}$. Therefore, t_i^B is such that $S_j^{t_i^B} = B_j$ for each $j \in N$. Denote by $T^b_{N,L}$ the set of such of rooted spanning trees for the communication graph (N, L). For reasons that will appear shortly, an element of $T^b_{N,L}$ will be called a *Harsanyi tree*. The following proposition provides a simple characterization of the set $T^b_{N,L}$, which will prove useful throughout this section.

Proposition 4.1 *Let (N, L) be a graph on N. A rooted spanning tree t_k belongs to $T^b_{N,L}$ if and only if for each $\{i, j\} \in L$ it holds that either $i \in S_j^{t_k}$ or $j \in S_i^{t_k}$.*

Proof. Pick any $t_k \in T^b_{N,L}$ and any $\{i,j\} \in L$. We have to show that either $i \in S_j^{t_k}$ or $j \in S_i^{t_k}$. Consider the unique agent $r \in N$ such that $\{i,j\} \subseteq B_r = S_r^{t_k}$ and for any other agent $p \in N$, where $\{i,j\} \subseteq B_p = S_p^{t_k}$, we have $r \in S_p^{t_k}$. Assume that $r \notin \{i,j\}$. Because $\{i,j\} \in L$, condition 3 described above implies that there exists a successor of r, say s_r, such that $\{i,j\} \subseteq S_{s_r}^{t_k} = B_{s_r}$. But, we have a contradiction with the definition of r. We conclude that $r \in \{i,j\}$, which gives the desired result.

For the converse part, pick any rooted spanning tree t_k of (N,L) such that for each $\{i,j\} \in L$, it holds that either $i \in S_j^{t_k}$ or $j \in S_i^{t_k}$. We have to show that the collection of coalitions $(S_1^{t_k}, \ldots, S_n^{t_k})$ satisfies conditions 1, 2 and 3 described above.

Condition 1 follows from the definition of $S_i^{t_k}$, $i \in N$.

By definition of a rooted spanning tree, for each pair of distinct agents $\{i,j\}$, it holds that either $S_i^{t_k} \subseteq S_j^{t_k} \backslash \{j\}$ or $S_j^{t_k} \subseteq S_i^{t_k} \backslash \{i\}$, or $S_i^{t_k} \cap S_j^{t_k} = \emptyset$. Assume that there is a pair of distinct agents $\{i,j\}$ such that $S_i^{t_k} \cap S_j^{t_k} = \emptyset$. Then, for each $i_c \in S_i^{t_k}$ and each $j_c \in S_j^{t_k}$, we have $i_c \notin S_{j_c}^{t_k}$ and $j_c \notin S_{i_c}^{t_k}$ and so $\{i_c, j_c\} \notin L$. Therefore, $S_i^{t_k} \cup S_j^{t_k}$ can not be a connected coalition of (N,L). We conclude that condition 2 holds.

Pick any $i \in N$ and consider the subgraph $(S_i^{t_k} \backslash \{i\}, L(S_i^{t_k} \backslash \{i\}))$ of (N,L) induced by $S_i^{t_k} \backslash \{i\}$. Assume, for the sake of contradiction, that there exists a connected component C of $(S_i^{t_k} \backslash \{i\}, L(S_i^{t_k} \backslash \{i\}))$ such that $C \neq S_j^{t_k}$ for each $j \in t_k^{-1}(i)$. Then, there necessarily exists a set of distinct agents $\{j_1, j_2, \ldots, j_q\}$ included in $t_k^{-1}(i)$ such that $\{S_{j_1}^{t_k}, S_{j_2}^{t_k}, \ldots, S_{j_q}^{t_k}\}$ forms a partition of C. But, we have a contradiction with condition 2, so condition 3 holds. □

Corollary 4.2 *If (N,L) is a cycle-free graph, then the set of all rooted spanning trees coincides with the set of Harsanyi trees. If (N,L) is the complete graph K_N, then the set of Harsanyi trees coincides with the set of line-trees, i.e. the set of all rooted spanning trees where each agent has at most one successor.*

From corollary 4.2, two properties of the corresponding average tree solution emerge. These properties, already proved by HLTY [7], are contained in the following proposition.

Proposition 4.3 *(HLTY [7], Theorem 3.2 and Theorem 3.3) If (N,L) is a cycle-free graph, then, for each $v \in \mathcal{C}_{N,L}$, the average tree solution defined with* *respect to $T^b_{N,L}$ and given by (2.2) is the average of n marginal contribution vectors and coincides with (4.1). If (N,L) is the complete graph K_N, then, for $v \in \mathcal{C}_{K_N}$, the average tree solution defined with respect to $T^b_{N,L}$ and given by (2.2) is the average of $n!$ marginal contribution vectors and coincides with the Shapley value given by (2.1).*

A *sharing system* on N is a system $z = (z^S)_{S\in C(L)\setminus\{\emptyset\}}$, where each sharing vector $z^S \in \mathbb{R}^n_+$ is defined as: $z^S_i = 0$ for each $i \in N\setminus S$, $z^S_i \geq 0$ for each $i \in S$ and $\sum_{i\in N} z^S_i = 1$. A solution f is a *Harsanyi solution* if it assigns to each game $v \in C_{N,L}$ and to each $i \in N$ the payoff

$$f_i(v) = \sum_{\substack{S\in C(L):\\ i\in S}} z^S_i a_S$$

for some sharing system z. Harsanyi solutions have been proposed by Vasil'ev [11] and studied for line-graph games by van den Brink, van der Laan and Vasil'ev [3] (see also van den Brink, van der Laan and Pruzhansky [2]). The system $h = (h^S)_{S\in C(L)\setminus\{\emptyset\}}$ defined in section 3 is not necessarily a sharing system since $B(h^S)$ may contain agents outside S. In case $B(h^S) \subseteq S$, the system h is a sharing system. Proposition 4.4 below points out another advantage of considering Harsanyi trees as the set of admissible trees. It states that the set of Harsanyi trees is the largest set of rooted spanning trees such that the corresponding average tree solution is a Harsanyi solution.

Proposition 4.4 *Let (N,L) be a graph on N and assume that the set of admissible trees is the set of Harsanyi trees $T^b_{N,L}$. Then, we have: (i) for each $S \in C(L)\setminus\{\emptyset\}$, $B(h^S) = S$. (ii) The set $T^b_{N,L}$ is the largest set of rooted spanning trees of (N,L) such that (i) holds. (iii) For each $v = \sum_{S\in C(L)\setminus\{\emptyset\}} a_S u_S$ in $\mathcal{C}_{N,L}$ the average tree solution on $T^b_{N,L}$ assigns to each $i \in N$,*

$$AT_i(v) = \sum_{\substack{S\in C(L):\\ i\in S}} h^S_i a_S \tag{4.3}$$

Proof. (i) Consider the set $T^b_{N,L}$ of Harsanyi trees. Pick any $S \in C(L)\setminus\{\emptyset\}$. We first show that $B(h^S) \supseteq S$ by proving that, for each agent $i \in N$, there is at least one spanning tree t_i rooted at i such that $t_i \in T^b_{N,L}$. We proceed by induction on the number n of agents in N.

Initial step: if $n = 1$, we are done.
Induction step: assume that the assertion holds for each set of agents N with at most n agents, and pick N with $n+1$ agents. Consider a communication graph (N, L), an agent $i \in N$ and the connected components of the subgraph $(N\backslash\{i\}, L(N\backslash\{i\}))$ of (N, L) induced by $N\backslash\{i\}$. For each connected component C of $(N\backslash\{i\}, L(N\backslash\{i\}))$, there is $j \in C$ such that $\{i, j\} \in L$. By the induction hypothesis, there is a spanning tree t_j^C rooted at j such that $t_j \in T^b_{C,L(C)}$. For each $t_j^C \in T^b_{C,L(C)}$, construct the directed edge (i, j) and add it to the tree t_j^C. By construction, the resulting directed graph on N is a spanning tree rooted at i and it is easy to see that it belongs to $T^b_{N,L}$. We conclude that each $i \in N$ is the root of at least one element of $T^b_{N,L}$, so that $B(h^S) \supseteq S$.

It remains to show that $B(h^S) \subseteq S$. Assume, for the sake of contradiction, that there is $i \in B(h^S)\backslash S$. Then, there is $t_k \in T^b_{N,L}$ such that $i = r_{\hat{t}_k(S)}$. Consider the subgraph $t_k(S)$ of t_k induced by S. It follows that $t_k(S)$ is a forest, and for any pair of agents in S belonging to distinct components of $t_k(S)$, one agent of this pair cannot be the subordinate of the other in t_k. Because $S \in C(L)$, there is at least one such a pair of agents incident to the same edge in (N, L). By proposition 4.1, $t_k \notin T^b_{N,L}$, a contradiction. Conclude that $B(h^S) \subseteq S$. Because S was an arbitrary non-empty coalition in $C(L)$, we have established that $B(h^S) = S$ for each $S \in C(L)\backslash\{\emptyset\}$.

(ii) The proof of point (i) establishes that if $t_k \in T^b_{N,L}$, then $r_{\hat{t}_k(S)} \in S$ for each $S \in C(L)\backslash\{\emptyset\}$. Conversely, if $r_{\hat{t}_k(S)} \in S$ for each $S \in C(L)\backslash\{\emptyset\}$, then $t_k \in T^b_{N,L}$. To see this, assume that $r_{\hat{t}_k(S)} \in S$ for each $S \in C(L)\backslash\{\emptyset\}$ and $t_k \notin T^b_{N,L}$. By proposition 4.1, there exists $\{i, j\} \in L$ such that neither $i \in S_j^{t_k}$ nor $j \in S_i^{t_k}$. This implies that $r_{\hat{t}_k(\{i,j\})} \notin \{i, j\} \in C(L)$, a contradiction. From this, we obtain that $T^b_{N,L}$ is the largest set of rooted spanning trees of (N, L) for which $B(h^S) = S$ for all $S \in C(L)\backslash\{\emptyset\}$.

(iii) Equation (4.4) is just a consequence of the fact that, for all $S \in C(L)$, $B(h^S) = S$ in equation (3.3). □

4.3 Triangle-free trees

In this section we introduce the set of triangle-free trees. A rooted spanning tree t_k of (N, L) is *triangle-free* if $\{i, j\} \notin L$ whenever $t_k(i) = t_k(j)$. Denote by $T^*_{N,L}$ the set of triangle-free trees of (N, L). Note that for each $i \in N$, there is a least one triangle-free tree rooted at i. An application of proposition 4.1

gives $T^*_{N,L} \supseteq T^b_{N,L}$.

Example 4.5

Assume that $N = \{1, 2, 3, 4\}$, $L = \{\{1, 2\}, \{1, 3\}, \{2, 4\}, \{3, 4\}\}$ and construct t_3 with the set of directed edges $\{(3, 1), (3, 4), (1, 2)\}$. It so happens that $t_3 \in T^*_{N,L} \backslash T^b_{N,L}$.

One of the advantages of the set $T^*_{N,L}$ is that it coincides with $T^a_{N,L}$ for the class of clique-free graphs. A graph (N, L) is *clique-free* if for any coalition S, $|S| = 3$, the induced subgraph $(S, L(S))$ is different from K_S. Hypercubes, tori, the Petersen graph, among others, are clique-free graphs. On the contrary, $T^a_{N,L} = T^b_{N,L}$ if and only if (N, L) is cycle-free. To see this, consider any cyclic graph (N, L) and $i \in N$ involved in a cycle $(C, L(C))$. As in example 4.5, one can construct a rooted spanning tree t_k such that i has two successors in $t_k(C)$. Obviously, such a tree is not a Harsanyi tree. Proposition 4.1 yields $T^*_{N,L} = T^b_{N,L}$ in the two extreme cases when (N, L) is either cycle-free or the complete graph K_N. Hence, as in HLTY [7], we directly obtain:

Proposition 4.6 *If (N, L) is a cycle-free graph, then, for each $v \in \mathcal{C}_{N,L}$, the average tree solution defined with respect to $T^*_{N,L}$ and given by (2.2) is the average of n marginal contribution vectors and coincides with (4.1). If (N, L) is the complete graph K_N, then, for $v \in \mathcal{C}_{K_N}$, the average tree solution defined with respect to $T^*_{N,L}$ and given by (2.2) is the average of $n!$ marginal contribution vectors and coincides with the Shapley value given by (2.1).*

We now examine the two sets $T^*_{N,L}$ and $T^b_{N,L}$ from a computational point of view. We adopt the classical framework of asynchronous distributed computing (see Attiya and Welch [1]). Suppose that the agents have to decide on the formation of some rooted spanning tree, knowing the underlying communication graph. Agents communicate by sending messages over communication links. An algorithm for this message-passing system consists of a local program for each agent. A local program for an agent provides the ability for the agent to perform local computation and to send messages to and receives messages from each of its neighbors in the graph. The question concerns which rooted spanning trees can be formed through such an algorithm.

General framework

Given a graph (N, L), each agent $i \in N$ is modeled as a finite state machine with state set X_i. The state set X_i contains a non empty subset of *initial states*. A *configuration* is a vector of states $x = (x_1, \ldots, x_n)$ where $x_i \in X_i$ for each $i \in N$. An initial configuration is a configuration whose states are all initial. Each state of agent $i \in N$ contains at least $2|L_i|$ components, $\mathsf{inbuf}_i(j)$ and $\mathsf{outbuf}_i(j)$ for each $j \in L_i$. These components are sets of messages: $\mathsf{inbuf}_i(j)$ holds the last message that has been delivered to i through the link $\{i, j\}$; $\mathsf{outbuf}_i(j)$ holds the last message that $i \in N$ has sent to j through the link $\{i, j\}$. These messages contain local information in the sense that it concerns exclusively the neighborhood of the senders. We model the fact that any message sent by agent i to agent j is immediately delivered, i.e. $\mathsf{inbuf}_j(i) = \mathsf{outbuf}_i(j)$. Each state can also contain other *internal state variables* that can be used by agent i for the local computation. The agent's state, excluding the $\mathsf{outbuf}_i(j)$ components, constitutes the *accessible state* of i. A *transition* on agent i is an ordered pair (s_i, x_i), where s_i, the *input* of the transition, is an accessible state of i and x_i, the *output* of the transition, is a (complete) state of i. Each component $\mathsf{outbuf}_i(j)$ of x_i is the message sent by i to j during the transition. The transition is *passive* if the state of agent i is identical to the old one (in particular no new message is delivered to his or her neighbors). An algorithm is given by a set of transitions.

A computation step or *event* on agent i corresponds to the application of i's transition function to its current accessible state. Formally, an event on agent i is a pair formed by a configuration $x = (x_1, \ldots, x_n)$ and a transition (s_i, y_i) on agent i such that s_i is the accessible state contained in x_i. The output of the event is the configuration $z = (z_1, \ldots, z_n)$ such that

- $z_i = y_i$;
- for each $j \in L_i$, z_j is identical to x_j except for the content of $\mathsf{inbuf}_j(i)$, which is given by the equality $\mathsf{inbuf}_j(i) = \mathsf{outbuf}_i(j)$;
- $z_j = x_j$ otherwise.

The first condition indicates that the state of i is updated; the second condition indicates that emitted messages by i are immediately transmitted to his neighbors; the third condition indicates that the state of the other agents are not updated.

In order to handle the case where two or more agents would like to simultaneously perform a local computation, we introduce a fictitious agent to act as chairman. The chairman activates the agents successively during the execution of the algorithm, which renders the system asynchronous. Formally, a *chairman* is an infinite sequence $(i^q)_{q\in\mathbb{N}}$ of agents. A chairman is said to be *fair* if for each $i \in N$, there is a constant subsequence $(i^{g(q)})_{q\in\mathbb{N}}$ equal to i. The fairness condition ensures that each agent will get the opportunity to perform local computations in the future.

An *execution* governed by the above chairman is an infinite sequence of the following form: $x^0, e^1, x^1, e^2, x^2, e^3, \ldots$, where for each integer $q \in \mathbb{N} \cup \{0\}$, x^q is a configuration, e^{q+1} is a transition, (x^q, e^{q+1}) is an event on agent i^{q+1}, x^{q+1} is the output of (x^q, e^{q+1}), and x^0 is initial. For the sake of consistency, the configuration x^q will be called the q^{th} configuration of the execution.

Such an execution is *fair* if its chairman is fair and for each agent not all transitions are passive. Here, the fairness condition ensures that each agent will not ignore indefinitely the received information. The execution *terminates* if there exists a positive integer q^* such that for $q > q^*$ the transition e^q is passive. In this case, the configuration x^q, $q > q^*$, is the *final configuration*. Note that our model has two sources of non determinism: the choice of a chairman and multiple transitions with the same input.

An algorithm for the construction of triangle-free trees

We describe an algorithm that constructs a triangle-free spanning tree t_r rooted at r, based on a graph (N, L), a fair chairman $(i^q)_{q\in\mathbb{N}}$ and a root r. The construction is *monotone* in the sense that once an edge has been added, its deletion will never be considered. To this end, two new types of variables are introduced. In addition to the inbuf and outbuf components, the state of each agent $i \in N$ contains:

- the integer variable $p(i)$, which holds either 0 or i's unique predecessor;
- a set variable $P(i)$, which holds 0 and the labels of the agents who can potentially become i's unique predecessor.

The pseudo-code of our algorithm is in **Algorithm 1**. In an initial configuration, $p(i) = 0$ for all $i \in N$ since no agent has a predecessor, and $P(i) = \{0\}$, which indicates that no agent in N is yet to become i's predecessor. Moreover, for each $i \in N$ and each $j \in L_i$, $\mathsf{inbuf}_i(j)$ and $\mathsf{outbuf}_i(j)$ are empty. Some transitions will change the original value of $p(i)$ into an agent's label. Because each transition whose input state is such that $p(i) = j$, $j \neq 0$, does not change the value of $p(i)$, our algorithm is monotone. A non empty message contains one agent's label, which forbids the recipient to take this agent as predecessor. Exceptions are messages sent by the root, which contain the label 0.

Lines 1, 2 and 3 of the algorithm encode the non passive transition on the root r. Lines 4, 5 and 6 encode how the set $P(i)$ is updated: first, neighbors of i previously put in the tree are added (line 5), afterwards each i's neighbor whose predecessor is an element of L_i is removed (line 6). Line 7 indicates how an agent chooses its predecessor. By lines 8 and 9, each neighbor j of i is informed that $p(i)$ will never become its predecessor.

Algorithm 1

Input: a finite graph (N, L).
Initial conditions: $\forall i \in N$, $p(i) = 0$, $P(i) = \{0\}$, all buffers are empty.

Transitions on agent i:

1: if $i = r$ and $p(r) = 0$, then
2: $\quad p(r) \longleftarrow r$,
3: $\quad \forall j \in L_r$, $\mathsf{outbuf}_r(j) \longleftarrow 0$,
4: if $i \neq r$ and $p(i) = 0$, then
5: $\quad P(i) \longleftarrow P(i) \cup \{j \in L_i \mid \mathsf{inbuf}_i(j) \neq \emptyset\}$,
6: $\quad P(i) \longleftarrow P(i) \backslash \{k \in N \mid \exists j \in L_i \text{ such that } \mathsf{inbuf}_i(j) \text{ contains } k\}$,
7: $\quad p(i) \longleftarrow l$ for some $l \in P(i)$,
8: $\quad$ if $l \neq 0$ then
9: $\quad\quad \forall j \in L_i$, $\mathsf{outbuf}_i(j) \longleftarrow l$,
10: else, the transition is passive.

Remark that the pseudo-code enforces the following local behavior: when $p(i) = 0$, no computation step has previously been done on i; when $p(i) \neq 0$, no computation step will be later be done on i. Therefore, for each execution, we have at most one active transition on each agent i.

Proposition 4.7 *Each execution of* **Algorithm 1** *terminates.*

Proof. Given an execution (not necessarily fair), consider the set N^q of agents i such that $p(i) \neq 0$ in the q^{th} configuration. The sequence $(N^q)_{q \in \mathbb{N} \cup \{0\}}$ is non decreasing. Because N is a finite set, there exists an integer q^* such that $N^{q^*} = N^q$ for $q > q^*$. For each $i \in N^{q^*}$, we have $p(i) \neq 0$ and so no more transition is possible on i. By definition, $p(i) = 0$ for each $i \notin N^{q^*}$ and each configuration, which means that no active transition has been done on i. Thus, the $q^{*^{th}}$ configuration is final. □

Proposition 4.8 *(i) For every fair execution,* **Algorithm 1** *constructs a rooted spanning tree $t_r \in T^*_{N,L}$. (ii) For each $t_r \in T^*_{N,L}$ and each fair chairman $(i^q)_{q\in\mathbb{N}}$, there exists an execution of* **Algorithm 1**, *governed by $(i^q)_{q\in\mathbb{N}}$, that constructs t_r.*

Proof. (i) For any $q \in \mathbb{N} \cup \{0\}$ and any $i \in N$, let $p^q(i)$ be the value of $p(i)$ taken in the q^{th} configuration. Let $N^q = \{i \in N \mid p^q(i) \neq 0\}$. We first prove by induction the following fact:

> **Fact 1** For each $q \in \mathbb{N} \cup \{0\}$, it holds that $p^q(N^q) \subseteq N^q$. For each $i \in N^q$, the sequence $(i^k)_{k\in\mathbb{N}\cup\{0\}}$ defined as $i^0 = i$ and $i^{k+1} = p^q(i^k)$, $k \in \mathbb{N} \cup \{0\}$, contains an element i^m such that $i^m = r$. Moreover, for any pair $\{i, j\} \in N^q \setminus \{r\} \times N^q \setminus \{r\}$ such that $p^q(i) = p^q(j)$, it holds that $\{i, j\} \notin L$.

Initial step: Because $N^0 = \emptyset$, **Fact 1** holds.
Induction step: Let $q \in \mathbb{N}$ and assume that N^q satisfies **Fact 1**. It is sufficient to consider the case $N^{q+1} \neq N^q$. By the pseudo-code of **Algorithm 1**, $p^q(i^{q+1}) = 0$, and there exists an agent $j \in N$ such that $p^{q+1}(i^{q+1}) = j$. Thus, we obtain $N^{q+1} = N^q \cup \{i^{q+1}\}$. From line 5 of **Algorithm 1**, we get $j \in N^q$. We conclude that $p^q(N^q) \subseteq N^q$ for every $q \in \mathbb{N} \cup \{0\}$. Note that no (directed) cycle can be created by addition of a pending (directed) link. This remark and the induction hypothesis together yield that the desired sequence of agents exists.

Now, pick any pair $\{i, j\}$ in $(N^{q+1}\setminus\{r\}\times N^{q+1}\setminus\{r\})\cap L$ and assume, for the sake of contradiction, that $p^{q+1}(i) = p^{q+1}(j)$. By the induction hypothesis, the set N^q satisfies **Fact 1**. Thus, one agent of this pair, say agent i, is such that $i = i^{q+1}$. Consider the event when j takes $l = p^{q+1}(i)$ as predecessor. This event occurs before the $(q + 1)^{th}$ event. After this event, $\mathsf{inbuf}_{i^{q+1}}(j)$ contains l by line 9 and definition of an event. By lines 6 and 7 of **Algorithm 1**, we cannot have $p(i^{q+1}) \longleftarrow l$ at the $(q + 1)^{th}$ event, a contradiction. Therefore, $p^{q+1}(i) \neq p^{q+1}(j)$, as desired. This completes the proof of **Fact 1**.

Lastly, because the execution is fair, not all transitions are passive for each agent. Thus, there exists $q^* \in \mathbb{N}$ such that $N^q = N$ for each $q > q^*$, and the corresponding set $A^q = \{(p^q(i), i) \,|\, i \in N \setminus \{r\}\}$ constitutes the set of all directed links of a rooted spanning tree $t_r \in T^*_{N,L}$. This completes the proof of point (i).

(ii) Pick any rooted spanning tree t_r in $T^*_{N,L}$. There exists a permutation σ of N such that for each agent i, where $i \neq r$, we have $\sigma(i) > \sigma(t_r(i))$. Note that $\sigma(r) = 1$ for every such permutation. Since $(i^q)_{q\in\mathbb{N}}$ is fair, there exists a n-uple $(k_1, k_2, \ldots, k_n)$ such that for each agent i, $i^{k_i} = i$, k_r is the lowest value such that $i^{k_r} = r$, and for each pair $\{i, j\}$ of distinct agents such that $\sigma(i) > \sigma(j)$, it holds that $k_i > k_j$.

We claim that there exists an execution such that, for each $i \in N$, the $(k_i)^{th}$ event is an active transition on i which makes: $p^{k_i}(i) \longleftarrow t_r(i)$ (all other transitions being passive). To show this claim, it suffices to prove **Fact 2** below. We proceed by induction on the value of $\sigma(i)$.

> **Fact 2** The k_i first events are possible. After the $(k_i)^{th}$ event, it holds that $\{j \in N \,|\, p^{k_i}(j) \neq 0\} = \{j \in N \,|\, \sigma(j) \leq \sigma(i)\}$. Therefore, $p^{k_i}(j) = t_r(j)$ for each such agent j.

Initialisation: If $\sigma(i) = 1$, then $i = r$. By definition of k_r each configuration preceding the $(k_r)^{th}$ event is initial. It follows that the desired transition, which yields $p^{k_r}(r) \longleftarrow r$, can be done at the $(k_r)^{th}$ event.

Induction hypothesis: Let p be an integer such that $1 \leq p < n$. Let i be the integer such that $\sigma(i) = p + 1$, and assume that **Fact 2** is true for each integer j such that $\sigma(j) \leq p$. The unique event on $t_r(i)$ before the $(k_i)^{th}$ configuration makes $\mathsf{inbuf}_i(t_r(i))$ not empty. Because t_r is triangle-free, there is no other agent in L_i whose predecessor is $t_r(i)$. This property and the induction hypothesis together yield that no buffer $\mathsf{inbuf}_i(j)$ contains $t_r(i)$. It follows that $t_r(i)$ is added in $P(i)$ during the $(k_i)^{th}$ event by line 5 of **Algorithm 1**, which in turn allows to make $p^{k_i}(i) \longleftarrow t_r(i)$ during the same event. This completes the proof of point (ii). $\square$

A counterexample for Harsanyi trees

As shown by the example below, it is not possible to establish a similar result for the set of Harsanyi trees.

Proposition 4.9 *There does not exist a (monotone) message-passing algorithm such that any fair execution constructs a rooted Harsanyi tree.*

Proof. Assume, for the sake of contradiction, that there exists such an algorithm. Inputs can be any connected graph (N, L), any fair chairman $(i^q)_{q\in\mathbb{N}}$ and any root $r \in N$. In particular, we can consider the undirected graph (N, L) given by $N = \{1, 2, 3, 4, 5\}$ and $L = \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 5\}\}$, agent 3 as the root and any fair chairman such that the first three activated agents are $i_1 = 3$, $i_2 = 2$ and $i_3 = 4$. Then, there exists a fair execution of the algorithm which constructs t_3, the unique spanning tree on (N, L) rooted at agent 3. Further, note that there must exist a fair execution of the algorithm such that the edge $(3, 2)$ is added in the current tree before the edge $(4, 5)$. Therefore, there is a configuration during this particular execution of the algorithm such that the tree constructed so far contains both edges $(3, 2)$ and $(3, 4)$.

Now, execute the same algorithm with inputs given by $(N, L \cup \{\{1, 5\}\})$, the same fair chairman as above and root 3. Consider the same execution of the algorithm as before. Because players 2, 3 and 4 have exactly the same neighbors in $(N, L \cup \{\{1, 5\}\})$ and (N, L), the configurations of the algorithm must be the same as with (N, L) until both edges $(3, 2)$ and $(3, 4)$ have been added to the current tree. However, by proposition 4.1, there is no element of $T^b_{N,L\cup\{\{1,5\}\}}$ containing these two edges, a contradiction. □

5 Conclusion

In this paper we studied average tree solutions for graph games. Each such a solution is defined with respect to a set of admissible rooted spanning trees. Given a set of admissible rooted spanning trees, we associated a system that is used to distribute the Harsanyi dividends over the agents in the population. For each connected coalition and each agent, this system measures the strategic location on the graph of this agent in terms of information that he or she can receive from this connected coalition. We gave an axiomatic characterization of the average tree solutions on the class of all graph games and showed that the payoff of an agent is equal to the sum of its shares in the dividends of all connected coalitions for which his or her location is strategic.

Although any set of admissible rooting spanning trees may be considered, we focused our attention on the set of Harsanyi trees and on the set of triangle-free trees. We showed that that the average tree solution defined with respect to the set of Harsanyi trees is a Harsanyi solution. This is not the case for the average tree solution defined with respect to the set of triangle-free trees. On the other hand, both solutions can be viewed as generalizations of the Shapley value and of the average tree solution for cycle-free graph games as introduced in HLT [6]. The set of triangle-free trees is larger than the set of Harsanyi trees, which means that there are more ways to transmit information in the direction of a particular agent in the former case. On the class of clique-free graphs, the set of triangle-free trees coincides with the set of all rooted spanning trees. In case the set of admissible rooted spanning trees is the set of Harsanyi trees, the above property holds if and only if the underlying graph is cycle-free. Finally, we compared both admissible sets of rooted spanning trees from a computational point a view. We constructed a distributed algorithm for searching triangle-free trees. This algorithm is a growth process of a spanning tree. Starting at the designated root, the tree is grown by adding at most one (directed) edge at each iteration. Each computation step relies only on local information received from neighbors so that the resulting triangle-free tree is, in some sense, the result of agent computation but not of agent design. We showed that no such distributed algorithm for searching Harsanyi trees exists.

References

[1] Attiya H., J. Welch. *Distributed computing. Fundamentals, simulations and advanced topics*. Wiley-Interscience, New Jersey (2004).

[2] Brink, R. van den, G. van der Laan, V. Pruzhansky. Harsanyi power solutions for graph-restricted games. Tinbergen Discussion Paper 04/095-1, Tinbergen Institute and Free University, Amsterdam.

[3] Brink, R. van den, G. van der Laan, V. Vasil'ev. Component efficient solutions in line-graph games with applications. *Economic Theory* **33** (2007) 349-364.

[4] Faigle U., W. Kairn. The Shapley value for cooperative games under precedence constraints. *International Journal of Game Theory* **21** (1992) 249-266.

[5] Gilles R., G. Owen, R. van den Brink. Games with permission structures: the conjunctive approach. *International Journal of Game Theory* **20** (1992) 277-293

[6] Herings P., G. van der Laan, D. Talman. The average tree solution for cycle free games. *Games and Economic Behavior* **62** (2008) 77-92.

[7] Herings P., G. van der Laan, D. Talman, Z. Yang. The average tree solution for cooperative games with limited communication structure. Preprint (2008).

[8] Lange F., M. Grabisch. Values on regular games under Kirchhoff's laws. Cahiers de la MSE (2006) 2006.87 1-21.

[9] Myerson R. Graphs and cooperation in games. *Mathematics of Operations Research* **2** (1977) 225-229.

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

[11] Vasil'ev V. On a class of operators in a space of regular set functions. *Optimizacija* **28** (1982) 102-111 (in russian).