On the existence of Nash Equilibria in Strategic Search Games*

C. Àlvarez[1], A. Duch[1], M. Serna[1], D.M. Thilikos[2]

[1] ALBCOM Research Group. Technical University of Catalonia
[2] Department of Mathematics. National and Kapodistrian University of Athens.

Abstract. We consider a general multi-agent framework in which a set of n agents are roaming a network where m valuable and sharable goods (or resources or services or information) are hidden in m different vertices of the network. We analyze several strategic situations that arise in this setting by means of game theory. To do so we introduce a class of strategic search games. In such a game each agent has to select a simple path in the network that starts from a predetermined set of initial vertices. Depending on how the value of the retrieved goods is splitted among the agents we consider two game types: *finders-share* in which the agents that find a good split among them the corresponding benefit and *firsts-share* in which only the agents that first find a good share the corresponding benefit. We show that finders-share games always have pure Nash equilibria (PNE). For obtaining this result we introduce the notion of *Nash preserving reduction* between strategic games. We show that finders-share games are Nash reducible to single-source network congestion games. This is done through a series of Nash preserving reductions. For firsts-share games we show the existence of games with and without PNE. Furthermore we identify some graph families in which the firsts-share game has always a PNE that is computable in polynomial time.

1 Introduction

In the classical setting search games are intended to look upon the situation as a game between a searcher and a hider and the aim of the analysis is to provide optimal strategies for the participants [4, 3]. That is strategies that allow the searcher to find the hider and the hider to avoid the searchers. In our approach we are interested in analyzing the strategic situation that arises when a set of hiders do not move and a set of searchers set their strategies in a selfish way considering economical benefits and rewards. We consider a general framework of strategic search in which a set of n mobile agents are roaming a network where m valuable items or resources or information are hidden in m different

* The first and third authors were partially supported by TIN-2007-66523 (FORMALISM). The second author was supported by TIN-2007-11345 (ALINEX-2). The fourth author was supported by the project "Kapodistrias" (AΠ 02839/28.07.2008) of the National and Kapodistrian University of Athens (project code: 70/4/8757).

vertices. We want to take into consideration different aspects that affect the agents decisions and rewards in order to analyze the existence of equilibria. This framework differs from other resource sharing strategic games considered in the literature, in particular from the well known framework of *congestion games* [11, 8]. In this initial work we concentrate in analyzing the existence or not of pure Nash equilibria in a static draw of the proposed games, before defining the games, we consider the main parameters and take some initial decision for the model.

Benefit? Benefit depends on one side on the cost that the agents have to pay for traversing network links and on the other in the way in which the rewards or the value of the goods found by the agent is distributed among the agents that discover the same good. We consider two natural reward models. When the good is non portable any agent that discovers it will get some benefit. When the good is portable only agents that arrive for the first time to the good location can benefit from the discovery. Therefore we consider two game variants: The *finders-share game* in which the item value is splitted equitably among all the players that discover it and the *firsts-share game* in which the item value is shared only among all the agents that discover the item first (all of them at the same time).

Where do the agents start their roaming? We consider two different possibilities: Players start their roaming at one initial vertex or can choose one from a set of initial vertices. In both cases we consider the particular case in which the initial vertices (or set of vertices) is the same for all the players.

What is the cost for the agents? It seems natural that they have to incur some cost in traversing a link. This cost might arise as the cost of communication or movement. We assume that each link in the network has associated a non negative cost. To any agent's trajectory we associate as cost the sum of the cost of the edges present in it.

How the players move? We consider different kinds of trajectories. Initially we study the problems assuming that the players strategy is formed by the selection of a simple path (without repeated nodes) in the network. We analyze also finders-share games under two other trajectories: paths, now nodes can be repeated but edges can not appear twice, and trees. When the trajectory is a path, a player can pass more that once through one edge in order to access additional valuable resources. The tree trajectory arises naturally assuming that the agents are buying the links in their trajectories, so that they can cross them as many times as they wish without additional payment, thus avoiding cycles.

We show that finders-share games in which the players are restricted to select a simple path always have pure Nash equilibria (PNE). This result is independent of the type of initial location or on whether the network is directed or undirected. For doing so we introduce the notion of *Nash preserving reduction* between strategic games. This is an appropriate extension of traditional reducibility among problems. Those reductions preserve the existence of PNE and the fact that a PNE can be computed in polynomial time. We show that finders-share games are Nash reducible to single-source network congestion games. This is done through a series of Nash preserving reductions. First, by a series of transformations, we reduce the general case to the single-source finders-share game. Finally, the single-source finders-share game is reduced to the single-source network congestion game. These reductions guarantee also the property that a PNE can be computed in polynomial time.

For the firsts-share games in which the players are restricted to select a simple path we show the existence of games with and without PNE, for different variations of the type of game. Furthermore, we identify some graph families in which the firsts-share game has always a PNE. In those cases we provide algorithms for computing a PNE in polynomial time.

Finally we consider the two variations on the trajectories, allowing paths with repeated nodes or allowing trees. We show that in both cases the finders-share games can be Nash reduced to congestion games. This reduction shows the existence of PNE but leaves open the existence or not of a polynomial time algorithm for computing a PNE for such games.

2 Definitions and preliminaries

All through the paper we use the standard graph notation and in particular we consider that for an undirected graph: A *walk* is a sequence of vertices such that for each pair of consecutive vertices the corresponding edge is present in the graph. A *path* is a walk in which none of the edges appears twice. A *simple path* is a walk in which none of the vertices appears twice.

In the case of considering arcs instead of edges we add to the name of these sequences the adjective *directed* (*directed* walk, *directed* path and *directed* simple path, respectively).

A *strategic game* $\Gamma = (N, (\Pi_i)_{i\in N}, (u_i)_{i\in N})$ is defined by a finite set of *players or agents* $N = \{1, \ldots, n\}$, a finite set of *strategies* (or actions) Π_i, for each agent $i \in N$, and a *payoff function* $u_i : \Pi \to \mathbb{R}$, for each player $i \in N$. Define the set $\Pi = \times_{i\in N}\Pi_i$, every element $(p_1, \ldots, p_n) \in \Pi$ is known as a *pure strategy profile* or *configuration* and represents a possible outcome of the game. We also denote Π of Γ by $\Pi(\Gamma)$.

Given a profile $\pi = (p_1, \ldots, p_n)$, p_i represents the strategy followed by agent $i \in N$. In addition, it is usual to denote by (π_{-i}, p), with $i \in N$, the profile that we obtain substituting the i-th element of π (p_i) by p. A *Pure Nash Equilibrium* (PNE, for short) is a configuration $\pi = (\pi_1, \ldots, \pi_n)$ such that for each agent $i \in N$ $u_i(\pi) \geq u_i((\pi_{-i}, p))$ for any $p \in \Pi_i$. We denote as PNE(Γ) the set of pure Nash equilibria of game Γ.

A *congestion game* is defined by a tuple $\Gamma = (N, E, (\Pi_i)_{i\in N}, (d_e)_{e\in E})$ where $N = \{1, \ldots, n\}$ is the set of players, E is a finite set of resources, $\Pi_i \subset \mathcal{P}(E)$ is the set of allowed actions for each player $i \in N$, and $d_e : \mathbb{N} \to \mathbb{R}$ is the delay function of each resource $e \in E$, which is assumed to be polynomial-time computable and models the delay $d_e(k)$ provoked by resource e under a congestion $k \in \{1, \ldots, n\}$. $d_e(k)$ is nondecreasing in k. Let $\Pi = \times_{i\in N}\Pi_i$. For all $\pi = (p_1 \ldots, p_n) \in \Pi$ and for every $e \in E$ let $\omega_e(\pi)$ be the number of users of resource e according to the configuration π, $\omega_e(\pi) = |\{i \in N : e \in p_i\}|$. Each player $i \in$ has associated a cost function $c_i : \Pi \to \mathbb{R}$ defined by

$$c_i(\pi) = \sum_{e \in p_i} d_e(\omega_e(\pi)).$$

We can also say that each player i has a payoff function u_i and it is defined in terms of the cost function as usual as $u_i(\pi) = -c_i(\pi)$.

Using the definition coming from [1] a *network congestion game* Γ is a congestion game defined in a directed graph using the arcs as resources. Formally, it is defined by a tuple $\Gamma = (N, G, (s_i, t_i)_{i \in N}, (d_e)_{e \in E(G)})$ where $N = \{1, \ldots, n\}$ is the set of players, $G = (V, E)$ is a directed graph, $(s_i, t_i) \in V \times V$ is the pair of origin and destination nodes (or source and target nodes) for each player $i \in N$, and $d_e : \mathbb{N} \to \mathbb{R}$ is the delay function of every edge $e \in E$, which is assumed to be polynomial-time computable.

The strategy set of player i consists of simple paths in the directed graph G. In fact, Π_i is the set of all simple paths from s_i to t_i, denoted as all (s_i-t_i) paths, where the notation (s-t) path refers to a simple path between the nodes s and t. Since only simple paths are considered, the set formed by all the (s_i-t_i) paths is finite. In the case in which all the pairs (s_i, t_i) coincide with a unique pair (s, t), the game is said to be a *single-commodity network congestion game*, (otherwise it is called *multi-commodity*) and since all players share the same strategy-set the game is said to be symmetric.

It is useful to define a suitable notion of reduction among strategic games that preserves the existence of PNE and if this is the case, the complexity of finding a PNE.

Let $\mathcal{G}_1, \mathcal{G}_2$ be two classes of strategic games. We say that $\mathcal{G}_1$ is *Nash preserving reducible* or *reducible* to $\mathcal{G}_2$ (in polynomial-time) if there exist two (polynomial-time) computable functions f and g such that for any strategic game Γ, if $\Gamma \in \mathcal{G}_1$ then

i) $f(\Gamma) \in \mathcal{G}_2$,
ii) if π is and strategy profile of the game $f(\Gamma)$ then $g(\pi)$ is a strategy profile of Γ, and
iii) if π is a PNE of $f(\Gamma)$ then $g(\pi)$ also is a PNE of Γ.

The following result follows from the definition.

Theorem 1. *Let $\mathcal{G}_1$, $\mathcal{G}_2$ be two classes of strategic games. If any game in $\mathcal{G}_2$ has a pure Nash equilibrium and $\mathcal{G}_1$ is reducible to $\mathcal{G}_2$ then any game in $\mathcal{G}_1$ has a pure Nash equilibrium. If any game in $\mathcal{G}_2$ has a pure Nash equilibrium computable in polynomial time and $\mathcal{G}_1$ is reducible to $\mathcal{G}_2$ in polynomial time then any game in $\mathcal{G}_1$ has a pure Nash equilibrium computable in polynomial time.*

There is a rich literature on congestion games [11, 7, 6, 10, 1, 9, 5, 2, 8], here are some results concerning PNE that we use.

Theorem 2 (Rosenthal [11]). *Every congestion game has a* PNE.

Theorem 3 (Fabrikant, Papadimitriou, Talwar [1]). *There is a polynomial time algorithm to compute a* PNE *in symmetric network congestion games (single-commodity network congestion games).*

In what follows we consider that a *network* $\mathcal{N}$ is a tuple consisting of a weighted graph $G = (V, E)$ with non-negative weights a_e associated to each edge $e \in E(G)$ (the toll of traversing edge e) and non-negative weights b_v associated to each vertex $v \in V(G)$ (the value of the hidden item), this is, $\mathcal{N} = (G, (a_e)_{e \in E(G)}, (b_v)_{v \in V(G)})$. In the case that the graph is directed we use the term *directed network* and for undirected graphs the term *undirected network*.

3 Finders-share games

We start introducing the first family of strategic search games in which the benefit obtained from a node is splitted evenly among all the agents that have discovered the node.

A finders-share game is a tuple $\Gamma = (N, \mathcal{N}, (s_i)_{i \in N})$ representing the strategic game in which: N is a set of n players. $\mathcal{N} = (G, (a_e)_{e \in E(G)}, (b_v)_{v \in V(G)})$ is a network. For each player i there is a special vertex s_i of the graph which is its *starting point* (its source or origin). The strategies Π_i for player i are the set of simple paths in G starting from source s_i.

Given a configuration $\pi = (p_1, \ldots, p_n)$, the payoff or utility function u_i for player i is defined as follows.

$$u_i(\pi) = \sum_{v \in p_i} \frac{b_v}{l_v(\pi)} - \sum_{e \in p_i} a_e.$$

where $l_v(\pi) = |\{i | v \in p_i\}|$ is the number of players whose strategy contains vertex v.

Without lost of generality, all trough this article, we consider that the weight associated to each starting point is zero. This fact does not affect any of the results as we can consider the following transformation of the graph. We add an additional vertex per each source. The new source is connected only to the original source. Assigning weight zero to the new sources and to the connecting links we have a polynomial reduction to the variant in which the sources have always zero weight.

In the case in which all the s_i coincide with a unique vertex s the game is said to be a *single-source*, denoted as $\Gamma = (N, \mathcal{N}, s)$. Otherwise the game is *multi-source*.

In the case of strategic search games in which the source point for a player is a set of vertices instead of a single vertex the game is said to be *multi-start* and can be single or multi-source, depending on whether the starting set is common or not to all the players. Observe, that the most general class is formed by the multi-start multi-source games that include all the other classes.

Given an undirected network with associated graph G, we consider the directed network with associated graph G^d. G^d is obtained by transforming every edge $\{u, v\} \in V(G)$ with the same associated weight $a_{\{u,v\}}$ to the two arcs (u, v), (v, u) each with associated weight $a_{\{u,v\}}$. Observe that there is a one-to-one correspondence between the set of simple paths in G and the set of simple paths in G^d. Using this argument and taking into account that the node and edge weights do not change we obtain the following result.

Lemma 1. *The class of finders-share games for undirected networks is polynomial time reducible to the class of finders-share games for directed networks.*

Now we show the reduction from multi-start to multi-source finders-share games.

Lemma 2. *For directed networks, the class of multi-start multi-source finders-share games is polynomial time reducible to the class of multi-source games finders-share.*

Proof. Given $\Gamma = (N, \mathcal{N}, (S_i)_{i\in N})$ a multi-start multi-source finders-share game, we define the corresponding multi-source finders-share game $\Gamma' = f(\Gamma)$ as follows. Assume that $\mathcal{N} = (G(V, E), (b_v)_{v\in V}, (a_e)_{e\in E})$. Then $\Gamma' = (N, \mathcal{N}', (s_i)_{i\in N})$ where $\mathcal{N}' = (G(V', E'), (b'_v)_{v\in V}, (a'_e)_{e\in E})$ with:

- $V' = V \cup \{s_i | i \in N\}$, where s_i is a new vertex for player i. For each $v \in V, b'_v = b_v$ and $\forall i \in N, b'_{s_i} = 0$.
- $E' = E \cup \{(s_i, u) | i \in N \wedge u \in S_i\}$ where for each player i we add one edge from s_i to each different starting node $u \in S_i$. For each $e \in E, a'_e = a_e$ and $\forall i \in N, u \in S_i, a'_{(s_i,u)} = 0$.

Finally, $(s_i)_{i\in N}$ is the set of added vertices and s_i is the source of each player $i \in N$.

In order to distinguish the utility functions of both games, let us denote by u_i (u'_i) the utility function of player i in Γ (Γ').

Additionally, for any simple path p' of $G(V', E')$ starting at a source node of s_i, we define its corresponding simple path p of $G(V, E)$ as follows:

i) If $p' = s_i, v_0, \ldots, v_m$ then $p = v_0, \ldots, v_m$. Notice that s_i is a new node of Γ' and $p' = s_i, p$ where p is a simple path in $G(V, E)$ starting at $v_0 \in S_i$.
ii) If $p' = s_i$ then $p = v$ for some arbitrary node $v \in S_i$

We define a mapping $g : \Pi(\Gamma') \to \Pi(\Gamma)$ such that for every strategy profile $\pi' = (p'_1, \ldots, p'_n) \in \Pi$, $g(\pi') = \pi$ where $\pi = (p_1, \ldots, p_n)$. Note that $g(\pi'_{-i}, p'_i) = (\pi_{-i}, p_i)$. If we consider the load of each $v \in V - \bigcup_{1\leq i\leq n} S_i$ in both profiles π' and $\pi = g(\pi')$ we have that $l_v(\pi')$ in Γ' coincides with $l_v(\pi)$ in Γ. The load of the source nodes $v \in \bigcup_{1\leq i\leq n} S_i$ in Γ may be different from the load in Γ' but in both games the benefit $b_v = 0$ as well as $b_{s_i} = 0$ for each new s_i. Finally, note that for the new added edges $a_{(s_i,u)} = 0$. Hence, for each player i, $u'_i(\pi') = u_i(g(\pi')) = u_i(\pi)$.

Therefore, if $\pi' = (p'_1, \ldots, p'_n)$ is in $\textsc{pne}(\Gamma')$ then for every player i and every p' starting at s_i $u'_i(\pi') = u_i(\pi) \geq u'_i((\pi'_{-i}, p')) = u_i((\pi_{-i}, p))$ implying that $\pi = g(\pi')$ is in $\textsc{pne}(\Gamma)$.

Since f and g are polynomial-time computable, the result follows. □

Finally we reduce to the class of single-source finders-share games.

Lemma 3. *For directed networks, the class of multi-source finders-share games is polynomial time reducible to the class of single-source finders-share games.*

Proof. Given a multi-source finders-share game $\Gamma = (N, \mathcal{N}, (s_i)_{i \in N})$ we define the corresponding single-source finders-share game $f(\Gamma) = \Gamma' = (N, \mathcal{N}', s)$ as follows:

Assume that $\mathcal{N} = (G(V,E), (a_e)_{e \in E}, (b_v)_{v \in V})$ and that s_i is the starting vertex of k_i players. Let $b = \sum_{v \in V(G)} b_v$, $k = \max\{k_i | i \in N\}$ and $a = (k+1)b$. Then we define $\mathcal{N}' = (G(V', E'), (b'_v)_{v \in V}, (a'_e)_{e \in E})$ where $V' = V \cup \{s\}$ and $E' = E \cup \{(s, s_i) | i \in N\}$. The weights are defined as:

- $b'_s = 0$, for each player i, $b'_{s_i} = k_i a$, and for each v in $V \setminus \{(s_i)_{i \in N}\}$, $b'_v = b_v$.
- For each player i, $a'_{(s,s_i)} = a$ and, for each $e \in E$, $a'_e = a_e$.

Let us denote by u_i the utility function of player i in Γ and by u'_i the utility function of player i in Γ'. Notice that by the definition of Γ', each simple path p' in Γ' starts at s it continues visiting some of the original source nodes s_i of Γ. Hence $p' = s, p$ where p is a simple path of Γ. By definition of a and b, in any strategy profile π' of Γ', if a node s_i in Γ' is visited by more than k_i players then $u'_i(\pi') < 0$. Hence it can not be a $\textsc{pne}$ since $u'_i(\pi'_{-i}, s) = 0$.

We define a mapping $g : \Pi(\Gamma') \longrightarrow \Pi(\Gamma)$ such that for every $\pi' = (p'_1, \ldots, p'_n) \in \Pi$, $g(\pi') = \pi$ where $\pi = (p_1, \ldots, p_n)$ where

i) If $p'_i = s, s_i, p$ (p may be empty), then $p_i = s_i, p$, and
ii) If $p'_i = s, s_j, p$ (p may be empty) and $j \neq i$, then $p_i = s_i$.

Notice that $\forall i \in N$,

$$u_i(\pi) = \begin{cases} u'_i(\pi') & \text{if } p'_i = s, s_i, p, \\ 0 & \text{otherwise } (u'_i(\pi') < 0 \text{ and then } \pi' \text{ is not a } \textsc{pne}.) \end{cases}$$

Therefore, if π' is in $\textsc{pne}(\Gamma')$ we have that $u'_i(\pi') = u_i(\pi) \geq u'_i((\pi'_{-i}, p)) = u_i(g(\pi'_{-i}, p))$ for any strategy p of player $i \in N$ of Γ', implying that π is in $\textsc{pne}(\Gamma)$.

Since f and g are polynomial-time computable, the result follows. □

Next result shows the reduction to single-commodity network congestion games.

Lemma 4. *For directed networks, the class of single-source finders-share games is polynomial time reducible to the class of single-commodity network congestion games.*

Proof. Given a single-source finders-share game $\Gamma = (N, \mathcal{N}, s)$, we define the corresponding network congestion game $\Gamma' = f(\Gamma)$ as follows. Assume that $\mathcal{N} = (G(V, E), (a_e)_{e \in E}, (b_v)_{v \in V})$. $G' = (V', E')$ where:

- $V' = V \cup \{t\} \cup \{u' | u \in V \setminus \{s\}\}$.
- $E' = E \cup \{(u, u') | u \in V \setminus \{s\}\} \cup \{(u', t) | u' \in V' \setminus \{V \cup \{t\}\}\} \cup \{(u', v) | (u, v) \in E\}$.
- We define the non-decreasing delay function $d_e(x)$ as follows.

$$d_e(x) = \begin{cases} 0 & \text{if } e = (u, t),\ u \in V' \setminus V \\ a_{e'} & \text{if } e' = (u, v) \in E \text{ and } e = (u', v) \\ -\frac{b_u}{x} & \text{if } e \in \{(u, u') | u \in V, u' \in V' \setminus V\} \end{cases}$$

Finally, $\Gamma' = (N, G', (s, t), (d_e)_{e \in E(G)})$.

Additionally, for every strategy profile $\pi' = (p'_1, \ldots, p'_n)$ in $\Pi(\Gamma')$ such that $p'_i = s, v_0, v'_0, \ldots, v_k, v'_k, t$ is a simple path, we define $\pi = g(\pi')$ of $\Pi(\Gamma)$ as $\pi = (p_1, \ldots, p_n)$ with $p_i = s, v_0, \ldots, v_k$. Notice that $\forall i \in N$, p_i is a simple path and that $c_i(\pi') = u_i(\pi)$. Therefore, if π' is in PNE(Γ') we have that $c_i(\pi') = u_i(\pi) \geq c_i((\pi'_{-i}, p)) = u_i(g(\pi'_{-i}, p))$ for any strategy p of player $i \in N$ of Γ', implying that π is in PNE(Γ).

Since f and g are polynomial-time computable, the result follows. □

As a consequence of the previous results and Theorems 2 and 3 we can state the following.

Theorem 4. *Every multi-start multi-source finders-share game on a directed or undirected graph has a* PNE*. Furthermore, a* PNE *can be computed in polynomial time.*

4 Firsts-share games

Now we introduce the second family of strategic search games in which the benefit obtained from a node is splitted evenly only among all the agents that discover first the node. We assume uniformity on the time to traverse a link and measure time by number of traversed links.

A firsts-share game is a tuple $\Gamma = (N, \mathcal{N}, (s_i)_{i \in N})$ representing the strategic game in which strategies are the same as for the finders-search games but given a configuration $\pi = (p_1, \ldots, p_n)$, the utility function u_i for player i is defined as:

$$u_i(\pi) = \sum_{\substack{v \in p_i \\ d(v, p_i) = d_{min}(v, \pi)}} \frac{b_v}{l_v(\pi)} - \sum_{e \in p_i} a_e$$

where, $d(v, p_i)$ denotes the distance from the source to v in p_i (and it is defined as the length of the path from the source to v if v is in p_i and as ∞ otherwise), $d_{min}(v, \pi) = \min\{d(v, p_i) \mid p_i \in \pi\})$ is the minimum distance of v over every p_i

in the strategic profile π and, $l_v(\pi) = |\{i \in N | dist(v, p_i) = d_{min}(v, \pi)\}|$ is the number of players whose strategy contains vertex v with minimal distance to the source.

Let us observe that the difference between firsts-share games and finders-share games relies on the definition of $l_v(\pi)$. As we shall see in what follows, this difference in the splitting of discoveries has relevant implications on the existence of PNE as the games have very different properties.

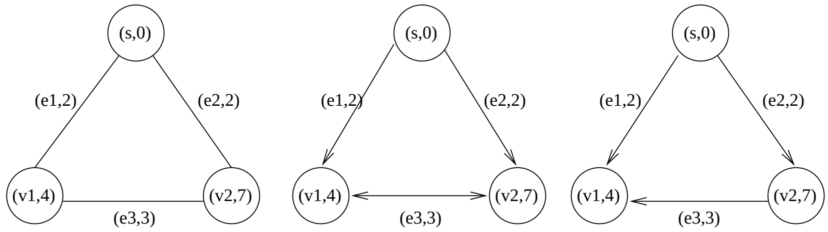

Fig. 1. Examples of firsts-share games for 2 players that do not have PNE.

Theorem 5. *In the class of firsts-share games there are games with* PNE *and games without* PNE.

Proof. The games with two players associated to the graphs in Fig. 1 do not have a PNE. The proof is by exhaustive inspection of all strategy profiles. Examples of firsts-share game with PNE can be obtained from the graphs in Figure 1 changing the weights of vertices v_1 and v_2 to 2, of edges e_1 and e_2 to 1 and of edge e_3 to 0. In all the cases the proof of existence or not of PNE is by inspection of all the possible strategy profiles for the two players. □

Using a construction inspired in the examples in Fig. 1 we can state conditions under which the family of search games that are played on a fixed graph does not always have a PNE.

Theorem 6. *Let G be a graph and $s \in V(G)$. If there is a vertex $v \in V(G)$ such that there are two paths of different length from s to v, then there is a weight assignment to G for which the corresponding search game has no* PNE.

Now we identify some subfamilies of games, defined by properties of the network, with PNE. According to the previous results we have to restrict our subfamilies to guarantee some equidistance properties for the sources. Observe that the reduction from the multi-source to the single-source version of the finders-share game given in Lemma 3 is not valid anymore as this reduction might generate paths of different lengths from the new source.

Observe that an undirected graph that contains a cycle accessible from a source verifies the conditions of Theorem 6. Therefore, for having always a PNE, independently of the weights, we must restrict to acyclic undirected graphs. In such a case the graph is a forest and therefore there is a unique simple path from every potential source to any other vertex of the same tree. In such a case firsts-share and finders-share benefits are the same and, according to Theorem 4, we have the following result.

Theorem 7. *Every single-source firsts-share game played in a forest has a* PNE *that can be computed in polynomial time.*

For the case of directed graphs we introduce three graph families: equidistant graphs, hierarchical-equidistant graphs and asymmetric tree coupling, and show the existence of PNE for the associated firsts-share game.

An *equidistant graph* is a directed network with a set of $k \geq 1$ sources $s_1, \ldots, s_k$ in which: (a) For any vertex u and any source s_i all the simple paths from s_i to u have the same length. (b) For any vertex u and any two sources s_i and s_j such that there is a path from s_i to u and from s_j to u, both paths have the same length.

Observing that in such a graph the utility function for every player is the same for firsts-share game than for finders-share game because the distances are equal we obtain the following result.

Theorem 8. *Every single and multi-source firsts-share game played in an equidistant graph has a* PNE *that can be computed in polynomial time.*

A *hierarchical-equidistant graph* is a directed network whose set of vertices V and of sources S can be partitioned into k subsets $V_1, \ldots, V_k$ and $S_1, \ldots, S_k$ respectively in such a way that: (a) The subgraph of G restricted to V_i and S_i, for every $1 \leq i \leq k$, is an equidistant graph. (b)For all i, j with $1 \leq i < j \leq k$ and every vertex $u \in V$, if there is a path from a source in $s_i \in S_i$ to u and a path from a source $s_j \in S_j$ to u then it follows that the path from s_i to u is shorter than the path from s_j to u.

We provide a polynomial time algorithm for computing a pure Nash equilibria. The algorithm uses self-reducibility and the polynomial time algorithm for equidistant graph. The recursion relies on the hierarchical structure of the sources.

Theorem 9. *Every single and multi-source firsts-share game played in a hierarchical-equidistant graph has a* PNE *that can be computed in polynomial time.*

Proof. Consider the following algorithm in which players from different sources play among them on a particular subgraph that is determined by the strategies of the previously considered players.

In round 1 the players whose source is in S_1 select their strategy according to a PNE π_1 in the graph $G[V_1]$. This Nash equilibrium is computed in polynomial time using the algorithm in Theorem 8. Since all the players whose source is not in S_1 arrive later to nodes in V_1 there is no conflict with the hidden items in these nodes and therefore players starting in S_1 won't have any incentive to change their strategy. Players starting from other sources cannot get any benefit from the discovered places. Therefore the selections of the players in S_1 remain fixed for forthcoming rounds. For doing so we modify the node weights of the nodes in the paths selected in π_1 to zero. The same procedure is repeated for rounds 2 to k. At round i the players in S_i compute a pure Nash equilibrium π_i on the graph modified according to the selected strategies $\pi_1, \ldots, \pi_{i-1}$.

Since for every round, the selection of strategies is performed in polynomial time and there are k such rounds, the PNE is computed in polynomial time. □

An *asymmetric tree coupling* is a directed network composed by two rooted trees on the same set of leaves, oriented from the root to the leaves, such that each leaf has a different distance from the two roots. We provide a polynomial time algorithm based on a *conquer and retreat* paradigm combined with a greedy algorithm for computing a PNE in a single-source firsts-share game played on a tree.

Theorem 10. *Every 2-source firsts-share game played in an asymmetric tree coupling has a PNE that can be computed in polynomial time.*

Proof. Our algorithm for computing a equilibrium is based on a conquer and retreat paradigm. Initially the players with source s_i $(i = 1, 2)$ play the search game on a subtree that contains only those leaves that are closer to their source. Along the algorithm players will be able to reconsider their position but allowing paths that use leaves that were not used by their opponent. Before describing the algorithm we need a piece that solves the problem of recomputing a PNE on a single-source tree with additional accessible leaves.

Assume that we have a tree T, and a subsets of leaves L. Assume also that we have a strategy profile π which is a PNE in the subtree in which the leaves in L' are removed. The following greedy rule computes a PNE for T.

> GreedyNash(T, L, π) Compute the path p_m in π with minimum benefit an the path p_M not in π with maximum benefit. If the benefit obtained in p_m is strictly smaller than that of p_M assign p_M to one of the player playing p_m. Repeat the process until no changes are made.

Observe that the algorithm finalizes in polynomial time as the number of considered paths is polynomial, the graph is a tree, besides the minimum and strictly increasing rule guarantees that an abandoned path will provide benefit below the minimum path benefit on the new profile and, therefore, will never be reconsidered again. At the end of the algorithm we have that all the non used paths have benefit at most the minimum over the selected paths, so the resulting strategy is a PNE.

Let $G = (V, E)$ be an asymmetric tree coupling formed by the two trees $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$. Let L_1 be the set of leaves whose distance to the root of T_1 is smaller than their distance to the root of T_2 and L_2 be the set of leaves in which this distance is greater.

Consider the following algorithm in which initially we compute separately PNEs for the two search games in which the two trees are separated and the players have access only to the shortest distance to their source leaves. The algorithm will refine this situation by allowing the conquest of the opponents unused leaves.

Set $A_1 = L_1$ and $A_2 = L_2$.
Compute π_1 a PNE for the game played on the tree T_1 removing the leaves in L_2.
Compute π_2 a PNE for the game played on the tree T_2 removing the leaves in L_1.
Let A'_1 be the set of leaves occupy by π_1
Let A'_2 be the set of leaves occupy by π_2
Set found $(A_1 = A'_1$ or $A_2 = A'_2)$.
while not found do
    $A_1 = A'_1$; $A_2 = A'_2$.
    $\pi_1 = \mathsf{GreedyNash}(T_1, A_2, \pi_1)$.
    Let A'_1 be the set of leaves occupy by π_1
    If $A_1 = A'_1$, found = true
    otherwise,
        $\pi_2 = \mathsf{GreedyNash}(T_2, A'_1, \pi_2)$.
        Let A'_2 be the set of leaves occupy by π_2
        If $A_2 = A'_2$, found= true
    endif
endwhile
return (π_1, π_2).

In the first steps the algorithm computes a PNE for the set of players with source s_i, in the graph formed by the subtree of T_i that results from subtracting the set of leaves closed to the other source. Observe that, if either π_1 or π_2 occupy the whole sets L_1 or L_2 respectively, then the strategic profile $\pi = (\pi_1, \pi_2)$, is a PNE for the game in which the whole network G is considered.

In the forthcoming rounds the algorithm starts with a set of leaves L'_i, for each player i, that has been occupy by the PNE computed in the previous step. In next round, we allow, first, players from s_1 to play in the tree with their closed leaves and the opponent unused leaves. Let π'_1 be the resulting PNE that doesn't occupy the set of leaves $E'_1 \subseteq L_1$. Then, either $E'_1 = L'_1$ and in this case $\pi = (\pi_1, \pi_2)$ is a PNE, or $E'_1 \supset L'_1$ since the unique way a player from s_1 can ameliorate his strategy is by means of a new path, one not considered in previous round, and therefore using at least an additional leaf closer to s_2. Observe that either we found a PNE or the subset of leaves used by players from source s_1 in L_2 has increased at least by one.

The process continues in alternative rounds until the set of occupy leaves doesn't change. The final strategic profiles of the two set of players will conform then a PNE for the game in the whole network G.

Since the size of the sets of conquered leaves from s_1 in L_2 and from s_2 in L_1 increases at each complete round, the maximum number of possible rounds is $O(|L_1| + |L_2|)$ and therefore a PNE can be computed in polynomial time. □

All along this section we have taken the number of edges as the measure of the length of a path. The results in this section also hold when each edge has associated a positive integer distance of polynomial length.

5 Finders-share games under other strategy definitions

We consider now the case in which the strategy for each player is selected from the set of all paths (instead of the set of all simple-paths) of the network starting at the designated origins. Recall that in a path the agent can pass more than once through a node but cannot use twice the same link (edge or arc). We have the following result.

Theorem 11. *Every finders-share game played in a directed or undirected search network where the set of trajectories consists of paths always has a* PNE.

Proof. We show that when the set of possible strategies Π consists of a set of paths of a directed or undirected network every finders-share game can be reduced to a congestion game. Thus, as a consequence of Theorem 2, we get the claimed result.

Consider a finders-share game $\Gamma = (N, \mathcal{N}, (S_i)_{i\in N})$ on an undirected network $\mathcal{N}$, where agent $i \in N$ is allowed to follow any path starting at some vertex in the set S_i. For any agent $i \in N$, set $\mathcal{P}(i)$ to be the set of allowed trajectories for i, that is all paths in $\mathcal{N}$ that start in a vertex in S_i. For any path p in $\mathcal{N}$ define $R(p)$ to be the set formed by all the nodes and edges that appear in p. We define the corresponding congestion game $\Gamma' = f(\Gamma) = (N, \mathcal{R}, (\Pi_i)_{i\in N}, (d_e)_{e\in\mathcal{R}})$ as follows. Assume that $\mathcal{N} = (G(V,E), (a_e)_{e\in E}, (b_v)_{v\in V})$, then $\mathcal{R} = V \cup E$. For any $i \in N$, set $\Pi_i = \{R(p) \mid p \in \mathcal{P}(i)\}$. For any $r \in \mathcal{R}$ we define the non-decreasing delay function $d_r(x)$ as follows.

$$d_r(x) = \begin{cases} a_r & \text{if } r \in E \\ -\frac{b_r}{x} & \text{if } e \in V \end{cases}$$

For every strategy for agent i in Γ' we associate, in a unique way, a valid path for agent i in $\mathcal{N}$. Observe that when the set of edges form a cycle there might be more that one path giving raise to this set. To break ties we will use the lexicographic order of edges going out of a node. In a cycle of an undirected graph we select the first edge in lexicographic order to start traversing the cycle. When the trajectory have more than one cycle, we traverse cycles in lexicographic order. In this way we define, for any strategy profile, $\pi' = (p'_1, \ldots, p'_n)$ in $\Pi(\Gamma')$ a strategy profile $\pi = (p_1, \ldots, p_n) = g(\pi')$ of $\Pi(\Gamma)$. Observe that g can be computed in polynomial time.

Notice that $\forall i \in N$, p_i is a path starting at some allowed vertex for agent i, and that $c_i(\pi') = u_i(\pi)$. Therefore, if π' is in PNE(Γ') we have that $c_i(\pi') = u_i(\pi) \geq c_i((\pi'_{-i}, p)) = u_i(g(\pi'_{-i}, p))$ for any strategy p of player $i \in N$ of Γ', implying that π is in PNE(Γ).

Since f and g are polynomial-time computable, the result follows.

For the case of a directed network the proof follows the same lines but we have to consider as resources in the congestion game the union of nodes and arcs. □

We can also consider the case in which the cost per edge corresponds to buying the right to traverse the edge as many times as wished. It is easy to show that, under such cost interpretation for the finders-share search game PNE happens only on strategies that correspond to a subtree rooted at the associated starting vertex of the graph. The proof of the following result is similar to the one for path strategies.

Theorem 12. *Every finders-share game played in a directed or undirected search network where the set of trajectories consists of trees always has a* PNE.

Proof. Consider a finders-share strategic search game $\Gamma = (N, \mathcal{N}, (S_i)_{i \in N})$ on an undirected network $\mathcal{N}$, where agent $i \in N$ is allowed to select any tree rooted at some vertex in the set S_i. For any agent $i \in N$, set $\mathcal{T}(i)$ to be the set of allowed trajectories for i, that is all trees in $\mathcal{N}$ rooted in a vertex in S_i. For any tree t in $\mathcal{N}$ define $R(t)$ to be the set formed by all the nodes and edges that appear in t. We define the corresponding congestion game $\Gamma' = f(\Gamma) = (N, \mathcal{R}, (\Pi_i)_{i \in N}, (d_e)_{e \in \mathcal{R}})$ as follows. Assume that $\mathcal{N} = (G(V, E), (a_e)_{e \in E}, (b_v)_{v \in V})$, then $\mathcal{R} = V \cup E$. For any $i \in N$, set $\Pi_i = \{R(p) \mid p \in \mathcal{P}(i)\}$. For any $r \in \mathcal{R}$ we define the non-decreasing delay function $d_r(x)$ as follows.

$$d_r(x) = \begin{cases} a_r & \text{if } r \in E \\ -\frac{b_r}{x} & \text{if } e \in V \end{cases}$$

Observe that for every strategy for agent i in Γ' we can associate, in a unique way, a valid tree for agent i in $\mathcal{N}$. In this way we define, for any strategy profile, $\pi' = (p'_1, \ldots, p'_n)$ in $\Pi(\Gamma')$ a strategy profile $\pi = (p_1, \ldots, p_n) = g(\pi')$ of $\Pi(\Gamma)$. Furthermore, $\forall i \in N$, p_i is a tree rooted at some vertex in S_i and that, by definition, we have that $c_i(\pi') = u_i(\pi)$. Therefore, if π' is in PNE(Γ') we have that $c_i(\pi') = u_i(\pi) \geq c_i((\pi'_{-i}, p)) = u_i(g(\pi'_{-i}, p))$ for any strategy p of player $i \in N$ of Γ', implying that π is in PNE(Γ).

Since f and g are polynomial-time computable, the result follows.

For the case of a directed network the proof is the same, considering as resources the union of nodes and arcs. □

The previous results guarantee only the existence of PNE but it remains open whether a polynomial time algorithm for computing one PNE exists in those particular cases.

6 Conclusions and open problems

We have defined a new class of strategic games, those games have been motivated by the study of resource discovery in distributed networks. We believe that this framework is general enough to incorporate other mechanism to splitting benefits and costs in other settings. We have also introduced the notion of Nash preserving reduction that could be used to derive further results in the study of other strategic games. Our results show a close connection with network congestion games for the finders-share model while for the firsts-share the games behave differently from the point of view of the existence of PNE.

There are still many open problems concerning the firsts-share model, among others, characterize the networks with PNE. Observe that in some cases this might be difficult as the existence of PNE depends on the edge and node weights. It will be of interest to determine whether the existence of PNE can be solved in polynomial time for non-equidistant networks. In the asymmetric tree coupling nodes are dominated by exactly one of the two sources, we do not know whether the existence of PNE can be established for a tree coupling in which a subset of the leaves are at the same distance from the two sources.

For the finders-share cost model we have shown the existence of PNE equilibria and that a PNE can be obtained in polynomial time, independently of the number of sources. It will be of interest to analyze further properties on the structure of the PNE in regard to some topological graph property.

References

1. A. Fabrikant, C. Papadimitriou, and K. Talwar. The complexity of pure nash equilibria. In *Proceedings of the thirty-sixth annual ACM Symposium on Theory of Computing (STOC)*, pages 604–612, 2004.
2. D. Fotakis, S. Kontogiannis, and P. Spirakis. Symmetry in network congestion games: Pure equilibria and anarchy cost. In *Lecture Notes in Computer Science*, volume 3879, pages 161–175. Springer Berlin / Heidelberg, 2006.
3. S. Gal. *Search Games.* Academic Press, 1980.
4. R. Isaacs. *Differential Games.* John Wiley and Sons, 1965.
5. S. Kontogiannis and P. Spirakis. Atomic selfish routing in networks: A survey. In *Proceedings of the First International Workshop on the Internet and Network Economics (WINE 2005)*, pages 989–1002. Springer, 2005.
6. E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *STACS 99: Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science*, volume Volume 1563, pages 404–413. Springer Berlin / Heidelberg, 1999.
7. I. Milchtaich. Congestion games with player-specific payoff functions. *Games and Economic Behavior*, 13(1):111–124, 1996.
8. N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors. *Algorithmic Game Theory.* Cambridge University Press New York, NY, USA, 2007.
9. P. Panagopoulou and P. Spirakis. Efficient convergence to pure nash equilibria in weighted network congestion games. *Lecture Notes in Computer Science*, 3503:203–215, 2005.
10. C. Papadimitriou. Algorithms, games, and the internet. In *STOC '01: Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 749–753, New York, NY, USA, 2001. ACM Press.
11. R. W. Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, (2):65–67, 1973.