

# A formal model based on Game Theory for the analysis of cooperation in distributed service discovery

Guillem Martínez-Cánovas, Elena Del Val, Vicente Botti, Penélope Hernández, Miguel Rebollo

*ERI-CES, Departamento de Análisis Económico, Universitat de València, Spain*  
*{guillem.martinez, penelope.hernandez}@uv.es*

*Departament de Sistemes Informàtics i Computació, Universitat Politècnica de València, Spain*

*{edelval, vbotti, mrebollo}@upv.es Tel. +34 963877350 Fax. +34 963877359*

---

## Abstract

New systems can be designed, developed, and managed as societies of agents that interact with each other by offering and providing services. These systems can be viewed as complex networks where nodes are bounded rational agents. In order to deal with complex goals, agents must cooperate with other agents to be able to locate the required services. The aim of this paper is to formally and empirically analyze under what circumstances cooperation emerges in decentralized search for services. We propose a repeated game model that formalizes the interactions among agents in a search process where each agent has the freedom to choose whether or not to cooperate with other agents. Agents make decisions based on the cost of their actions and the expected reward if they participate by forwarding queries in a search process that ends successfully. We propose a strategy that is based on random-walks, and we study under what conditions the strategy is a Nash Equilibrium. We performed several experiments in order to evaluate the model and the strategy and to analyze which network structures are the most appropriate for promoting cooperation.

*Keywords:* Networks, distributed service discovery, Nash Equilibrium, Repeated games

---

## 1. Introduction

Social computing has emerged as a discipline in different fields such as Economics, Psychology, and Computer Science. Computation can be viewed as a social activity rather than as an individual one. New systems are designed, developed, and managed as virtual societies of independent entities or agents that offer services and interact with each other by providing and consuming these services [23]. These systems and applications can be formally represented through formal models from the field of Complex Networks [27]. This area provides a theoretical basis for the development of models that help us to reason about how distributed systems are organized [19]. Complex Network models have been used in different contexts such as social networks (collaboration, music, religious networks), economic networks (trade, tourism, employment networks), Internet (structure and traffic networks), bio-molecular networks, and computer science networks among others [5, 26].

In the complex systems already mentioned, one of the challenges is the design of efficient search strategies to be able to locate the resources or services required by entities in order to deal with complex goals [2, 26, 9]. Taking into account the autonomy of the entities that participate in the search process, three levels of search decentralization could be considered. At the *first level*, we assume that the search process is centralized when there is a common protocol that is adopted by all the entities of the system and that this protocol dictates the actions that must be followed (i.e., the protocol specifies the entity that starts the process, the sequence of participation of entities, and the target). At the *second level*, this protocol can be relaxed. The entities adopt that protocol and carry out the same set of actions, but the search path (i.e., the sequence of entities that participate in the search process) is not specified. At the *third level*, a decentralized search can be considered when there is a protocol adopted by all the entities that specifies the set of available actions. However, these entities can decide whether or not they are going to follow the protocol. It would not be desirable to impose the same behavior on all of the nodes if it takes away their individual choice, (i.e., it would be desirable for all of the nodes to follow the protocol willingly). Therefore, we look for a concept of stability within the strategies of the entities of the system. This concept, which comes from Game Theory, is known as the Nash Equilibrium.

There are several scenarios where efficient decentralized search strategies

are required. Some of these scenarios are wireless ad-hoc networks where nodes rely on other nodes to forward their packets in order to reach the destination node (e.g., such as file sharing in P2P systems, streaming applications, discussion boards, on-line auctions, or overlay routing). In this paper, we consider a P2P system as an example of an application scenario. The P2P system is modeled as a multi-agent system. In this scenario, agents act on behalf of users by playing the role of a service provider or service consumer (see Figure 1). Agents that play the role of service consumers should be able to locate services, make contract agreements, and receive and present results [34]. Agents that play the role of service providers should be able to manage access to services and ensure that contracts are fulfilled. By considering the system as a network, it is assumed that all of the information is distributed among the agents. Since agents only have a local view of the services provided in the network, the collaboration of other agents is required in order to reach the target. For instance, in the scenario presented in Figure 1, if the agent that acts on behalf of *ClientA* (agent *ClientA*) requires a service to see a film without paying, it should interact with the provider agent of the service *P2PFilms*. However, agent *ClientA* has local knowledge about the available services and their providers (i.e., it only knows about *Netflix* and *rentalCar* services). Therefore, agent *ClientA* starts a search process for a provider agent that performs the task of *P2PFilms* service. To locate the provider, agent *ClientA* needs the collaboration of the rest of the agents in the network. We assume that there is at least one provider agent that is able to perform the task.

During a search process, agents can carry out a set of actions: create a task that must be performed by a qualified agent (i.e., start a search process), forward the task to one or several neighbors if they do not know how to solve the task, or perform the task if they can provide the required service. The cooperation of agents forwarding queries plays a critical role in the success of the search process [8]. This action facilitates the location of a resource based on local knowledge. However, in our scenario, this action has an associated cost and agents are free to decide whether or not the forwarding action is profitable for them based on its cost and the expected reward.

In this paper, we propose a model to formally describe the distributed search for services in a network as a game. Specifically, we use the repeated game framework to model both the process that a task follows through the network and the global task-solving process. In the former, each period is a decision stage for the agent who is in possession of the task. In the latter,

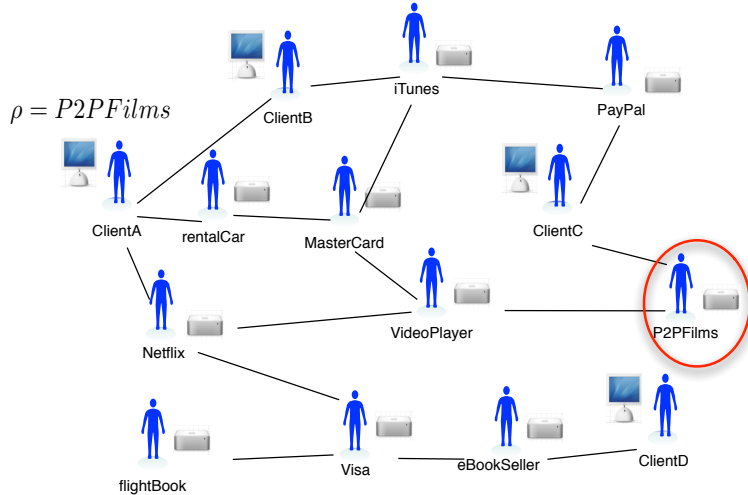


Figure 1: An example of a search scenario.

a project is generated in each period and randomly assigned to an agent in the network.

Our intention is to analyze the relationship between the cost of forwarding the task and the reward that agents obtain later when the task is solved in order to guarantee that cooperation is a stable behavior in the game. We refer to this reward as  $\alpha$ . The cost of the search process and the reward required to promote cooperation among agents depends on the network structure and the paths that the tasks follow (i.e., the search paths). We establish a bound for the length of the search process using Mean First Passage Time (MFPT), which is the average number of steps necessary to go from an agent  $i$  to another agent  $j$  in the network. Therefore, the structure of the network also characterizes  $\alpha$  through the MFPT, and, consequently, the network structure influences the agents' behavior. In order to verify this, we ran simulations to contrast the possible differences among network structures. The results show that for a set of networks with the same structural properties (i.e., number of nodes, edges, and average degree of connection), the network topology that offers the best results for the proposed search model is the scale-free topology. This is because the diameter of scale-free topology is closer to the limit of steps in the search process than other network structures.

The paper is organized as follows. Section 2 presents other works related

to emergence of cooperation in distributed environments. Section 3 introduces a repeated game model to formalize the search process of services in agent networks. In Section 4, some strategies that agents can follow in the repeated game are analyzed in order to determine whether or not they are Nash Equilibrium profiles. Section 5 describes several experiments that we performed to empirically evaluate the theoretical results in different network structures. In these experiments we also analyze the influence of the network structure and to determine which structure facilitates the emergence of cooperation in the proposed repeated game. Finally, Section 6 concludes the paper.

## 2. Related Work

Nowadays, centralized approaches cannot deal with current large-scale, highly dynamic systems such as streaming media systems [33], systems for sharing and distributing files [36], or telephony [15]. Weaknesses such as bottlenecks, lack of robustness, and the need for huge amounts of memory to store information about available resources make centralized approaches unsuitable for coping with dynamic system requirements. Distributed approaches provide a set of features that make them appropriate for dealing with resilience to faults and attacks, decentralized content management, or self-organization.

Distributed systems are populated by entities that deal with complex tasks and require services or resources that are provided by other entities in order to fulfill their goals. Therefore, search strategies are required to be able to locate them. There is a set of search algorithms that are based on Distributed Hash Tables (DHT) [11]. The main advantage of these algorithms is that the search process is bounded to  $O(\log(n))$  where  $n$  is the number of nodes in the network. Nevertheless, the maintenance of the indexes when nodes join and leave the system affects the performance of the system. These updates imply the interchange of messages among nodes; therefore, the system could be in an inconsistent state during a period of time due to outdated references among nodes.

Other works in the literature consider a decentralized search where each node only has local information, all the nodes are considered equal, and there is an arbitrary topology. These structures provide more flexibility and adaptability. However, nodes need the collaboration of the others in order to succeed in the search process. The search approaches in decentralized

systems use informed or blind search strategies for locating services or resources. *Informed* strategies exploit the information located in each node to improve the navigation of the network [7]. These strategies are domain dependent. *Blind* search strategies do not consider information stored in each node. The main advantage is that they can be applied in several domains because they do not require specific knowledge. An example of a blind search strategy is the use of flooding mechanisms [22]. The main drawback of flooding mechanisms is the overall communication traffic overhead.

*Random-walk* search strategies have been presented as an alternative to *flooding* strategies [4, 37, 22] since they reduce the traffic in the system and might provide better results [24, 39]. A random-walk search algorithm randomly selects a neighbor each time to forward the message to [14]. There are many search proposals that navigate networks using random-walks strategies since they do not require specific knowledge and can be applied in several domains. Some of these works have introduced modifications such as using random-walk strategies from multiples sources [40, 31] or adding information about routes [20, 1, 3] in an attempt to improve the search efficiency. Structural heterogeneity affects the nature of the diffusive and relaxation dynamics of the random walk strategy [28], and the biased random-walk strategy based on preferential transition probability [12].

One of the common assumptions in network search is that all of the agents have homogeneous behavior and that all of them are going to cooperate by forwarding messages. However, this does not correspond with real scenarios. In real, large-scale networks, decisions are often made by each agent independently, based on that agent's preferences or objectives. Game Theory models are well suited to explain these scenarios [25]. Game Theory studies the interaction of autonomous agents that make their own decisions while trying to optimize their goals. Game Theory provides a suite of tools for modeling interactions among agents with different behaviors [35].

There are works in the area of Game Theory that focus on the routing problem in networks where agents are characterized by selfish behavior [6]. Specifically, this problem has been studied in wireless and ad-hoc networks [35, 25, 10]. To deal with selfish agents, numerous approaches use reputation [18] (i.e., techniques based on monitoring the nodes' behavior from a cooperation perspective) or price-based techniques [17] (i.e., a node receives a payment for its cooperation in forwarding network messages and also pays other nodes which participate in forwarding its messages). One of the drawbacks of reputation systems is that nodes whose reputation values are higher

than a threshold are treated equally. Therefore, a node can maintain its reputation value just above the threshold in order to obtain the same benefit as nodes with higher reputation levels. One of the problems of price-based techniques is that they are not fair with nodes that are located in regions with low traffic that have few opportunities to earn credit. Li et al. [21] integrate both techniques and propose a game theory model to analyze the integrated system. However, this approach does not consider the influence of the underlying structure in the emergence cooperation.

To understand the social behavior of systems, it is important to consider the network structure. There are several works that analyze the influence of the network structure when the agents of the networks do not follow homogeneous behavior. These works study how structural parameters such as clustering or degree distribution affect the emergence and maintenance of cooperative behavior among agents [32, 30]. Hofmann et al. [16] present a critical study about the evolution of cooperation in agent societies. The authors conclude that there is a dependence of cooperation on parameters such as network topology, game interaction, update rules of states, and initial fraction of cooperators.

The proposal presented in this paper analyzes the problem of cooperation emergence ‘through a Game Theory model’ in the context of decentralized search. Our approach differs from other approaches in several ways. First, we considered a game that fits the characteristics of decentralized search better than other games proposed in the literature (which are based on the often studied Prisoner’s Dilemma). Second, agents’ decisions about cooperation are based on a utility function that takes into account the properties of network topology. Moreover, the utility function also considers a limit to the number of possible steps to reach the target agent. This feature is important in distributed systems in order to avoid traffic overhead. Third, the strategy that agents follow is based on a search mechanism that is often used in network navigation and does not require specific domain knowledge. Therefore, the model can be easily applied in different search contexts. Finally, to promote cooperation, instead of using reputation or price-based mechanisms, we used a mechanism that is based on incentives provided by the system. We formally and experimentally determine the minimum required reward that satisfies the stability condition of Nash Equilibrium.

### 3. The Model

Consider a finite set of agents  $N = \{1, 2, \dots, n\}$  that are connected by undirected links in a fixed network  $g$ . We use the adjacency matrix to represent the network, which is denoted by  $M$ . A link between agents  $i$  and  $j$  is represented by  $M_{ij} = 1$ , and by  $M_{ij} = 0$  if there is no link. The set of neighbors of agent  $i$  is  $N_i = \{j | M_{ij} = 1\}$ . For simplicity, we assume that  $M_{ii} = 0$  so all neighbors in  $N_i$  are different from  $i$ . Therefore, the cardinality of the set  $N_i$  is the number of neighbors that agent  $i$  has (its degree of connection), which is denoted by  $k_i = |N_i|$ .

We consider that each agent has a type that represents its ability or the service that it can provide, which we denote by  $\theta_i$ , where  $\theta_i \in [0, 1]$ . Agents are aware of the realization of their own type, but not of the other agents' types. Let  $\rho \in [0, 1]$  be a task or project to be solved by one of the agents in the network. An agent  $i$  can solve  $\rho$  if, for a fixed  $\varepsilon$ ,  $|\theta_i - \rho| \leq \varepsilon$  (i.e., its type  $\theta_i$  is 'similar' to the task to be solved). We assume that for each  $\rho$  there exists a set of agents denoted by  $K_\rho^*$  such that  $\forall k^* \in K_\rho^* : |\theta_{k^*} - \rho| < \varepsilon$  and  $|K_\rho^*| \geq 1$ .

We define an  $n$ -person repeated game  $\Gamma_\rho^\infty$  that takes place in the network  $g$ . Each agent has a set of actions  $A_i = \{\emptyset, 1, 2, \dots, k_i, \infty\}$ , where  $\emptyset$  means the agent does nothing;  $\infty$  means the agent solves  $\rho$ , and  $j$  in  $\{1, 2, \dots, k_i\}$  means the agent forwards the task to  $j$ , one of the agent's  $k_i$  neighbors.

In the first stage of the game ( $t = 1$ ), a task  $\rho$  is uniformly and randomly generated and assigned to a uniformly and randomly selected agent denoted by  $s$ . If  $s \in K_\rho^*$  (i.e.  $s$  is capable of solving the task), chooses action  $\infty$ , the game ends and  $s$  gets a payoff of  $1 - |\rho - \theta_s|$ . Note that the closer the agent's type  $\theta_s$  and the task  $\rho$  are, the higher the utility obtained. If  $s$  does not solve  $\rho$ , either  $s$  chooses to do nothing (action  $\emptyset$ ), (ending the process and getting a payoff of 0), or  $s$  forwards the task to one of its neighbors (action in  $\{1, 2, \dots, k_i\}$ ) with a cost  $c$ , where  $0 < c < 1$ . Meanwhile, all of the other agents (who do not have the task) choose action  $\emptyset$  (getting a payoff of 0 in that stage).

If  $\rho$  is forwarded to an agent  $i$ , the same process faced by  $s$  is now confronted by  $i$ . However, if  $i$  solves the task, all of the agents who previously forwarded  $\rho$  get a payoff  $\alpha$ , where  $c \leq \alpha$  (see Figure 2). The parameter  $\alpha$  can be considered to be as a compensation payoff for collaborating in the solving process, which is greater than the cost of forwarding but lower than the payoff for solving the task. If  $i$  chooses to do nothing, the solving process



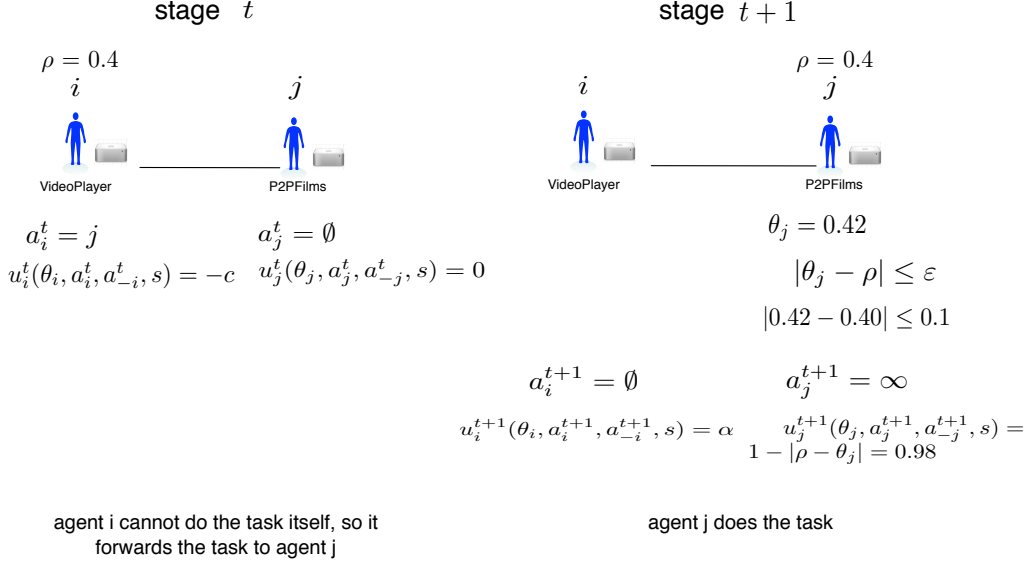


Figure 2: An example of a search process in which agent  $j$  solves  $\rho$ .

ends unsuccessfully and every agent gets a payoff of 0 in the last stage. Note that there may be some agents who paid the cost of forwarding and they may end up getting no compensation at all.

The utility function of an agent  $i$  at stage  $t$  is formally defined as:

$$u_i^t(\theta_i, a_i^t, a_{-i}^t, s) = \begin{cases} 1 - |\rho - \theta_i| & \text{if } a_i^t = \infty \\ -c & \text{if } a_i^t \in \{1, \dots, k_i\} \\ 0 & \text{if } a_i^t = \emptyset \wedge \nexists t' < t : a_i^{t'} \in \{1, \dots, k_i\} \\ \alpha & \text{if } a_i^t = \emptyset \wedge \exists t' < t : a_i^{t'} \in \{1, \dots, k_i\} \wedge \exists j \in N : a_j^t = \infty \end{cases}$$

where  $t$  is the stage,  $\theta_i$  is the type of agent  $i$ ,  $a_i^t$  is the action agent  $i$  chooses at stage  $t$ ,  $a_{-i}^t$  are the actions at state  $t$  of all other agents (i.e.  $a_j^t : \forall j \in N - \{i\}$ ), and  $s$  is the node where  $\rho$  starts at  $t = 1$ . Take into account that agents can choose among all possible actions only if they have the task, otherwise they can only play  $\emptyset$ .

In the next section, we compare different strategies that may be Nash Equilibrium of the game  $\Gamma_\rho^\infty$ . Note that in the game  $\Gamma_\rho^\infty$ , the agent  $s$  in

which  $\rho$  starts does not affect the analysis<sup>1</sup>; therefore, the parameter  $s$  can be omitted in  $u_i^t$  and  $u_i$  when no ambiguity occurs.

At stage  $t$ , agents are informed of actions that are chosen in previous stages of the game. Therefore, we say that the game  $\Gamma_\rho^\infty$  is under perfect information. Formally, let  $H^t$  be the cartesian product of  $A$  ( $t-1$ )-times, (i.e.,  $H^t = A^{t-1}$ ), with the common set-theoretic identification  $A^0 = \emptyset$ , and let  $H = \cup_{t \geq 0} H^t$ . The history  $H^t$  shows the path that  $\rho$  has taken through the network until stage  $t$ . A pure strategy  $\sigma_i$  for agent  $i$  is a mapping from  $H$  to  $A_i$ ,  $\sigma_i : H \rightarrow A_i$ . Obviously,  $H$  is a disjoint union of  $H^t$ ,  $t = 1, \dots, T$  and  $\sigma_i^t : H^t \rightarrow A_i$  as the restriction of  $\sigma_i$  to  $H^t$ . In this particular game, note that an agent only cares about whether or not  $\rho$  reaches it at stage  $t$  (i.e.,  $H^{t-1}$ ). Therefore,  $\sigma_i$  is a Markovian strategy since it only takes into account the previous stage. Actually, any possible history in  $H^t$  such that the project ends up in agent  $i$  has the same effect on decision of agent  $i$ . Therefore, those histories in  $H^t$  belong to an equivalence class.

The utility function for agent  $i$  in the game  $\Gamma_\rho^\infty$  is formally defined as:

$$u_i(\theta_i, \sigma_i, \sigma_{-i}, s) = \sum_{t > 0} u_i^t(\theta_i, a_i^t, a_{-i}^t, s) \quad (1)$$

where  $\sigma_i$  is the strategy played by agent  $i$  and  $\sigma_{-i}$  is the strategy profile composed by the strategies played by all agents except by agent  $i$ .

An action profile  $(\sigma_1^*, \dots, \sigma_n^*)$  is a Nash equilibrium in the network game  $\Gamma_\rho^\infty$ , if and only if no player has an incentive to change its strategy assuming other players follow their strategies. In other words, for any player  $i$ , there is no strategy  $\hat{\sigma}_i \neq \sigma_i^*$  that gives agent  $i$  a higher utility than  $\sigma_i^*$ . Formally,

$$u_i(\theta_i, \sigma_1^*, \dots, \sigma_n^*) \geq u_i(\theta_i, \sigma_1^*, \dots, \hat{\sigma}_i, \dots, \sigma_n^*), \quad \forall \sigma_i^* \neq \hat{\sigma}_i, \quad i \in N, \quad \text{and } \theta_i \in [0, 1].$$

The above utility criterion depends on where the process starts. Now, consider an extension of the previous model that satisfies the condition of independence on where the project starts. Let us define a new repeated game

---

<sup>1</sup>If the task does not reach agent  $i$ ,  $s$  does not affect its payoff for obvious reasons. If the task happens to reach agent  $i$ , two things can happen: agent  $i$  solves it because  $|\rho - \theta_i| \leq \epsilon$ , independently of  $s$ , or the task is forwarded. Since all forwarding strategies that agent  $i$  can follow share the same previous path until agent  $i$  is reached,  $s$  does not affect this comparison

denoted as  $(\Gamma_\rho^\infty)^N$  which consists of  $N$  repetitions of the game  $\Gamma_\rho^\infty$  starting in a different node at each repetition. The utility function in the game  $(\Gamma_\rho^\infty)^N$  for an agent  $i$  is formally defined as:

$$v_i(\theta_i, \sigma_i, \sigma_{-i}) = \frac{1}{N} \sum_{s=1}^N u_i(\theta_i, \sigma_i, \sigma_{-i}, s) \quad (2)$$

The set of equilibrium payoffs is the set of any payoff vector such that there exists an equilibrium strategy profile with the associated payoff vector. In order to characterize this, we make use of a well-known result called the Folk Theorem. The Folk Theorem states that the set of feasible and individual rational payoff vectors characterizes the set of equilibrium payoffs of the repeated game [13]. To define the set of feasible and individual rational payoffs, we first need to define first the min-max utility in pure actions. This level is sometimes called the reservation payoff, which is reached when players play their min-max action. The min-max action for agent  $i$  is the one that guarantees the highest possible payoff given the action profile  $a_{-i}$  that is the worst case scenario for agent  $i$ . Formally,

$$\bar{u}_i = \min_{a_{-i}} \max_{a_i} u_i(\theta_i, a_i, a_{-i}), a_i \in A_i, a_{-i} \in A_{-i} \quad (3)$$

In our set-up for the one-shot payoff function, the min-max strategy is  $\emptyset$ ; therefore, the min-max level is 0.

We now define the set of feasible payoff vectors  $F$  as any payoff profile in the convex hull of the set of possible payoff profiles of the game. The convex hull of a set  $X$  is the smallest convex set that contains  $X$  and is denoted by  $\text{conv}\{X\}$ . Formally,

$$F := \text{conv}\{x \mid \exists a \in A \text{ with } u(a) = x\} \quad (4)$$

The set of *strictly and individually rational payoff vectors* (relative to the min-max value in pure strategies) is composed by the payoff vectors that guarantee at least  $\bar{u}_i$  for all agents. Formally:

$$V := \{x = (x_1, \dots, x_n) \in F : x_i \geq \bar{u}_i \quad \forall i \in N\} \quad (5)$$

The Folk Theorem states that any payoff profile in  $V$  can be implemented as a Nash equilibrium payoff if the discount factor  $\delta$  applied to the utility of agents in each period is large enough. In the game  $\Gamma_\rho^\infty$ , we assume that all

agents are aware that the task will be solved soon enough to compensate the effect of  $\delta$ ; therefore, we consider the largest possible value for it which is 1.

The intuition behind the Folk Theorem is that any combination of payoffs where each agent gets at least its min-max payoff is sustainable in a repeated game, provided that each agent believes that there is a high probability that the game will be repeated. For instance, the punishment imposed on an agent who deviates is that the agent will be held to its min-max payoff for all subsequent rounds of the game. Therefore, the short-term gain obtained by deviating is offset by the loss in payoff in future rounds. Of course, there may be other less radical (less grim) strategies that also lead to the feasibility of some of those payoffs.

The good news from the Folk Theorem is that a wide range of payoffs may be sustainable in equilibrium, which it allows us to state that there exist equilibrium strategy profiles of the game  $\Gamma_\rho^\infty$ . The bad news is that there may be a multiple number of equilibria. In the next section, we examine whether or not some well-recognised strategies are indeed equilibrium strategies in  $\Gamma_\rho^\infty$ .

#### 4. Equilibrium strategies

In this section, we study pure Nash Equilibrium strategies of the game  $\Gamma_\rho^\infty$ . Namely, we start defining the *Nobody works* strategy, which basically consists of doing nothing, even in the case that an agent can perform the task. We prove that the strategy profile where all agents play the *Nobody works* strategy is not a Nash Equilibrium of the game. Then we consider the so-called random-walk strategy. In this strategy, when an agent is not able to solve the project, it uniformly and randomly chooses one of its neighbors to forward the task to. We establish the conditions under which the strategy profile in which every agent plays the random-walk strategy is a Nash Equilibrium of the game. We enrich the model by adding a threshold for the number of times that a task can be forwarded, and we also study under what conditions a random walk strategy profile is Nash Equilibrium in the game after adding this new condition.

##### 4.1. Nobody works

One possible strategy is the one we call *Nobody works*, in which every agent always chooses the action  $\emptyset$  and consequently gets a payoff of 0. One of the assumptions in our model is that, for all tasks  $\rho$ , there exists at least one agent  $k^*$  that is able to perform it. From our payoff criterion, we can

state that, in some period  $t$ , the project will start at agent  $k^*$ . In that case, if agent  $k^*$  chooses the  $\infty$  action (doing the project), agent  $k^*$  gets a payoff of  $1 - |\rho - \theta_{k^*}| > 0$ ; therefore, the *Nobody works* strategy is not an equilibrium strategy for agent  $k^*$ ; therefore, the action profile where all agents choose such a strategy is not a Nash Equilibrium of the game  $\Gamma_\rho^\infty$ .

#### 4.2. Random Walk

In this subsection, we study the case where all agents play a behavioral strategy denoted by  $\sigma_i^\infty$ , which is a mapping from the history they observe until the present period to a probability distribution over their action set, which leads to the well-known dynamics of “random-walk”. Formally:

$$\sigma_i^\infty : H^{t-1} \rightarrow \Delta(A_i) \quad (6)$$

We call this behavioral strategy the random-walk strategy. This strategy consists in doing nothing if no task arrives to an agent (action  $\emptyset$ ), solving the task if the agent is capable of doing so (action  $\infty$ ), and executing the forwarding action when the task arrives and the agent cannot solve it, choosing uniformly and randomly one of its neighbors to forward the task to.

This strategy is a “myopic” strategy since agents do not update their expected payoffs. Each agent  $i$  uniformly and randomly chooses one of its neighbors to continue searching for an agent  $k^*$  that can solve the task  $\rho$ . Recall that in our game, for all tasks  $\rho$ , there exists at least one agent who can solve it (i.e.,  $|K^*| \geq 1$  where  $\forall k^* \in K^* : |\theta_{k^*} - \rho| < \varepsilon$ ). As a consequence of the random-walk strategy, we can assert the existence of a finite time  $0 \leq \tilde{t} < \infty$  such that  $a_{k^*}^{\tilde{t}} = \infty$  [29]. Therefore, given a task  $\rho$ , the achieved payoff for each agent first depends on whether or not the agent was part of its solving path. If an agent  $i$  did not take part in the process, the agent gets a payoff of 0, which is the min – max payoff.

Now suppose that  $i$  is part of the solving path. Let us define some parameters that take part in the utility function. We refer to the probability of an agent being capable of doing the task as  $\gamma_i$ . Since agents’ types and  $\rho$  are drawn from a uniform probability distribution, independently,  $\gamma_i$  is the same for all agents and we simply call  $\gamma$ . The previously defined parameters  $\alpha$  and  $c$  are the reward and the cost of forwarding the task, respectively, and  $P_j^\infty$  is the probability that the task eventually reaches agent  $j \in \{1, \dots, n\}$ . To abbreviate purposes, let us denote the probability  $P_{\{k^* \in K^*\}}^\infty$  of reaching any  $k^* \in K^*$  simply as  $P_{k^*}^\infty$ .

Hence, the utility function of the game  $\Gamma_\rho^\infty$  for agent  $i$  is:

$$u_i(\theta_i, \sigma_i, \sigma_{-i}) = P_i^\infty (\gamma(1 - |\rho - \theta_i|) + (1 - \gamma)(P_{k^*}^\infty(\alpha - c) + (1 - P_{k^*}^\infty)(-c))) \quad (7)$$

The following proposition states that the strategy profile in which every agent plays a random-walk strategy is a Nash equilibrium in the game  $\Gamma_\rho^\infty$ .

**Proposition 1.** *There exists  $\alpha^* > 0$  such that the strategy profile  $(\sigma_1^\infty, \dots, \sigma_n^\infty)$  is a Nash Equilibrium in  $\Gamma_\rho^\infty$  starting at any node  $s \in \{1, \dots, N\}$ .*

*Proof.* Let  $\tilde{N} = \min_{i \in N} N_i$ .

We can suppose w.l.o.g that the project starts at  $s \in \{1, \dots, N\}$  and agent  $i$  is active in the search process. Note that if agent  $i$  can solve the project (i.e.,  $|\theta_i - \rho| < \varepsilon$ ), it is always the best response for agent  $i$  to solve it. Moreover, when agent  $i$  is not active in the search process (the solving path does not reach agent  $i$ ), the best response is the action  $\emptyset$ . Therefore, the only situation in which the agent  $i$  could improve its utility is when it is active in the search process but cannot solve the task itself. Then the most profitable action would be to change the search process mechanism from random-walk to a deterministic one. This means that agent  $i$  either plays  $\emptyset$  or forwards  $\rho$  to a fixed neighbour (denoted by  $i_k$ ) instead of forwarding  $\rho$  to a randomly chosen neighbor.

Let's first assume that agent  $i$  always plays  $\emptyset$  for any history if it cannot know how to solve the task. We denote the above strategy by  $\sigma_i^\emptyset$ . The utility function is:

$$u_i(\theta_i, \sigma_i^\emptyset, \sigma_{-i}^\infty) = P_i^\infty (\gamma(1 - |\rho - \theta_i|))$$

which is strictly smaller than

$$u_i(\theta_i, \sigma_i^\infty, \sigma_{-i}^\infty) = P_i^\infty (\gamma(1 - |\rho - \theta_i|) + (1 - \gamma)(P_{k^*}^\infty(\alpha - c) + (1 - P_{k^*}^\infty)(-c)))$$

if  $\alpha \geq \frac{c}{P_{k^*}^\infty}$  for all  $i$ . Let  $\alpha = c \frac{\sum_{i \in N} N_i}{\tilde{N}}$ ; then the above inequality holds since  $P_j^\infty = \frac{N_j}{\sum_{i \in N} N_i}$ .

Now, consider the strategy that agent  $i$  always forwards the project to its neighbour  $i_k$  when the agent cannot afford the task. The above strategy by  $\sigma_i^\theta$ . Then, the utility agent  $i$  gets is:

$$u_i(\theta_i, \sigma_i^{i_k}, \sigma_{-i}^\infty) = P_i^\infty \gamma (1 - (\rho - \theta_i)) + (1 - \gamma) (\tilde{P}_{k^*}^\infty (\alpha - c) + (1 - \tilde{P}_{k^*}^\infty) (-c))$$

Note that probability  $\tilde{P}_{k^*}$  could be zero since cycles may emerge ( $\tilde{P}_{k^*} < P_{k^*}$ ); therefore  $u_i(\theta_i, \sigma_i^{i_k}, \sigma_{-i}^\infty) \leq u_i(\theta_i, \sigma_i^\infty, \sigma_{-i}^\infty)$ .  $\square$

The above proof allows us to extend the result to the game  $(\Gamma_\rho^\infty)^N$ :

**Corollary 1.** *There exists  $\alpha^* > 0$  such that the strategy profile  $(\sigma_1^\infty, \dots, \sigma_n^\infty)$  is a Nash Equilibrium in  $(\Gamma_\rho^\infty)^N$ .*

Now we enrich the model by introducing a “time” condition to solve the task. It makes sense to limit the rewards for efforts to solve or forward the task based on a time limit within which the task must be solved (i.e., efforts are only rewarded if the task is solved in less than a certain number of steps).

#### 4.3. Random-walk strategy with a finite number of steps

An interesting measure for establishing the limit of steps that a task  $\rho$  can take to be solved is the Mean First Passage Time (hereafter MFPT). The MFPT between two nodes  $i$  and  $j$  of a network is defined as the average number of steps to go from  $i$  to  $j$  in that particular network [38]. Therefore, we define the strategy  $\sigma_i^\tau$  for an agent  $i$  as the one which consists in forwarding the task to a randomly selected neighbor only if it has advanced a number  $t_i < \tau$  times, where  $\tau$  is the average MFPT of the network (which we formally define below). Note that now the full path the task has taken until it reaches an agent matters to him, so now the whole set  $H^t$  influences its utility and not only the previous step  $H^{t-1}$ .

According to [38], the MFPT from any agent to a particular agent  $j$  in a network is defined as:

$$\begin{aligned} \langle T_j \rangle &= \frac{1}{1 - \pi_j} \sum_{i=1}^N \pi_i T_{ij} = \\ &= \frac{1}{1 - \pi_j} \sum_{k=2}^N \left( \frac{1}{1 - \lambda_k} \psi_{kj}^2 \sum_{i=1}^N \frac{k_i}{k_j} \right) - \frac{1}{1 - \pi_j} \sum_{k=2}^N \left( \frac{1}{1 - \lambda_k} \psi_{kj} \sqrt{\frac{K}{k_j}} \sum_{i=1}^N \psi_{ki} \sqrt{\frac{k_i}{K}} \right) \end{aligned} \quad (8)$$

where  $k_i$  and  $k_j$  are the degree of agents  $i$  and  $j$ , respectively,  $\psi_k$  is the  $k$ th eigenvector of  $S$  corresponding to the  $k$ th eigenvalue  $\lambda_k$  and  $\pi_j = k_j/K$  (with  $K = \sum_{j=1}^N k_j$ ). The  $k$ th eigenvector value  $\lambda_k$  is calculated considering that  $S = D^{-\frac{1}{2}}MD^{-\frac{1}{2}}$ ,  $M$  being the adjacency matrix of the network,  $D$  being the diagonal degree matrix of the network, and the eigenvalues being rearranged as  $1 = \lambda_1 > \lambda_2 \geq \lambda_3 \geq \dots \geq \lambda_N \geq -1$ ).

According to [38],  $\sum_{i=1}^N \psi_{ki} \sqrt{\frac{k_i}{K}} = \sum_{i=1}^N \psi_{ki} \psi_{1i} = 0$ . Thus, the second term is equal to 0. So

$$\langle T_j \rangle = \frac{1}{1 - \pi_j} \sum_{k=2}^N \left( \frac{1}{1 - \lambda_k} \psi_{kj}^2 \sum_{i=1}^N \frac{k_i}{k_j} \right) = \frac{1}{1 - \pi_j} \frac{K}{k_j} \sum_{k=2}^N \frac{1}{1 - \lambda_k} \psi_{kj}^2 \quad (9)$$

We define the maximum number of steps that must be taken for every task  $\rho$  to be solved as the average  $\langle T_j \rangle$  for all  $j \in N$ , which we denote as  $\tau$ . Formally:

$$\tau = \sum_{j=1}^N \frac{\langle T_j \rangle}{N} \quad (10)$$

We now define a new game  $\Gamma_\rho^\tau$ . In this game, if it takes more than  $\tau$  steps to solve the task, the game ends and the collaborating agents get no reward. In the following, we explain the equilibrium strategies for the game  $\Gamma_\rho^\tau$ .

Let us define some new parameters that play a role in the new game: the number of steps a task has advanced until it reaches agent  $i$  is  $t_i$ ;  $Q_{i,k^*}^{\tau-t_i}$  is the probability that the task reaches an agent  $k^*$  starting from agent  $i$  in  $\tau - t_i$  or less steps, and  $P_{s,i}^\tau$  is the probability that the task reaches an agent  $i$  starting from agent  $s$  in  $\tau$  or less steps.

In order to formally define  $P_{s,i}^\tau$ , we use the adjacency matrix of the network (denoted as  $M$ ) and one of its properties which states that the value at  $(i, j)$  position of the matrix  $M^n$  indicates the number of paths of length  $n$  between  $i$  and  $j$  in that network.  $P_{s,i}^\tau$  can be defined as the number of paths of length  $\tau$  or less between  $s$  and  $i$  divided by the total number of paths with the same length starting at  $s$  but ending at any possible agent  $j$  of the network. Formally:



$$P_{s,i}^\tau = \frac{\sum_{t=1}^{\tau} (M^t)_{si}}{\sum_{j=1}^N \left( \sum_{t=1}^{\tau} (M^t)_{sj} \right)} \quad (11)$$

Let us define  $r_i^{\tau-t_i}$ , or simply  $r_i$ , as the number of agents that the task can reach starting from  $i$  in  $\tau - t_i$  or less steps. For this purpose, we use the Reachability matrix, (denoted as  $R$ ), which is defined  $\forall i, j \in N$ , as

$$(R^{\tau-t_i})_{ij} = \begin{cases} 1 & \text{if there exists at least one path between } i \text{ and } j \\ & \text{of length } \tau - t_i \text{ or less} \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

The process for obtaining  $R$  from the adjacency matrix is straightforward. Then, we formally define  $r_i$  as

$$r_i = \sum_{j=1}^N (R^{\tau-t_i})_{ij} \quad (13)$$

To compute  $Q_{i,k^*}^{\tau-t_i}$ , let us define the probability that none of the reachable agents for agent  $i$  is able to solve the task, which is  $(1 - \gamma)^{r_i}$ . Then  $Q_{i,k^*}^{\tau-t_i}$  is:

$$Q_{i,k^*}^{\tau-t_i} = 1 - (1 - \gamma)^{r_i} \quad (14)$$

Hence, the utility function of the game  $\Gamma_\rho^\tau$  for an agent  $i$  when all agents play the strategy  $\sigma^\tau$  is

$$u_i(\theta_i, \sigma_i^\tau, \sigma_{-i}^\tau, s) = P_{s,i}^\tau (\gamma(1 - |\rho - \theta_i|) + (1 - \gamma)(Q_{i,k^*}^{\tau-t_i}(\alpha - c) + (1 - Q_{i,k^*}^{\tau-t_i})(-c))) \quad (15)$$

**Proposition 2.** *There exists  $\alpha^* > 0$  such that  $\forall \alpha > \alpha^*$  the strategy profile  $(\sigma_1^\tau, \dots, \sigma_n^\tau)$  is a Nash Equilibrium in the game  $\Gamma_\rho^\tau$ .*

*Proof.* The proof proceeds exactly like the proof for Proposition 1 but substituting the proper probabilities  $P_{s,i}^\tau$  and  $Q_{i,k^*}^{\tau-t_i}$  defined in 11 and 14, respectively.

We find a bound for  $\alpha_i$  for which agent  $i$  chooses strategy  $\sigma^\tau$  instead of  $\sigma^\emptyset$ . Specifically,  $\alpha_i \geq \frac{c}{Q_{i,k^*}^{\tau-t_i}}$ . By substituting 14, we have

$$\alpha_i \geq \frac{c}{1 - (1 - \gamma)^{r_i}} \quad (16)$$

The fact that  $\alpha_i$  depends on  $r_i$  implies that each agent has its own bound for  $\alpha$  which depends on the agent's connectivity. Let  $r = \max_{i \in N} r_i$ . Note that  $\frac{c}{1 - (1 - \gamma)^r} > \frac{c}{1 - (1 - \gamma)^{r_i}}$  for all  $i$  since  $0 < (1 - \gamma) < 1$  and  $\log((1 - \gamma)) < 0$ .

To complete the proof, let us recall the probabilities  $\tilde{P}_{s,i}^\tau$  and  $\tilde{Q}_{i,k^*}^{\tau-t_i}$  and the strategy  $\sigma_i^{i_k}$  that agent  $i$  always forwards the task to the same agent  $i_k$ . Since  $\tilde{P}_{s,i}^\tau \leq P_{s,i}^\tau$  and  $\tilde{Q}_{i,k^*}^{\tau-t_i} \leq Q_{i,k^*}^{\tau-t_i}$ , it is straightforward to show that, for  $\alpha^* = \frac{c}{1 - (1 - \gamma)^r}$ , we have that  $u_i(\theta_i, \sigma_i^{i_k}, \sigma_{-i}^\tau) \leq u_i(\theta_i, \sigma_i^\tau, \sigma_{-i}^\tau)$ . □

The above proof allows us to extend the result to the game  $(\Gamma_\rho^\infty)^N$ :

**Corollary 2.** *There exists  $\alpha^* > 0$  such that the strategy profile  $(\sigma_1^\tau, \dots, \sigma_n^\tau)$  is a Nash Equilibrium in the game  $(\Gamma_\rho^\tau)^N$ .*

The fact that  $\alpha$  depends on  $r_i$  means that the network structure has a deep impact on  $\alpha$  bounds. In highly clustered networks,  $r_i$  is high for each agent  $i$ , and, consequently,  $\alpha_i$  is low. The opposite occurs in low clustered networks (e.g., Erdős-Renyi networks) where  $\alpha_i$  is uniform among all agents. In networks with a non-uniform degree distribution (e.g., scale-free networks), average  $\alpha$  may be similar to the  $\alpha$  for Erdős-Renyi networks, but it varies significantly between hub and terminal agents.

## 5. Experiments

The goal of the experiments is to validate the proposed model not only formally, but also through a set of experiments considering both different types of network topologies as well as forwarding strategies. Therefore, the aim of the experiments is twofold: i) to determine which network topology enhances the emergence of cooperation in the proposed model; ii) to evaluate

the suitability of the proposed forwarding criterion for the search model and several network topologies.

In this section, we evaluate the proposed mathematical model for service search and compare it with other search strategies in different network structures. We analyze the influence of the structural parameters of networks in the required reward  $\alpha$  to promote cooperation (i.e., forwarding tasks) in the search process. To evaluate the influence of network structure, we compare the success rate of the searches and the average agent utility in different network topologies. The network topologies considered in the experiments are: random, scale-free, and small-world. We also analyze the criterion used in the forwarding process. To evaluate this criterion, we compare the success rate obtained in the search process when the forwarding criterion is the one proposed in the paper (i.e., random) with a domain-independent criterion based on the degree of neighbor nodes.

### 5.1. Experimental Design

The experiments are executed over undirected networks with 100 agents. We also tested different sizes of networks, but the conclusions were similar to those obtained with 100 agents and we do not include them here. The structural properties of the networks are shown in Table 1. Each agent has a type (i.e., service)  $\theta_i \in [0, 1]$  that represents its degree of ability. We assume that  $\theta_i$  is uniformly distributed among the agents. A task  $\rho$  is generated and assigned to an agent following a uniform probability distribution. Each agent has a set of actions to choose from when it receives a task: doing the task if the similarity between its ability and the task  $\rho$  is under a threshold  $|\theta_i - \rho| < \varepsilon$ ; forwarding the task based on the expected reward (Formula 16); or doing nothing. The forwarding action has an associated cost  $c = 5$ . A task  $\rho$  is successfully solved when an agent has an ability that is similar enough to the task ( $|\theta_i - \rho| < \varepsilon$ ) in less than  $\tau$  steps. For the experiments  $\tau = \log \text{MFPT}$ . We use this concave transformation to obtain clear results and illustrate the impact on the parameter with the structure of the network. The value for the  $\varepsilon$  parameter is 0.1. In order to motivate agents to solve a task  $\rho$ , instead of forwarding it to others, the value of the reward obtained when an agent  $i$  solves a task  $\rho$  (i.e.,  $1 - |\theta_i - \rho|$ ) is multiplied by a constant in order to obtain a higher reward than the reward  $\alpha$  associated to the forwarding action. We executed each experiment over 10 networks of each type and we generated 1,000 tasks  $\rho$  in each network. All agents chose their action following the  $\sigma^\tau$  strategy.

topology	N	edges	avDg	std	clust	dens	$\tau = \log \text{MFPT}$	d	diameter/ $\tau$
Random	100.0	200.00	4.00	1.46	0.02	0.04	5.00	7.09	1.418
	100.0	300.00	6.00	1.82	0.03	0.06	4.00	5.00	1.25
Scale-Free	100.0	197.00	3.94	3.94	0.02	0.04	5.00	5.09	1.09
	100.0	293.00	5.86	5.08	0.04	0.06	5.00	4.27	0.854
SmallWorld	100.0	200.00	4.00	1.02	0.08	0.04	5.00	7.55	1.55
	100.0	300.00	6.00	1.27	0.10	0.06	4.00	5.45	1.36

Table 1: Network structural properties: topology, number of agents, number of edges, average degree of connection of agents, standard deviation of the degree distribution, clustering, density,  $\tau = \log \text{MFPT}$ , diameter, ratio diameter/ $\tau$ .

In the experiments, we considered two different criteria in order to forward the query to a neighbor. One criterion is a random selection that has been used in the mathematical model presented in this paper. The other criterion is based on the degree of connection of the neighbors (i.e., the neighbor with the highest degree would be selected first in order to forward the query to).

### 5.2. The Influence of Structural Properties and the Reward Payoff

In this section, we analyze the influence of the structural properties of the network and the reward payoff  $\alpha$  in the search process. We consider values for  $\alpha$  in the range [4.99975, 5.0005] in order to see the effects on the search process (see Figure 3). In this interval, we observe the effects of considering values for  $\alpha$  that are lower than the cost of the forwarding action ( $c=5$ ), values that are equal to the cost of the forwarding action, and values that are greater than the cost of the forwarding action. With values of  $\alpha$  lower than or equal to  $c$ , the success rate was around 20%. This percentage represents the number of tasks that can be solved directly by the first agent that receives the task. Values of  $\alpha$  that are strictly superior to the cost of the forwarding action ( $\alpha > c$ ) provide an increase in the success rate of the search process (see Figure 3 left). The structural properties of the network considered in the search process have an important influence on the success rate. It can be that there are significant differences between the results in scale-free, random, and small-world networks. Scale-free provided better results than the other networks since its structural properties increased the number of agents that could be reachable. The diameter of the network is closer to  $\tau$  than the diameters of other network models (see Table 1). Another example of the influence of structural properties is the average degree of connection

of the agents. As the average degree of connection increases, the number of reachable agents increases and so does the probability of finding the required agent. Therefore, agents estimate that it is profitable to forward the task to their neighbors (see Figure 3 right).

If we compare the results using a random criterion to select the next neighbor to forward the query to with the results using a degree of connection criterion, it can be observed in Figure 3 (bottom) that there are no major differences in the percentage of success achieved. There is little difference in the percentage of success in the  $\alpha$  values that are close to the cost of the action. In such cases, the increase of the success is a bit faster than in the case based on the degree of connection strategy.

The structural properties and the reward value  $\alpha$  also influence the average utility obtained by an agent. In this experiment, we analyzed values of  $\alpha$  in the range  $[0, 60]$ . We considered a wider range in order to see the values that made the average agent utility positive and how this utility evolves (see Figure 4). Values of  $\alpha$  lower than or equal to  $c$  provided a utility equal to 0 since agents estimated that the expected reward was not enough to compensate the cost of the forwarding action. Values of  $\alpha$  that were in the interval  $(5, 10]$  made some agents estimate that the forwarding action was going to be profitable. Although the value for  $\alpha$  was enough for agents to consider forwarding tasks, their utility was not always positive for all of the agents. Therefore, the average utility had a negative value. The interval  $(5, 10]$  for  $\alpha$  values could be considered risky. The average utility became positive with  $\alpha$  values greater than 10 (see Figure 4 left). In this experiment, the network structure also had a significant influence. The scale-free network provided higher values of utility than the random or small-world networks. This difference was also observed when we increased the average degree of connection of agents (see Figure 4 right).

If we compare the results using a random criterion to select the next neighbor that forwards the query to with the results using a degree of connection criterion, it can be observed in Figure 4 (bottom) that there are differences in the average profit per agent achieved. The random criterion obtained better results than the degree-based criterion.

## 6. Conclusions

In this paper, we have analyzed the distributed search of resources in networks that model societies of agents. These agents offer services and interact

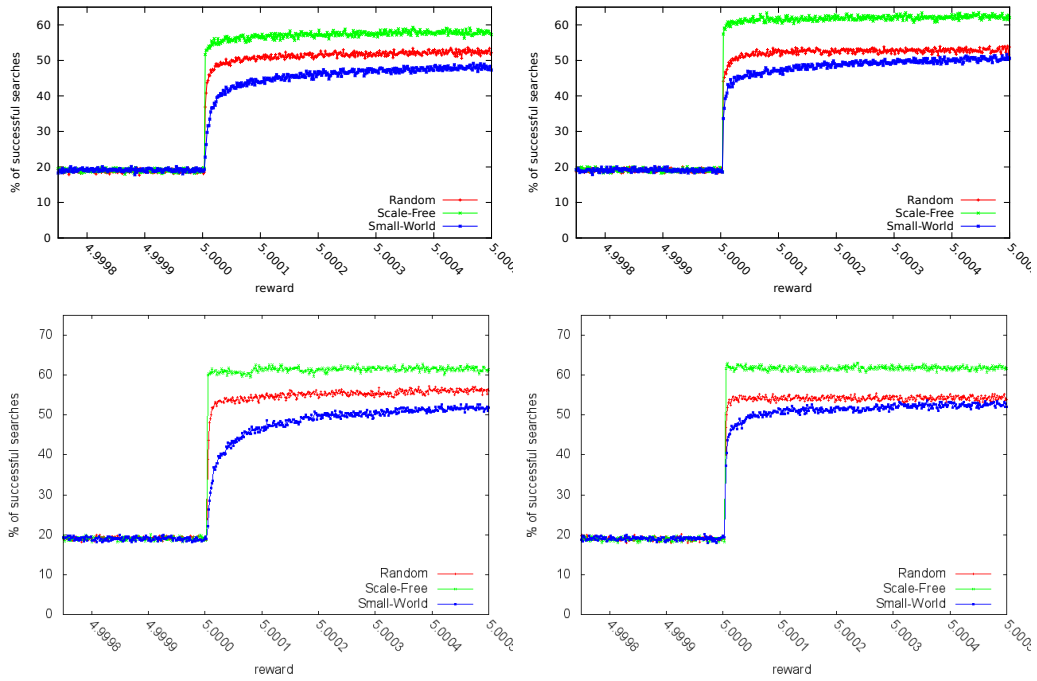


Figure 3: Influence of  $\alpha$  values on the percentage of successful searches in different network structures of 100 agents using different criteria for choosing the next node to forward the query to. Top left: network structures with an average degree of connection of 4 and random neighbor selection. Top right: network structures with an average degree of connection of 6 and random neighbor selection. Bottom left: network structures with an average degree of connection of 4 and degree-based neighbor selection. Bottom right: network structures with an average degree of connection of 4 and degree-based neighbor selection.

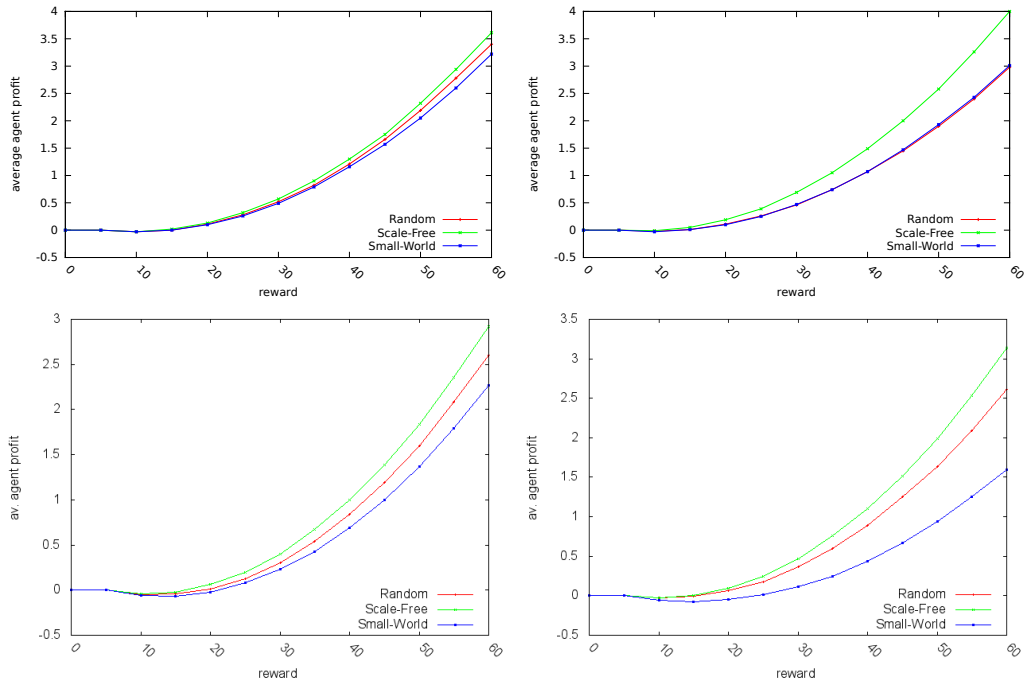


Figure 4: Influence of  $\alpha$  values on the utility in different network structures of 100 agents. Top left: network structures with an average degree of connection of 4 and random neighbor selection. Top right: network structures with an average degree of connection of 6 and random neighbor selection. Bottom left: network structures with an average degree of connection of 4 and degree-based neighbor selection. Bottom right: network structures with an average degree of connection of 6 and degree-based neighbor selection.

with each other by providing and consuming these services. The actions of these agents have an associated cost and not all of the agents have homogeneous behavior. In order to model this type of scenario, we have proposed the use of Game Theory to formally model the interactions between the agents as a repeated game. We have described a strategy for the search process that is based on the random-walk strategy, which facilitates the applicability of the model to a wider range of systems since it is domain independent. We have also established the conditions under which the random-walk strategy is a Nash Equilibrium. The strategy proposed has been extended by adding a constraint for contexts where the number of times a task can be forwarded is restricted. The conditions under which this extended strategy is a Nash Equilibrium have also been analyzed. Finally, we evaluated the proposed model and the latest strategy in different types of networks, and we compared the results obtained with the results using a degree-based search strategy. The results show that in order to promote cooperation among the agents of the network, the expected reward should be greater than the cost of the forwarding action. Moreover, the network structure has an important influence on the success of the search process and on the average utility of the system. scale-free structural parameters facilitate the success of the search process because their structural properties increase the number of agents that can be reached. The experiments also show that even though there are certain values of the reward that are enough to promote cooperation, these values are not enough to obtain a positive average utility value.

## 7. Acknowledgments

This work is supported by TIN2011-27652-C03-01, TIN2012-36586-C03-01, PROMETEOII/2013/019, Project ECO2013-46550-R, FEDER, PROMETEOII/2014/054], and the Program *Valorización y Recursos Conjuntos de I+D+i VLC/CAMPUS and the Ministerio de Educación, Cultura y Deporte* as part of the Campus of International Excellence Program SP2014800.

## 8. References

- [1] Lars Backstrom and Jure Leskovec. Supervised random walks: Predicting and recommending links in social networks. 2011.
- [2] Xiaomeng Ban, Jie Gao, and Arnout van de Rijt. Navigation in real-world complex networks through embedding in latent spaces. In *Meeting*



- on *Algorithm Engineering and Experiments (ALENEX10)*, pages 138–148, 2010.
- [3] Daniel O. Cajueiro. Optimal navigation in complex networks. *Physical Review E*, 79(4), 2009.
  - [4] Yatin Chawathe, Sylvia Ratnasamy, Lee Breslau, Nick Lanham, and Scott Shenker. Making gnutella-like p2p systems scalable. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '03, pages 407–418. ACM, 2003.
  - [5] L. F. Costa, O. N. Oliveira Jr., G. Travieso, F. A. Rodrigues, P. R. Villas Boas, L. Antiqueira, M. P. Viana, and L. E. C. Rocha. Analyzing and modeling real-world phenomena with complex networks: a survey of applications. *Advances in Physics*, 60(3):329–412, 2011.
  - [6] Mathijs M de Weerd, Yingqian Zhang, and Tomas Klos. Multiagent task allocation in social networks. *Autonomous Agents and Multi-Agent Systems*, 25(1):46–86, 2012.
  - [7] E. Del Val, M. Rebollo, and V. Botti. An Overview of Search Strategies in Distributed Environments. *The Knowledge Engineering Review*, 2013.
  - [8] E. Del Val, M. Rebollo, and V. Botti. Promoting Cooperation in Service-Oriented MAS through Social Plasticity and Incentives. *Journal of Systems and Software*, 86(2):520–537, 2013.
  - [9] E. Del Val, M. Rebollo, and V. Botti. Enhancing Decentralized Service Discovery in Open Service-Oriented Multi-Agent Systems. *Journal of Autonomous Agents and Multi-Agent Systems*, 28(1):1–30, 2014.
  - [10] Rachid El-Azouzi, Francesco De Pellegrini, and Vijay Kamble. Evolutionary forwarding games in delay tolerant networks. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2010 Proceedings of the 8th International Symposium on*, pages 76–84. IEEE, 2010.
  - [11] Ghofrane Fersi, Wassef Louati, and Maher Ben Jemaa. Distributed hash table-based routing and data management in wireless sensor networks: a survey. *Wireless networks*, 19(2):219–236, 2013.

- [12] Agata Fronczak and Piotr Fronczak. Biased random walks in complex networks: The role of local navigation rules. *Phys. Rev. E*, 80:016107, Jul 2009.
- [13] Drew Fudenberg and Eric Maskin. The Folk Theorem in Repeated Games with Discounting or with Incomplete Information. *Econometrica*, 54(3):533–554, 1986.
- [14] Christos Gkantsidis, Milena Mihail, and Amin Saberi. Random walks in peer-to-peer networks: Algorithms and evaluation. *Performance Evaluation*, 63(3):241–263, March 2006.
- [15] Xiaohui Gu, Zhen Wen, Philip S Yu, and Zon-Yin Shae. peertalk: A peer-to-peer multiparty voice-over-ip system. *IEEE Transactions on Parallel and Distributed Systems*, 19(4):515–528, 2008.
- [16] Lisa-Maria Hofmann, Nilanjan Chakraborty, and Katia Sycara. The evolution of cooperation in self-interested agent societies: A critical study. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 2, AAMAS '11*, pages 685–692, Richland, SC, 2011. International Foundation for Autonomous Agents and Multiagent Systems.
- [17] Hamed Janzadeh, Kaveh Fayazbakhsh, Mehdi Dehghan, and Mehran S. Fallah. A secure credit-based cooperation stimulating mechanism for manets using hash chains. *Future Gener. Comput. Syst.*, 25(8):926–934, September 2009.
- [18] Juan Jose Jaramillo and R. Srikant. A game theory based reputation mechanism to incentivize cooperation in wireless ad hoc networks. *Ad Hoc Networks*, 8(4):416–429, 2010.
- [19] Jon Kleinberg. Complex networks and decentralized search algorithms. In *Proceedings of the International Congress of Mathematicians (ICM)*, 2006.
- [20] Sungmin Lee, Soon-Hyung Yook, and Yup Kim. Searching method through biased random walks on complex networks. *Phys. Rev. E*, 80, 2009.

- [21] Ze Li and Haiying Shen. Game-theoretic analysis of cooperation incentive strategies in mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 11(8):1287–1303, August 2012.
- [22] Antonio L. Lopes and Luis M. Botelho. Improving multi-agent based resource coordination in peer-to-peer networks. *Journal of Networks*, 3:38–47, 2008.
- [23] M. Luck, P. McBurney, O. Shehory, and S. Willmott. *Agent Technology: Computing as Interaction (A Roadmap for Agent Based Computing)*. AgentLink, 2005.
- [24] Qin Lv, Pei Cao, Edith Cohen, Kai Li, and Scott Shenker. Search and replication in unstructured peer-to-peer networks. In *Proceedings of the 16th international conference on Supercomputing, ICS '02*, pages 84–95, New York, NY, USA, 2002. ACM.
- [25] Allen B. MacKenzie and Luiz A. DaSilva. *Game Theory for Wireless Engineers*. Synthesis Lectures on Communications. Morgan & Claypool Publishers, 2006.
- [26] M. E. J. Newman. The structure and function of complex networks. *SIAM REVIEW*, 45:167–256, 2003.
- [27] M. E. J. Newman. Resource letter CS-1: Complex systems. *American Journal of Physics*, 79(8):800–810, 2011.
- [28] Jae Dong Noh and Heiko Rieger. Random walks on complex networks. *Physical Review Letters*, 92:118701, 2004.
- [29] Jae Dong Noh and Heiko Rieger. Random walks on complex networks. *Physical review letters*, 92(11):118701, 2004.
- [30] Hisashi Ohtsuki, Christoph Hauert, Erez Lieberman, and Martin A. Nowak. A simple rule for the evolution of cooperation on graphs and social networks. *Nature*, 441(7092):502–505, 2006.
- [31] Cun-Lai Pu and Wen-Jiang Pei. Mixing search on complex networks. *Physica A: Statistical Mechanics and its Applications*, 389(3):587 – 594, 2010.

- [32] Josep M. Pujol, Jordi Delgado, Ramon Sangüesa, and Andreas Flache. The role of clustering on the emergence of efficient social conventions. In *IJCAI*, pages 965–970, 2005.
- [33] Naeem Ramzan, Emanuele Quacchio, Toni Zgaljic, Stefano Asioli, Luca Celetto, Ebroul Izquierdo, and Fabrizio Rovati. Peer-to-peer streaming of scalable video in future internet applications. *IEEE Communications Magazine*, 49(3):128–135, 2011.
- [34] Carles Sierra, Vicent Botti, and Sascha Ossowski. Agreement computing. *KI-Künstliche Intelligenz*, 25(1):57–61, 2011.
- [35] V. Srivastava, J. Neel, A. B. MacKenzie, R. Menon, L. A. Dasilva, J. E. Hicks, J. H. Reed, and R. P. Gilles. Using game theory to analyze wireless ad hoc networks. *Communications Surveys & Tutorials, IEEE*, 7(4):46–56, 2005.
- [36] Yuh-Min Tseng and Fu-Gui Chen. A free-rider aware reputation system for peer-to-peer file-sharing networks. *Expert Systems with Applications*, 38(3):2432–2440, 2011.
- [37] B. Yang and H. Garcia-Molina. Efficient search in peer-to-peer networks. In *Proceedings of the International Conference on Distributed Computing Systems (ICDCS)*, 2002.
- [38] Zhongzhi Zhang, Alafate Julaiti, Baoyu Hou, Hongjuan Zhang, and Guanrong Chen. Mean first-passage time for random walks on undirected networks. *The European Physical Journal B*, 84:691–697, 2011.
- [39] Ming Zhong. Popularity-biased random walks for peer-to-peer search under the square-root principle. In *Proceedings of the 5th International Workshop on Peer-to-Peer Systems (IPTPS)*, 2006.
- [40] Tao Zhou. Mixing navigation on networks. *Physica A: Statistical Mechanics and its Applications*, 387(12):3025 – 3032, 2008.