A Caching Algorithm using Evolutionary Game Theory in a File-Sharing System

Masahiro Sasabe
Cybermedia Center
Osaka University
1-32 Machikaneyama-cho, Toyonaka-shi
Osaka 560-0043, Japan
m-sasabe@cmc.osaka-u.ac.jp

Naoki Wakamiya, Masayuki Murata
Grad. School of Information Science and Technology
Osaka University
1-5 Yamadaoka, Suita-shi
Osaka 565-0871, Japan
{wakamiya, murata}@ist.osaka-u.ac.jp

Abstract

In a P2P file-sharing system, a node finds and retrieves its desired file. If multiple nodes cache the same file to provide others, we can achieve a file-sharing system with low latency and high file availability. However, a node has to spend costs, e.g., processing load or storage capacity, on caching of a file. Consequently, a node may selfishly behave and hesitate to cache a file. In such a case, there is a possibility that unpopular files disappear from the system. In this paper, we aim to accomplish effective caching in the whole system that emerges from autonomous and selfish node behavior. We first discuss relationship between selfish node behavior and system dynamics according to evolutionary game theory. As a result, we show that a file-sharing system can be robust to file disappearance depending on a cost and demand model for caching even if nodes behave selfishly.

1. Introduction

In a Peer-to-Peer (P2P) file-sharing system, a node exchanges information and files with other nodes on a logical network that the nodes construct by establishing a logical link between two nodes. Each node caches its original or retrieved files into a local storage to share them with other nodes. If multiple nodes cooperatively cache an identical file into their storages, a node can find it with higher probability and retrieve it more quickly. This leads to enhance performance, availability, and reliability of the system [6, 11].

However, caching of a file incurs costs such as processing load and storage capacity. A node consisting of the file-sharing system selfishly behaves because it is primitively a user terminal. As a result, some nodes may not be cooperative to cache files [3]. In such a situation, there is a problem that a node cannot use a file due to disappearance caused by its unpopularity. It is difficult for a system administrator to monitor and manage all nodes so that the system achieves effective caching. Thus, it is desirable that selfish and autonomous node behavior leads to effective caching in the whole system.

Game theory can reveal how the selfish node behavior affects dynamics of the whole system [5, 10]. In Ref. [5], they assume that a node behaves to minimize its own costs for using a file. The costs are storage capacity consumed by caching and latency to retrieve a file from its corresponding file holder, i.e., provider. Then, they prove that Nash equilibria, namely stable states of a system, exist in such a situation by using game theory. Furthermore, they show an optimum state of the system is equal to one of the Nash equilibria by introducing a payment model in which a node can obtain payments to cache a file from other nodes that retrieve the cache file. On the other hand, in Ref. [10], they discuss cooperative node behavior in a file-sharing system under the framework of Multi-Person Prisoner's Dilemma. If there is no incentive for caching a file, all nodes become free-riders in a Nash equilibrium. Through analyses and simulation evaluations, they present that a node intends to contribute to caching if it can obtain payments or reputations from other nodes in compensation for caching a file.

Although these studies reveal how selfish node behavior affects the system performance in a Nash equilibrium, they insist that incentive mechanisms, e.g., payments or reputations from other nodes, are essential to achieve cooperative caching. However, such incentive mechanisms are not necessarily applied to a file-sharing system. For example, there is a file-sharing system that hides a provider from nodes requesting the corresponding file so as to improve the anonymity among nodes. In such a system, it is difficult for a provider to obtain payments from the requesting nodes. In this paper, we focus on a more general file-sharing system without such incentive mechanisms. We aim to accomplish effective caching in the whole system that emerges from selfish and autonomous node behavior taking into account its demand and cost for caching. For this purpose,

we use evolutionary game theory to examine how local interaction between selfish nodes affects the dynamics of the whole system [7].

In a society of organisms, various individuals influence each other. Evolutionary game theory originally tries to figure out a mechanism in which optimum behavior comes down to offspring in evolutionary process of organisms by using game theory. Suppose that individual behavior defined by genes is a strategy of game theory and the number of offspring following certain behavior is payoffs acquired by the corresponding strategy. In such a case, various individuals are in strategically mutual dependence relation of game theory. Thus, by using game theory, we can explain the phenomenon that superior behavior spreads over a society of organisms through inheritance from ancestors to offspring. For example, many researchers focus on how individual behavior that seems to be selfish and uncooperative affects the emergence of cooperative phenomena in the whole society [7, 9, 12]. We expect that evolutionary game theory can also reveal how much selfish node behavior based on local interaction has impact on caching condition in the whole system. In this paper, we derive the relationship between node behavior and the number of cache files in a stable state of the whole system by using replicator dynamics and agent-based dynamics of evolutionary game theory. Then, we examine effective caching in the whole system that emerges from selfish node behavior. Replicator dynamics and agent-based dynamics are models to derive a strategy distribution in an evolutionary stable state.

Replicator dynamics is a mathematical model in which a strategy increases when it can yield more payoffs than average payoffs of all strategies. Note that replicator dynamics can be applicable when the number of individuals composed of the society is relatively large and the network among the individuals is mean-field like. As a result, we can obtain the system characteristics in the case that a node selfishly behaves in accordance with global information on caching condition in a full mesh network. On the other hand, agent-based dynamics models a phenomenon that a superior strategy spreads over the network in a hop-by-hop manner. In agent-based dynamics, an individual plays a game once with all neighboring individuals, determines superiority of its own strategy based on the game results, and finally decides the next strategy. Consequently, we can find out the system characteristics in the case that a node selfishly behaves based on its local information in various network topologies.

In this paper, we first make modeling a file-sharing system as a game taking into account cost and demand for caching. Then, we theoretically derive the number of cache files in the whole system by replicator dynamics when a node can obtain global information. Furthermore, through simulation experiments based on agent-based dynamics, we evaluate the number of cache files and distance from a node to a provider in the case that a node behaves according to its local information. Finally, we describe an appropriate caching model to achieve effective caching from the analyses and simulation results.

Table 1. payoff matrix (general)

player 1 \ player 2	cooperator	defector
cooperator	(R, R)	(S, T)
defector	(T, S)	(P, P)

The rest of the paper is organized as follows. In section 2, we describe models of caching games with which we deal in this paper. Then, we evaluate the system characteristics by replicator dynamics in section 3 and by agent-based dynamics in section 4. From the results, we examine whether the models are suited to effective caching. Finally, section 5 gives conclusions and future work.

2. Caching Game

In evolutionary game theory, a game between two arbitrary players is defined as a pay-off matrix as shown in Tab. 1. A defector exploiting a cooperator gets T and the exploited cooperator receives S. Both players receives R (P) when they cooperate (defect) each other. Prisoner's Dilemma Game ($T > R > P > S$) and SnowDrift Game ($T > R > S > P$) are examples of well-known games.

In this paper, we first model caching in a file-sharing system as a caching game between two neighboring nodes. If there is no capacity limit on nodes that results in no competition of storage capacity among different files, we can assume that the caching is composed of multiple separated caching games each of which deals with a single file. In what follows, we define caching games for a single file for sake of simplicity. For the single file caching, each node has two strategies: caching or no caching. Since a node satisfies its demand to the file by leveraging the file, we regard the benefit obtained by the use of the file as its demand b. On the other hand, a node has to spend a cost on caching a file. In this paper, we investigate two kinds of costs: processing load c_l caused by self or other node's access to a file and storage capacity c_s consumed by caching a file. As future work, we plan to consider dynamic costs, e.g., network bandwidth or delay required to retrieve a file from a provider. We do not consider the incentive mechanisms for caching [5, 10] because we assume a general file-sharing system in this paper.

In the case that the cost is the processing load, the parameters of payoff matrix become $R = b - c_l, T = b, S = b - 2c_l, P = 0$ (Tab. 2). Note that $b - 2c_l$ is greater than 0. Since the relationship among parameters of Tab. 2 sat-

Table 2. payoff matrix (caching game, cost: load)

player 1 \ player 2	caching	no caching
caching	$(b-c_l, b-c_l)$	$(b-2c_l, b)$
no caching	$(b, b-2c_l)$	(0,0)

Table 3. payoff matrix (caching game, cost: storage)

player 1 \ player 2	caching	no caching
caching	$(b-c_s, b-c_s)$	$(b-c_s, b)$
no caching	$(b, b-c_s)$	(0,0)

isfies $T > R > S > P$, Tab. 2 is the same as SnowDrift Game. On the contrary, we obtain $R = b - c_l, T = b, S = b - c_l, P = 0$ in the case that the cost is the storage capacity (Tab. 3). Note that $b - c_s$ is greater than 0. Tab. 3 holds $T > R = S > P$ and does not correspond to any kind of existing games.

In the succeeding sections, we explore how the number of nodes caching a file, that is the number of nodes taking the strategy "caching", changes in accordance with the parameters of payoff matrix and examine the realization of a file-sharing system with high file availability.

3. Theoretical Analysis by Replicator Dynamics

In this section, we theoretically derive the relationship between the parameter settings in Tabs. 2 and 3 and the number of nodes caching a file in a stable state of a file-sharing system by replicator dynamics [7]. If the payoff matrix can enforce more nodes to take the strategy "caching," we can enhance the file availability of the file-sharing system composed of nodes that selfishly behave based on the payoff matrix.

3.1. Replicator Dynamics

The basic concept of replicator dynamics is that the growth rate of nodes taking a strategy is proportional to the payoff acquired by the strategy. Thus, the strategy that yields more payoff than average payoff of the whole system increases, and vice versa. Note that replicator dynamics assumes that an node plays a game with all other nodes.

In what follows, we explain the derivation process of replicator dynamics in a general game described as Tab. 1. Suppose that the ratio of cooperators at a certain time is x and that of defectors is $1 - x$. The average payoff of cooperators becomes

$$Rx + S(1-x). \tag{1}$$

On the contrary, that of defectors is

$$Tx + P(1-x). \tag{2}$$

Consequently, the average payoff of the whole nodes can be derived as follows:

$$x(Rx + S(1-x)) + (1-x)(Tx + P(1-x)). \tag{3}$$

The payoff difference between the average payoff of cooperators and that of the whole nodes is expressed as

$$\begin{aligned} & \{Rx + S(1-x)\} \\ & -\{x(Rx + S(1-x)) + (1-x)(Tx + P(1-x))\} \\ = \ & \{(R+P-T-S)x + S - P\}(1-x). \end{aligned} \tag{4}$$

Finally, replicator dynamics $\dot{x}$ that indicates the transition of x is defined as follows:

$$\dot{x} = \{(R+P-T-S)x + S - P\}(1-x)x. \tag{5}$$

3.2. The Number of Cache Files in a Stable State

The equilibria of x that satisfy $\dot{x} = 0$ are 0, 1, and $\frac{P-S}{R+P-T-S}$. Next, we describe the stability of their equilibria. In Eq. (5), $(1-x)x$ is constantly over zero for $0 \leq x \leq 1$. The remaining part $(R+P-T-S)x+S-P$ becomes positive for $x < \frac{P-S}{R+P-T-S}$ and negative for $x > \frac{P-S}{R+P-T-S}$ because of the definition of games in section 2. As a result, x approaches to $\frac{P-S}{R+P-T-S}$ independently of the initial value of x. Thus, the stable equilibrium is

$$x = \frac{P-S}{R+P-T-S}. \tag{6}$$

Eq. (6) indicates that the ratio of cooperators at a stable state depends on the parameters of the payoff matrix. Based on Refs. [9, 12, 8], we first define r as the cost-to-benefit ratio of mutual cooperation. r means a risk that a node should take when it behaves as a cooperator. r ranges (0,1]. The smaller value of r indicates that cooperators increase. In this paper, r is determined by b and c_l or by b and c_s. Therefore, r can be regarded as the ratio of demand to cost for caching.

If the cost is the processing load, from Eq. (6) and Tab. 2, the ratio x of cooperators at a stable state becomes

$$x = \frac{b-2c_l}{b-c_l}. \tag{7}$$

On the other hand, the cost-to-benefit ratio of mutual cooperation r is as follows:

$$r = \frac{c_l}{b-c_l}. \tag{8}$$

Finally, the relationship between x and r satisfies

$$x = 1 - r. \quad (9)$$

We can also derive the relationship between x and r in the case where the cost is the storage capacity as follows:

$$x = \frac{1}{1+r}. \quad (10)$$

From Eqs. (9) and (10), the ratio x of cooperators deteriorates with the increase of the r, namely the decrease of the demand b to the file, independently of the cost models. In this paper, we aim to achieve a file-sharing system with high file-availability that is robust to file disappearance. If the cost is the processing load, Eq. (9) denotes that a file disappears from the system when r is close to 1 that means the demand to the file is low. On the other hand, the cost model of the storage capacity enhances the file availability by preventing x falling in 0 even if r is 1.

4. Simulation-based Analysis by Agent-based Dynamics

Agent-based dynamics [7] can reveal how much caching based on local interaction among neighboring nodes has impact on the number of cache files in the whole system. Moreover, we can evaluate search latency for a file since we can obtain the strategies of all nodes, namely the locations of cache files.

4.1. Agent-based Dynamics

In agent-based dynamics, a node determines its strategy by comparing its own payoff with that of a neighboring node. A node initially selects a strategy at random. The initial ratio of cooperators and that of defectors are fifty-fifty in Refs. [8, 9, 12]. Once a node i determines its strategy, it plays a game once with all neighboring nodes. This is one generation. At the end of the generation, the node i calculates average A_i of payoffs acquired, then determines the strategy of the next generation as follows.

Step 1: Selection of a neighboring node for comparison of payoffs

The node i randomly chooses a node j from neighboring nodes.

Step 2: Decision of the next strategy based on the comparison of average payoffs

If $A_j > A_i$ is satisfied, the node i imitates the strategy of node j with the following probability:

$$P_A(i,j) = \frac{A_j - A_i}{T - P}. \quad (11)$$

Otherwise, it does not change its strategy. The node i tends to imitate the strategy of a node that obtained more payoffs than it. In addition, P_A increases in proportion to the payoff difference.

Table 4. maximum hop count between two arbitrary nodes

	$m = 2$	$m = 4$
scale-free network	7.4	5
random network	8.7	6

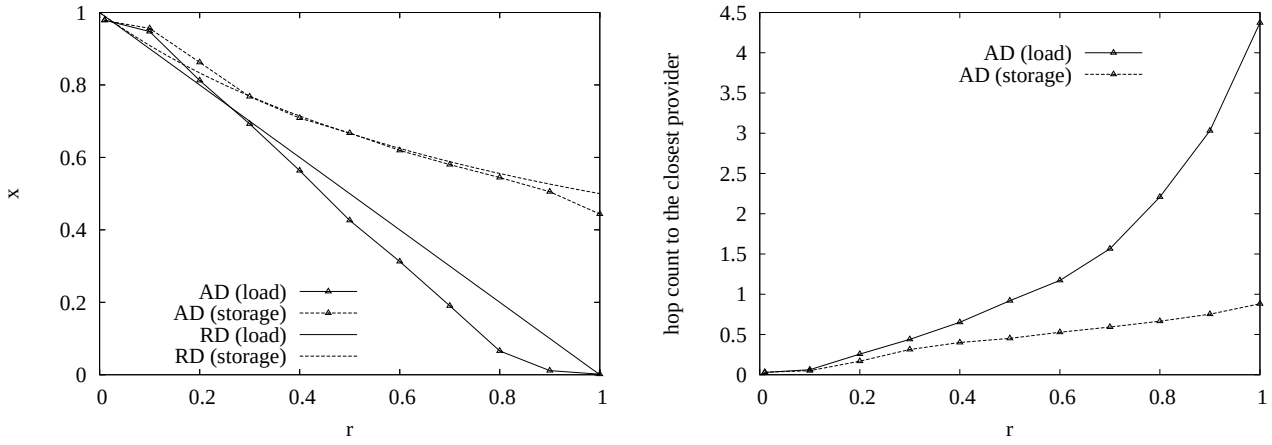
In an actual system, we should consider the overheads incurred by playing games and exchanging payoff with neighboring nodes. We expect that these processes can be realized by applying the keep-alive messaging as in Gnutella with slight modifications.

4.2. Simulation Experiments

Through several simulation experiments, we evaluate how node behavior based on the payoff matrix affects file availability and search latency of the whole system in scale-free and random networks. We evaluate the file availability by the ratio x of nodes taking the strategy "caching." In the case of single-file caching, the number of cache files is the same as the product of x and the number of nodes. On the other hand, we define the search latency as the average hop count between a node to its closest provider including itself. Note that we alternatively use the maximum hop count between two arbitrary nodes when a file disappears from the system.

4.2.1. Simulation Model. We used NetLogo [2] in our simulation experiments. Based on Refs. [8, 9, 12], we set simulation configurations as follows. We generated scale-free and random networks of 1000 nodes by using the topology generator BRITE [1]. The scale-free network was based on Barabási-Albert (BA) model [4] and the random network followed waxman algorithm [13] with $\alpha = 0.15$ and $\beta = 0.2$. We also set the number m of connections that a newly participating node established to 2 and 4. Table 4 represents the average of maximum hop count between two arbitrary nodes in twenty networks. We set the caching costs c_l and c_s to 1, respectively. Thus, independently of the cost models, $b = \frac{1+r}{r}$ was derived. We configured that the initial ratio of strategy "caching" and that of strategy "no caching" were fifty-fifty. To investigate the system characteristics in a stable state, we show results when 1000 generations passed. The following results indicate the average of twenty simulations. We abbreviate replicator dynamics to RD, agent-based dynamics to AD, scale-free to SF, and random to RND in the following figures.

4.2.2. Impact of Payoff Matrix. Figure 1(a) illustrates that the relationship between r and the ratio x of nodes taking



(a) fraction of cooperators

(b) hop count to the closest provider

Figure 1. payoff matrix vs. file availability and search latency (scale-free: m=4)

the strategy "caching" that is derived by agent-based dynamics in a scale-free network with $m = 4$. We discover that x deteriorates with the increase of r, namely the decrease of demand, regardless of the cost models. In addition, the cost model of processing load causes the disappearance of files with larger r. On the other hand, the file availability is enhanced by using the cost model of storage capacity even if r is 1. Next, Fig. 1(a) also shows x derived by replicator dynamics. If the cost is the processing load, x of agent-based dynamics is lower than that of replicator dynamics excluding the case of smaller r. On the contrary, the cost model of storage capacity enhances the file availability as x of agent-based dynamics achieves almost the same as that of replicator dynamics.

Figure 1(b) depicts that the search latency increases with the growth of r, independently of the cost models. This is because the number of providers decreases as shown in Fig. 1(a). In the case that the cost is the storage capacity, x is higher than 0.4 even for a low-demand file. Consequently, a node can reduce the search latency by finding out a closer provider.

4.2.3. Impact of Network Structure. Figure 2 illustrates x and the search latency in scale-free networks with $m = 2, 4$. Since the average degree is $2m$, larger m makes a network dense. Figure 2(a) shows that smaller m promotes to increase high-demand files, independently of the cost models. This is due to the effect of high degree nodes. A high degree node tends to acquire more payoffs than other nodes and be chosen for comparison of payoffs by its neighboring nodes. As a result, the strategy of a high degree node is likely to spread over the network. The impact of high degree nodes is accelerated in a network with small m where low degree nodes frequently exist.

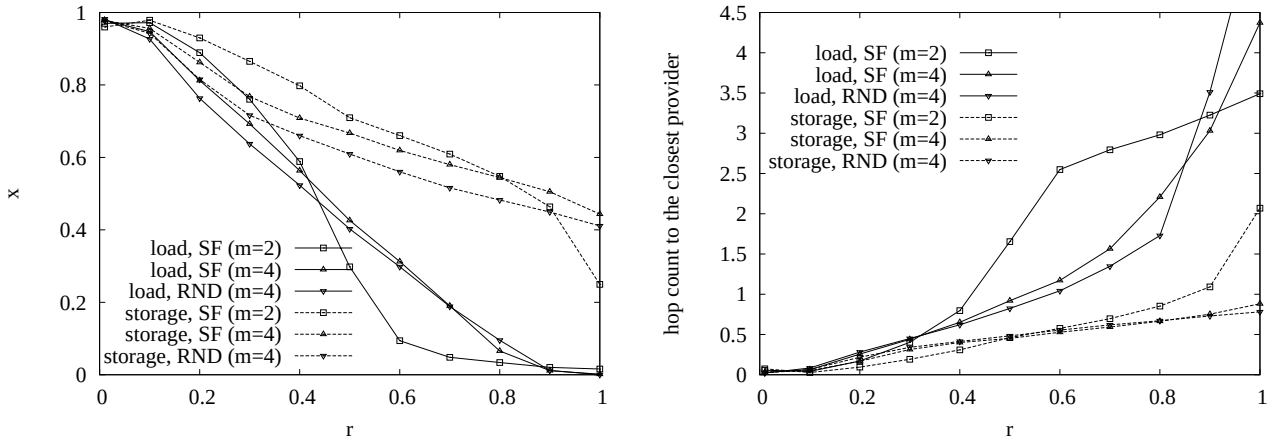
On the other hand, Fig. 2(b) presents that the search latency of $m = 2$ is lower than that of $m = 4$ if r is smaller than 0.3 in the cost model of processing load and 0.5 in the cost model of storage capacity. This is because files with smaller r are more cached in the case of $m = 2$ as shown in Fig. 2(a). Since the increase of r results in the decrease of providers, the search latency of $m = 4$ becomes superior to that of $m = 2$. Note that there is slight difference between $m = 2$ and $m = 4$ in the cost model of storage capacity.

Figure 2 also shows that x and the search latency in the random network with $m = 4$. We find that x in the scale-free network is mainly larger than that in the random network as shown in Fig. 2(a). Contrary to our expectation, Fig. 2(b) presents that the search latency in the scale-free network is not so superior to that in the random network despite of its lower diameter of the network (Tab. 4). This is because files tend to be cached at regular nodes rather than high degree nodes in the scale-free network. In other words, we can alleviate load concentration on high degree nodes while suppressing the search latency. Note that the difference of degree distribution does not so much affect to the search latency if the cost is the storage capacity.

In summary, we can accomplish a file-sharing system with high file-availability and low search-latency by using the cost model of storage capacity independently of the network structures.

5. Conclusions

In this paper, we revealed the relationship between node behavior and effective caching in the whole system by evolutionary game theory so as to accomplish a file-sharing system with high file availability and low search latency even if nodes behaved selfishly and autonomously. We first made modeling a file-sharing system as two kinds of caching games between two neighboring nodes. Then, we



(a) fraction of cooperators

(b) hop count to the closest provider

Figure 2. impact of network structure (m=2 vs. m=4, scale-free vs. random)

showed the basic characteristics of the models by analytically deriving the number of cache files in the system based on replicator dynamics. Furthermore, through simulation experiments based on agent-based dynamics, we evaluated how much local node behavior had impact on the system performance. Simulation results showed that modeling the cost for caching as the storage capacity made a file-sharing system robust to fie disappearance independently of the network structures even if nodes behave selfishly.

As future research work, we should examine a caching game that takes into account combination of multiple costs including dynamic costs, e.g., delay and bandwidth. Furthermore, we plan to analyze the dynamics of information networks other than the file-sharing system and propose control mechanisms based on the analysis. For example, information travels along a chain of intermediate nodes toward its destination in the following information systems: information distribution on an overlay network, e.g., application level multicast, and information diffusion and gathering on a sensor network. Since forwarding information takes costs, such as network bandwidth and electricity consumption, an effective forwarding mechanism taking into account the significance of information is needed. Evolutionary game theory can reveal node behavior suitable for such effective information transfer.

Acknowledgement

This research was supported by a Grant-in-Aid for Young Scientists (B) 17700058 and "New Information Technologies for Building a Networked Symbiosis Environment" of The 21st Century Center of Excellence Program of the Ministry of Education, Culture, Sports, Science and Technology in Japan.

References

[1] BRITE. available at `http://www.cs.bu.edu/brite/`.

[2] NetLogo. available at `http://ccl.northwestern.edu/netlogo/`.

[3] E. Adar and B. A. Huberman. Free Riding on Gnutella. *Technical Report Xerox PARC*, 2000.

[4] A.-L. Barabási and R. Albert. Emergence of Scaling in Random Networks. *Science*, 286, 1999.

[5] B.-G. Chun, K. Chaudhuri, H. Wee, M. Barreno, C. H. Papadimitriou, and J. Kubiatowicz. Selfish Caching in Distributed Systems: A Game-Theoretic Analysis. In *Proceedings of PODC 2004*, pages 21–30, Aveiro, July 2004.

[6] K. P. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy, and J. Zahorjan. Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload. In *Proceedings of SOSP 2003*, pages 314–329, New York, Oct. 2003.

[7] S. Gyorgy and F. Gabor. Evolutionary Games on Graphs. *ArXiv Condensed Matter e-prints*, July 2006.

[8] C. Hauert and M. Doebeli. Spatial Structure Often Inhibits The Evolution of Cooperation in The Snowdrift Game. *Nature*, 428, 2004.

[9] J. M. Pacheco and F. C. Santos. Network Dependence of the Dilemmas of Cooperation. In *Proceedings of CNET 2004*, pages 90–100, Aveiro, Aug. 2004.

[10] K. Ranganathan, M. Ripeanu, A. Sarin, and I. Foster. Incentive Mechanisms for Large Collaborative Resource Sharing. In *Proceedings of CCGRID 2004*, pages 1–8, Washington, DC, 2004.

[11] A. Rowstron and P. Druschel. Storage Management and Caching in PAST, a Large-Scale, Persistent Peer-to-Peer Storage Utility. In *Proceedings of SOSP 2001*, pages 188–201, New York, Oct. 2001.

[12] F. C. Santos and J. M. Pacheco. A New Route to the Evolution of Cooperation. *Journal of Evolutionary Biology*, 19(3):726–733, 2006.

[13] B. M. Waxman. Routing of Multipoint Connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617–1622, Dec. 1988.