How to Build the Network of Contacts

Selecting the Cooperative Partners

Pedro Mariano[1], Luís Correia[2], and Carlos Grilo[2,3]

[1] Transverse Activity on Intelligent Robotics – IEETA – DETI, Universidade de Aveiro, Portugal
[2] LabMAg – Dep. de Informática, Faculdade de Ciências, Universidade de Lisboa, Portugal
[3] Dep. Eng. Informática, Escola Superior de Tecnologia e Gestão Instituto Politécnico de Leiria, Portugal

Abstract. We address the problem of finding the correct agents to interact with from a general standpoint. We take the payoff obtained by agents in any game with dilemma as an input to our model. Its output is a probability distribution used in the partner selection that increasingly favours cooperative agents. Our approach contrasts with others designed for specific games without concerns of generality. We show both theoretically and experimentally that the major factor affecting cooperators selecting only themselves is the agents' strategies. This result does not depend on game nature or the initial probability distribution.

1 Introduction

Models of reputation or the chance of punishing are typically models analysed in situations where a dilemma is present. Reputation is used by agents as a measure of how well one behaves, and thus leads to one agent being favoured in detriment of others. Punishment is used as a method to discourage the proliferation of non-cooperators [7]. These models are often used with concrete, but representative, games [13, 2].

Such models are put forward to explain the prevaillance of cooperation in a population of interacting agents. Typically such models assume that the agents are laid in some lattice that restricts with whom an agent may interact [20]. Graphs are often used to describe such restrictions. Such graphs are also called networks of contacts. There are also models that focus on which type of network favours the development of cooperation [12].

The model that we present focuses on partner selection by agents. Our agents are able to select with whom they wish to interact. This approach is often taken on models of reputation [9, 18, 19, 7], where an agent maintains some vector that classifies its potential partners. This vector is then used to pick up the most promising partners.

We will consider that the only information an agent gets from an interaction is its payoff. It does not know which actions were taken by its partners. This feature allows an agent to select its partners, only requiring knowledge about the payoffs of whichever game is used for interaction.

The motivation of the model presented in this paper is to address the problem of how cooperative agents are able to survive in a population subject to interaction constraints. There are several approaches to instill cooperative behaviour in a population of interacting agents. This is a topic of much research in Game Theory (see for instance [6, 8, 5, 21, 1, 4, 8]), Sociology, Biology (see [16, 13, 3]), and Artificial Intelligence (see [9, 19, 22]), to name a few areas.

As we use Game Theory to model the interactions between agents, our model makes direct use of the payoffs of a game in order to select the cooperative partners. The model consists in a probability vector maintained by each agent that in each position represents the probability of selecting an agent as a partner to play a game.

Since the agent has to find the best partner, the algorithm can be compared to a Cournot adjustment process [10] were players iteratively adjust their strategies to their partner responses. In this paper, an agent strategy remains constant but it adjusts its preferences towards more profitable or cooperative partners. Similar approaches to partner selection have been tackled in [14] but they focused on a specific game such as Prisoner's Dilemma (PD). With the approach we now propose, we are able to do partner selection in any situation capable of being described as a game, and we are not limited to any particular game.

2 Definitions

We will use Game Theory as the tool to model interaction between agents. To this end, we will consider that a population $\mathbb{P}$ of agents interacts accordingly to the rules of some n-player game $\mathcal{G}$. The game describes the strategies available to players and the payoffs they obtain as a function of the strategies used. The game has a n-dimensions strategy space $\mathbf{S} = \mathbb{S}_1 \times \mathbb{S}_2 \times \ldots \times \mathbb{S}_n$ where agents can draw a strategy $s_i \in \mathbb{S}_i$ to play a game. The vector $\mathbf{s} = (s_1, \ldots s_n)$ represents a strategy profile of the n players involved in the game. The game also has n payoff functions, $u_i : \mathbf{S} \to \mathbb{R}$, with $i \in \{1, 2, \ldots, n\}$. The payoff values are bounded and belong to $\mathbb{R}$. Let $\underline{u}$ be the lowest payoff and $\overline{u}$ be the highest payoff in game $\mathcal{G}$.

Agents are composed of a strategy $s_i \in \mathbb{S}_i$. This means agents are characterised by the role they play in game $\mathcal{G}$. As an example, a game may model a buyer/seller scenario. Generally speaking, the game is asymmetric. This means the following condition is true:

$$\exists i \exists j : i \neq j \implies \mathbb{S}_i \neq \mathbb{S}_j .$$

An agent to play a game $\mathcal{G}$ must select $n-1$ agents of the appropriate role, i.e. each one with a strategy s_j with $j \neq i$. If the game is symmetric, then all the strategy sets $\mathbb{S}_i$ are equal.

We also aim at reaching a position where cooperative agents only interact between themselves. As cooperative agents we define those that form a strategy profile that maximises the average payoff of the players. We define the payoff obtained by such Pareto Optimal profile as follows:

$$u_P = \max_{\mathbf{s}} \sum_i \frac{u_i(\mathbf{s})}{n}.$$

As an example, in the Iterated Prisoner's Dilemma (IPD) game [11], a cooperative agent is one that does not defect, and in a Public Good Game (PGG) [11] a cooperative agent is one that contributes to the common good.

Table 1 summarises the nomenclature used throughout this document.

α, β, γ	agents
$\mathbb{P}$	the population of agents
$\mathcal{G}$	a n-player game
$\overline{u}$	highest payoff in game $\mathcal{G}$
$\underline{u}$	lowest payoff in game $\mathcal{G}$
u_P	payoff obtained by an agent in a Pareto Optimal profile
$w_{\alpha,\beta}$	label of the edge from agent α to agent β

Table 1. Nomenclature used in this document

3 Model Description

A population $\mathbb{P}$ of agents is represented by a directed simple graph where a vertex represents an agent α and a labelled edge from α to β represents the probability of agent α interacting with β.

$$\mathbb{P} = (V, E)$$

with:

$$\begin{aligned} V &= \{\alpha, \beta, \gamma, \ldots\}, \\ E &= \{(\alpha, \beta, w_{\alpha,\beta}), \ldots\}, \\ w_{\alpha,\beta} &\geq 0, \\ \sum_{\beta} w_{\alpha,\beta} &= 1. \end{aligned}$$

This definition allows us to represent the directed simple graph by a matrix. If there is no link from agent α to β, then it is assumed that probability $w_{\alpha,\beta}$ is zero.

Each agent performs the following algorithm:

1. partner selection,
2. play the game,

3. update $w_{\alpha,\beta}$ probabilities.

In the first iterations of the algorithm, the agent must find the best partners and increase their probability of being selected, while decreasing the chance of selecting bad partners. Our model must strike a balance between exploring new neighbours and exploiting the best agents it has found so far.

The algorithm can be run in a distributed fashion by each agent. No communication between agents is necessary, nor any central control is required. The $w_{\alpha,\beta}$ matrix could be split in vectors, each one maintained by the corresponding agent.

3.1 Update policy

The edge weight update policy for agent α is a function defined as follows:

$$w_{\alpha,\beta}^{t+1} = \zeta(w_{\alpha,\beta}^{t}, u_{\alpha}^{t})$$

where $w_{\alpha,\beta}^{t}$ is the edge weight before the game and u_{α}^{t} is the payoff agent α obtains in the game.

The main focus of the work presented in this paper is the analysis of an update policy that meets the following two conditions:

Cooperative aggregation Cooperative agents are mostly connected to each other. If α and a neighbour β are part of a Pareto Optimal profile, then in the limit the sum of the probability of selecting all β should be 1:

$$\sum_{\beta} \lim_{t\to\infty} w_{\alpha,\beta}^{t} = 1 \quad \forall_{\beta}\, u_{\alpha}(\ldots, s_{\beta}, \ldots) = u_P. \tag{1}$$

Stability The update policy must be robust in order to resist perturbations in the population and to be applicable to any n-player game. In the long run and in the absence of perturbations, weights must stabilise:

$$\lim_{t\to\infty} (w_{\alpha,\beta}^{t+1} - w_{\alpha,\beta}^{t}) = 0. \tag{2}$$

We are assuming that the number of cooperative agents is equal or higher than the number of players and partner selection is done without replacement. Otherwise, a cooperative agent does not have enough partners to play a game.

The edge weight update policy is divided in two cases depending on whether an agent played the game with agent α or not.

Agent β played the game A simple policy is to multiply the old weight by a factor that is proportional to the distance between the payoff u obtained by agent α and the Pareto Optimal payoff u_P:

$$w_{\alpha,\beta}^{t+1} = \begin{cases} w_{\alpha,\beta}^{t} \dfrac{u - \underline{u}}{u_P - \underline{u}} & u < u_P \\ w_{\alpha,\beta}^{t} & u = u_P \\ w_{\alpha,\beta}^{t} \dfrac{\overline{u} - u}{\overline{u} - u_P} & u > u_P. \end{cases}$$

This rule by itself does not guarantee the condition stated by equation (1). Only combined with the rule for the case of agents that were not selected we achieve it. Regarding the stability, this rule will keep weights unchanged if the payoff is equal to u_P. Otherwise they will tend to zero as in the first and third cases $w^t_{\alpha,\beta}$ is multiplied by a factor always less than 1. Either way, equation (2) is met.

This policy from the viewpoint of an uncooperative agent is irrational. Such agents defect in the PD, do not provide the good in PGG, keep all the money in the Ultimatum. Their payoff is often greater than u_P. They should keep selecting the partner in order to exploit him. Instead of a single peak, the update policy should have a cutoff threshold:

$$w^{t+1}_{\alpha,\beta} = \begin{cases} w^t_{\alpha,\beta} \dfrac{u - \underline{u}}{u_P - \underline{u}} & u < u_P \\ w^t_{\alpha,\beta} & u \geq u_P. \end{cases} \tag{3}$$

This rule also verifies the stability condition. Another characteristic is that if $u \geq u_P$ the agent will not change its probabilities and thus it will probabilistically play with the same agents. Since it is rational for all agents, we will consider it for the remainder of the paper.

Agent γ did not play the game The multiplicative factor used for agents that played the game implies that the weight of all agents that played will either stay the same or decrease. If they decrease, the difference must be distributed among the other edge weights. A simple solution is to distribute equally:

$$w^{t+1}_{\alpha,\gamma} = w^t_{\alpha,\gamma} + \frac{s}{x} \tag{4}$$

where s is the sum of the difference of all link values, egressing node α, that have played the game in the current round,

$$s = \sum_{\beta} (w^{t+1}_{\alpha,\beta} - w^t_{\alpha,\beta})$$

and variable x is the number of neighbours of agent α that were not selected.

This equation shows that this policy explores alternative partners if $u < u_P$, since the probability of others being selected in the next iteration is increased.

Equation (4) combined with equation(3) is able to achieve the condition expressed by equation (1). If a cooperative agent selects an uncooperative, the corresponding weight will decrease towards zero. The difference is distributed among the weights of players that were not selected. However, the weight of a second uncooperative partner also increases, but not by much. Even if this second partner is selected, its weight is reduced and distributed among all the partners. The point is that, in the long run, weights of uncooperative agents decrease while weights of cooperative agents will absorb the distributed differences.

4 Experimental Analysis

The purpose of the experiments reported in this paper is to assess the time cooperative agents take to only select themselves as partners of interaction. To this end, simulations with different proportion and quantities of cooperative agents were performed. Other conditions were also analysed such as initial probability values, different games and, in one case, different game parameters.

4.1 The Games

The selected games were the Give-Take game and IPD. These games have been chosen because they pose a dilemma. The first one is a 2-player game with several Pareto Optimal strategy profiles [15], and the second is commonly used.

The Give-Take game is played between two agents that must share a single resource. Only one agent can hold it and benefit from it at each iteration. That agent can give the resource to its partner. The partner can take (grab) the resource or do nothing. Their roles will remain the same, provided both players do nothing. Agents change role whenever the resource ownership is changed. Whenever an agent gives the resource away, it can receive a bonus, b_g. An agent that takes (grabs) the resource (from its partner) pays a penalty as its partner.

Two sets of parameters were used differing on the value of parameter b_g: possession of the resource per iteration was set to 1, giving the resource, b_g, yields 0.5 or 0 units, taking the resource costs 2 units to the performer and 1 unit to the subject of the action. The game had at least 100 iterations, and after the 100^{th} the probability of occurring one more iteration was set to 0.5. Due to implementation decisions, game length was limited to 1000 iterations.

The strategy used by the agents is deterministic and has two parameters: t_g number of iterations the agent waits (after obtaining the resource possession) before giving it to its partner; t_t number of iterations the agent waits (after loosing the resource possession) before taking it from its partner. These parameters allow us to have strategies that never give the resource ($t_g = \infty, t_t = x$) or that do nothing ($t_g = \infty, t_t = \infty$).

The payoffs of IPD were: temptation to defect 5; cooperate 3; suckers 0; penalty 1. Each game lasted at least 10 iterations, and after the 10^{th} the probability of occurring one more iteration was set to 0.5. The strategy used by agents is stochastic and has one parameter: the probability to cooperate.

4.2 The Population

Regarding the agents, different strategies where used, which can be roughly classified in how cooperative they are. The strategies used in the Give-Take game are:

S1, S2, S3 These strategies always give the resource, but do so at different times: 1, 10 and 20 iterations. They never take the resource. Therefore, if both agents use the same strategy, their form a Pareto Optimal profile.

However, when agents use different strategies, the one that gives the resource sooner is explored;

S4 An uncooperative and aggressive strategy, as it never gives the resource, and takes it immediately after it looses its possession.

The strategies used in IPD are:

S1 A cooperative strategy that always cooperates;
S2, S3 Two strategies that cooperate with probabilities 0.7 and 0.3;
S4 One is uncooperative as it always defects.

The number of strategies in the population, for each type, varied between 2, 4, 8, 16 or 32 agents. This allows us to study the time taken for different proportions and quantities of cooperative strategies to mostly select themselves as partners. For instance 32 **S1** versus 2 of each **S2**, **S3** and **S4** is a scenario that favours cooperative agents. A scenario with 8 of each **S1**, **S2**, **S3** and **S4** is balanced for all strategies. On the contrary, a scenario with 2 **S1** versus 32 of **S4** and any number of **S2** and **S3** is very unfavourable for the cooperative agents as they will take longer in finding the correct partners.

Total population varied among $\{8, 10, \ldots, 112, 128\}$. This allows us to study how cooperative agents select themselves as interaction partners in different scenarios. We opted for a majority of cases with a small number of agents because we are interested in analysing the edge weights and viewing the resulting graph. For bigger populations we have 128^2 edges to examine, therefore, in this case, we only analyse global results. As for the initial edge weights, two options were used: identical values so that every agent has the same chance of being selected; random values.

4.3 Other Parameters

Each simulation consisted of 1000 rounds of games. In each round all agents played at least one game, since the following steps were performed per round for every agent: select $n-1$ partners proportionally to the edge weights, play the game, update the edge weights of the agent that selected partners. After each round the edge weights were recorded.

The simulation was developed in JDK 1.6. Random numbers were produced using an instance of the `cern.jet.random.engine.MersenneTwister` class [17], a pseudo-random number generator that has a large period of $2^{19937}-1$.

Table 2 shows all the conditions tested in the experiments. For each combination of conditions 10 runs were performed and results averaged.

4.4 Results

We have plotted the average probability of a group of agents (all with the same strategy) of selecting a group of agents (again all equal in terms of strategy). We have also calculated the standard deviation.

games	Give-Take $b_g = 0.5$ Give-Take $b_g = 0$ IPD
initial edge weight	random identical
# **S1**	2, 4, 8, 16, 32
# **S2**	2, 4, 8, 16, 32
# **S3**	2, 4, 8, 16, 32
# **S4**	2, 4, 8, 16, 32

Table 2. Summary of the conditions tested in the simulations.

Figure 1 shows the results from the simulations with Give-Take and $b_g = 0$. In this figure we additionally show the average probability at the 10000^{th} round. Figure 2 shows the probability graph obtained in a simulation with 8 agents.

Figure 3 shows some results from the simulations with Give-Take and $b_g = 0.5$ while figure 4 shows the results obtained with IPD. Since results are similar, there are fewer plots in these figures.

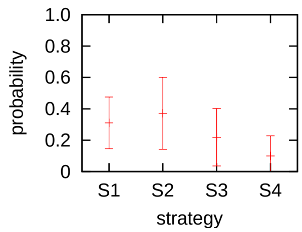
5 Discussion

In the following discussion we will use the term *aggregation time* to refer to the time taken for a group of strategies to mostly select themselves. That is to say, how many rounds of edge weight updating must occur in order to observe an approach to the condition stated by equation (1).

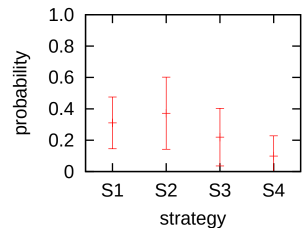
Initial edge weights do not influence aggregation time. This can be observed in figures 1(a) and 1(b). Both refer to Give-Take with $b_g = 0$ and two **S1** agents. Other numbers of **S1** agents confirm these findings (results not shown).

Game nature does not influence the fact that in the long run cooperative agents aggregate. However, aggregation time depends on game nature as well as on agent strategies. When we compare figure 3(a) with figure 4(b), both have the same number of cooperative strategies, but aggregation time differs. The same line of reasoning can be established between figure 1(a) and figure 4(a), although the first only includes simulations with initial random edge weight and the second includes both identical and random initial edge weights. All other number of **S1** agents (see table 2) produced similar results (not shown).

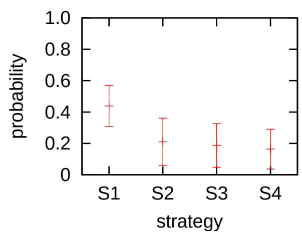
In the Give-Take with $b_g = 0$ there can be three groups of cooperative strategies. However, whenever a game is played, the number of iterations may vary, and it may not be a multiple of the time each strategy keeps the resource before giving it. This means that in a game between strategies **S3**, one of them may get a payoff lower than u_P and thus decrease the probability of selecting its peer. In figure 2, taken from a simulation with 2 agents per strategy, we show the edge weight graph with weights lower than 0.15 omitted. It can be seen that **S3** agents prefer **S2** and **S1** agents.



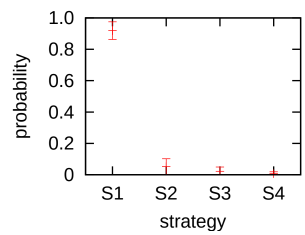
(a) Initial random edge weights and 2 agents with **S1** strategy.



(b) Initial identical edge weights and 2 agents with **S1** strategy.



(c) Initial random edge weights and 2 agents with **S1** strategy. Edge weights taken at $10,000^{\text{th}}$ round.



(d) Simulations with 16 **S1** strategies.

Fig. 1. Results of Give-Take with $b_g = 0$. Vertical axis represents the probability of strategy **S1** choosing a strategy in the horizontal axis. In cases 1(a) to 1(c) the results are averages of all possible combinations of strategies **S2** to **S4** (see table 2). In case 1(d) we also varied the initial edge weights.

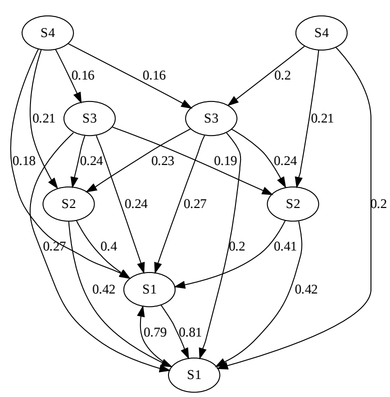
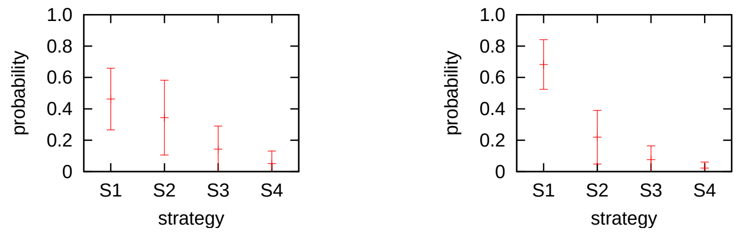
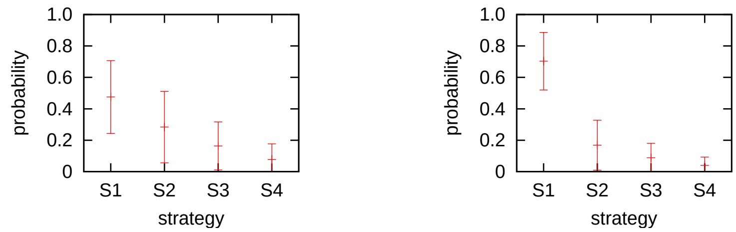


Fig. 2. Results of Give-Take with $b_g = 0$. Probability graph in the last round of a single simulation run with 2 agents per strategy type. Edge weights lower than 0.15 were omitted.



(a) 4 agents with strategy **S1**. (b) 8 agents with strategy **S1**.

Fig. 3. Give-Take with $b_g = 0.5$. Results are averages of all combinations of parameters not fixed (see table 2).



(a) 2 agents with strategy **S1**. (b) 4 agents with strategy **S1**.

Fig. 4. IPD results - averages of all combinations of parameters not fixed (see table 2).

Results show that the number of cooperative strategies influences the aggregation time. The higher is their number the shorter is aggregation time. This can be observed in figure 1(d), which refers to Give-Take with $b_g = 0$ and 16 agents with strategy **S1**, that has the highest probability of this strategy selecting a peer. Comparing figures 3(a) and 3(b), which refer to Give-Take with $b_g = 0.5$ and have, respectively, 4 and 8 agents with strategy **S1**, we observe that the second figure has the highest probability. If we increase the number of rounds, the probability of cooperative strategies to select themselves increases, as can be seen by comparing figure 1(a) taken at the 1000^{th} round and figure 1(c) taken at the 10000^{th} round.

Deterministic and stochastic strategies do not impinge on aggregation time. Instead, strategies' cooperative nature has an influence on it. Agents with cooperative strategies that are part of a Pareto Optimal profile will only select themselves.

6 Conclusions and Future Work

We have proposed and analysed theoretically a model for partner selection in a Multi-Agent System interacting through a game. We have complemented this analysis with an experimental simulations. The results confirm the theoretical analysis. In particular, game nature or initial edge weights do not influence the final aggregation result, although the former affects aggregation time while the latter does not. In the long run, cooperative agents will only select themselves as partners of interaction. The number of cooperative agents strongly influences their aggregation time. The more they are, the faster they aggregate.

This model requires a memory size that grows linearly with the number of partners. It also requires that agents are uniquely identifiable. The agent must know in advance the population size.

Despite some limitations, the edge weight update policy does not suffer from the problem of a period of learning followed by a period of reaping the benefits of education. If an agent does not get a good payoff from the current players, it will raise the edge weights of all the other agents, giving them a chance of being selected. However, further confirmation of this property should be tested with simulations where new agents are periodically introduced in the population, and the games and agents that select them should be monitored.

If agents select their partners, they can also refuse to play with certain agents. This could lead to situations where an agent cannot play because all its neighbours refuse to play. However, the possibility of refusing brings the problem of collecting experience back. During the period in which an agent is testing its neighbours, it may be explored by non-cooperators. In addition, after an agent closes its learning window, new agents will never play with it, unless we allow the reopening of the learning window.

Agent strategies are fixed. We should analyse the effects of adaptation, and apply the edge weight update policy to a framework of evolutionary games. An agent that cannot find a suitable partner can overcome this if it changes its strategy.

Acknowledgements

This work is partially supported by FCT/MCTES grant SFRH/BD/37650/2007.

References

1. James Andreoni, Paul M. Brown, and Lise Vesterlund. What makes an allocation fair? Some experimental evidence. *Games and Economic Behaviour*, 40:1–24, 2002.
2. Pat Barclay. Reputational benefits for altruistic punishment. *Evolution of Human Behaviour*, 27:325–344, 2006.
3. R Boyd and P J Richerson. Punishment allows the evolution of cooperation (or anything else) in sizable groups. *Ethology and Sociobiology*, 13:171–195, 1992.
4. Jordi Brandts and Neus Figueras. An exploration of reputation formation in experimental games. *Journal of Economic Behavior & Organization*, 50:89–115, 2003.
5. David V. Budescu, Amnon Rapoport, and Ramzi Suleiman. Common pool resource dilemmas under uncertainty: Qualitative tests of equilibrium solutions. *Games and Economic Behaviour*, 10(1):171–201, 1995.

6. Terence C. Burnham. Engineering altruism: a theoretical and experimental investigation of anonymity and gift giving. *Journal of Economic Behavior & Organization*, 50(1):133–144, 2003.
7. F.A.C.C. Chalub, F.C. Santos, and J.M. Pacheco. The evolution of norms. *Journal of Theoretical Biology*, 241(2):233–240, 2006.
8. David J. Cooper and Carol Kraker Stockman. Fairness and learning: an experimental examination. *Games and Economic Behaviour*, 41(1):26–45, 2002.
9. Chrysanthos Dellarocas. Goodwill hunting: An economically efficient online feedback mechanism in environments with variable product quality. In Rino Falcone and Larry Korba, editors, *5th Workshop on Deception, Fraud and Trust in Agent Societies*, pages 26–40, 2002.
10. Drew Fudenberg and Jean Tirole. *Game Theory*. MIT Press, 1991.
11. Herbert Gintis. *Game Theory Evolving - A problem-centered introduction to modeling strategic interaction*. Princeton University Press, 2000.
12. Christoph Hauert. Spatial effects in social dilemmas. *Journal of Theoretical Biology*, 2006.
13. Joseph Henrich and Robert Boyd. Why people punish defectors: Weak conformist transmission can stabilize costly enforcement of norms in cooperative dilemmas. *Journal of Theoretical Biology*, 208:79–89, 2001.
14. Leslie Luthi, Mario Giacobini, and Marco Tomassini. Synchronous and asynchronous network evolution in a population of stubborn prisoners. In *CIG*. IEEE, 2005.
15. Pedro Mariano and Luís Correia. A resource sharing model to study social behaviours. In Fernando Moura Pires and Salvador Abreu, editors, *Progress in Artificial Intelligence - 11th Portuguese Conference on Artificial Intelligence, EPIA 2003*, pages 84–88. Springer-Verlag, 2003.
16. John Maynard Smith. *Evolution and the Theory of Games*. Cambridge University Press, 1982.
17. cern.jet.random.engine.mersennetwister class part of colt library. http://acs.lbl.gov/ hoschek/colt/. last checked March 2009.
18. Sarvapali D. Ramchurn, Dong Huynh, and Nicholas R. Jennings. Trust in multi-agent systems. *The Knowledge Engineering Review*, 19(1):1–25, 2004.
19. Jordi Sabater and Carles Sierra. Review on computational trust and reputation models. *Artif. Intell. Rev.*, 24(1):33–60, 2005.
20. Francisco C. Santos, Marta D. Santos, and Jorge M. Pacheco. Social diversity promotes the emergence of cooperation in public goods games. *Nature*, 454:213–216, July 2008.
21. Frans van Dijk, Joep Sonnemans, and Frans van Winden. Social ties in a public good experiment. *Journal of Public Economics*, 85:275–299, 2002.
22. Bin Yu and Munindar P. Singh. Detecting deception in reputation management. In *AAMAS'03*. ACM, 2003.