Cooperation through Reciprocity in Multiagent Systems: An Evolutionary Analysis

Christian Hütter and Klemens Böhm
Institute for Program Structures and Data Organization
Karlsruhe Institute of Technology, Germany
[christian.huetter, klemens.boehm]@kit.edu

ABSTRACT

The Service Game is a model for reciprocity in multiagent systems. Here, agents interact repeatedly by requesting and providing services. In contrast to existing models where players are matched randomly, players of the Service Game may choose with whom they play. The rationale behind *provider selection* is to choose a provider that is likely to perform a task as desired. We develop a formal model for provider selection in the Service Game. An evolutionary process based on a genetic algorithm allows us to incorporate notions of bounded rationality, learning, and adaptation into the analysis of the game. We conduct a series of experiments to study the evolution of strategies and the emergence of cooperation. We show that cooperation is more expensive with provider selection than with random matching. Further, populations consisting of discriminators and defectors form a bistable community.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Economics, Experimentation

Keywords

Agent cooperation, Formal model, Social simulation, Evolution

1. INTRODUCTION

In settings without payments between individuals, the social mechanism of *reciprocity* is the basis for cooperation [16]. Examples for such settings are peer-to-peer systems and social search. With direct reciprocity, an individual A rewards helpful acts or punishes uncooperative acts of another individual B. Think of tit-for-tat, where A reciprocates B's previous action. With indirect reciprocity in turn, an action is rewarded or punished by a third individual C, not involved in the original interaction. Such strategies typically rely on reputation and status [2]. While cooperation through direct reciprocity can only emerge in repeated interactions between two individuals, it can emerge through indirect reciprocity in one-shot interactions as well.

Reciprocity has been a prominent research topic, both in theoretical [3, 16] and in experimental work [8, 10, 12]. In existing models for reciprocity, players are either matched randomly (for indirect reciprocity), or the same pairs of individuals interact repeatedly (for direct reciprocity). In realistic settings, however, individuals can choose whom to interact with. In the following, we will call this choice *provider selection*. Given the choice of interaction, both direct and indirect reciprocity can emerge. If individual A has done individual B a favor, A can redeem this favor either directly by interacting with B again or indirectly by interacting with another individual C. Thus, comprehensive models should cover both forms of reciprocity.

In this paper, we study the following research question: How efficient is reciprocity under provider selection? Answering this question is difficult for several reasons. First, we expect provider selection to change the game substantially, compared to random matching. This is because cooperative providers, which are typically preferred by requesters, are likely to have high workloads. Second, provider selection has not yet been analyzed in existing studies of reciprocity. As we will explain, a formal model required for the analysis cannot be derived directly from existing models where players are matched randomly. Third, an analytical solution of evolutionary games is hard if discriminating strategies are present [2]. Discriminating strategies differentiate between players according to certain attributes (e.g., their cooperativeness), as opposed to strategies that treat all players as equal. Simulations have shown that discriminating strategies can lead to cooperative populations [12].

We meet these challenges by proposing the *Service Game*, a formal model for reciprocity under provider selection. Here, players interact repeatedly in the roles of requester and provider. The *requester* sends a request to the *provider*, who decides on processing the request. Players have a benefit if other players process their requests, whereas processing requests for others incurs cost. In contrast to existing models, ours allows for both direct and indirect reciprocity by letting players choose with whom they play. Note that, in our model, all nodes are fully connected. Hence, network formation is not an issue here.

Traditional game theory assumes that all players are fully rational and homogeneous. While these assumptions are crucial for the mathematical tractability of the models, more recent work has dropped them [14]. In evolutionary game

Cite as: Cooperation through Reciprocity in Multiagent Systems: An Evolutionary Analysis, Christian Hütter and Klemens Böhm, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 241-248.

theory, the role of perfect rationality is taken over by a learning process in a heterogeneous society of agents. An evolutionary process modeled by a genetic algorithm updates the behavior of the agents. Genetic algorithms have frequently been used in experimental work on cooperation to describe social learning [1, 10, 14].

We have designed and carried out a series of experiments to study the evolution of strategies and the emergence of cooperation under provider selection. It shows that cooperation is more expensive with provider selection than with random matching. We deem this a fundamental observation because it means that direct and indirect reciprocity should not be analyzed separately. Further, populations consisting of discriminators and defectors form a bistable community. That is, provider selection preserves the evolutionary stable states predicted by theory [3].

2. RELATED WORK

In this section we discuss related work. After examining existing models for reciprocity, we review studies of the evolution of cooperation. Finally, we delimit our work from existing studies of provider selection.

While direct reciprocity has been a popular research topic, more recent work on reciprocity has focused on indirect reciprocity. One popular model to analyze cooperation through indirect reciprocity is the *helping game* [12]. There, pairs of a donor and a recipient are formed randomly. The recipient asks the donor he is matched with for costly 'help'. Nowak and Sigmund have shown that cooperation can evolve if information about the past behavior of the players is available. Each player is assigned a public *image score* which increases through cooperation and decreases through defection. Strategies based on image scoring have proven to be evolutionary stable. A strategy population is called *evolutionary stable* if the evolutionary process rejects the invasion of the population by mutant strategies [9].

Gal and Pfeffer [5] showed that reciprocity has significant implications for the behavior of agents. When they interact with humans over time, agents need to learn social factors that affect people's play. Brandt et al. [3] analyzed direct and indirect reciprocity theoretically using replicator dynamics. While replicator dynamics are common to study the evolutionary dynamics in games, they lack the notions of learning and adaption [14]. Researchers often resort to numerical simulations using genetic algorithms [8, 10, 12], which explicitly model the individual agents. We will make use of this methodology as well by carrying out an evolutionary analysis of cooperation in multiagent systems. Research has proposed various approaches on cooperation among deceptive agents that misreport the behavior of other agents. Sen [15] has proposed a probabilistic strategy based on reciprocity which is able to resist this kind of deception. While we do not consider deceptive agents, the approach by Sen might be applied to provider selection as well.

There also are models for the evolution of cooperation not based on reciprocity. One such model uses so-called 'tags' [6] to help agents in identifying the groups they belong to. It has been shown that provider selection based on tags produces stable cooperation. While tags correspond to the origin of the agents, provider selection in our setting is based on behavior. Fullam et al. [4] proposed a testbed for trust and reputation mechanisms. Agents may choose other agents from which they seek help, and the agents asked for help decide whether to cooperate or not. Whereas Fullam et al. compare a fixed set of strategies, we investigate the evolution of strategies through a genetic algorithm. Provider selection has also been studied in the area of peer-to-peer systems [13]. However, existing studies we are aware of focus on performance figures such as the ratio of successful interactions. In the following, we develop a full-fledged economic model for reciprocity in multiagent systems.

3. THE SERVICE GAME

The Service Game is a repeated game where players may send requests to other players and decide whose requests they process. That is, each player actually plays multiple games in one round—one as requester and zero or more as provider, depending on the number of requests he receives. The *fee* for sending a request to another player is f, with $f > 0$. The *cost* of processing the request of another player is c, with $c > 0$. The *benefit* of a player if another player processes his request is b, with $b > 0$. Fee, cost and benefit are the same for all players. As usual in helping games, we assume that $b > c > f > 0$ holds. The Service Game models a homogeneous system in which all players send and process requests at the same rate. To simplify matters, we assume that there is only one kind of service which every player can provide. Providers may neither pass on requests to other players, nor postpone the decision on processing a request. All players make their cooperation decisions simultaneously.

We implement the Service Game as a multiagent system, i.e., agents act as players. Each agent pursues two so called 'policies' which define the actions it performs. A *placing policy* specifies the set of players which the agent sends requests to. An *accepting policy* specifies the set of players whose requests the agent is willing to process. As described in Algorithm 1, every round t of the game consists of two steps: provider selection and service decisions. First, each agent i evaluates its placing policy p_i to select a provider j. If the placing policy is empty, the agent does not send a request. If it contains several players, a provider is selected at random from this set. Second, each agent j decides on processing incoming requests by evaluating its accepting policy a_j. If the accepting policy contains requester i, the agent processes the task, otherwise it rejects the request.

Algorithm 1 The Service Game

```
for all rounds $t$ do
    for all players $i \in I$ do {provider selection}
        provider $j \leftarrow$ evaluate($p_i$)
        queue $Q_j \leftarrow Q_j \cup \{i\}$
    for all players $j \in I$ do {service decisions}
        for all requesters $i \in Q_j$ do
            if $i \in$ evaluate($a_j$) then
                process the task
            else
                reject the request
```

3.1 Formal Model

The Service Game is characterized by the set of players $I = \{1, \ldots, n\}$, the policy sets A and P, and the payoff or net return n_k^t of player $k \in I$ in round t of the game. A *placing policy* $p_k \subseteq I \setminus \{k\}$ specifies the set of players which player k sends a request to. E.g., the placing policy $p_k =$

$\{l \in I \setminus \{k\} \,|\, r_l \geq 0.8\}$ means that player k sends requests only to players with a cooperativeness of 80% or more. An *accepting policy* $a_k \subseteq I \setminus \{k\}$ specifies the set of players whose requests player k is willing to process. E.g., the accepting policy $a_k = \{l \in I \setminus \{k\} \,|\, r_l \geq 0.5\}$ means that player k processes requests only for players with a cooperativeness of 50% or more. We denote the set of placing policies as $P = \{p_1, \ldots, p_n\}$ and the set of accepting policies as $A = \{a_1, \ldots, a_n\}$. We say that player k *meets* placing policy p resp. accepting policy a if $k \in p$ resp. $k \in a$ holds.

We introduce a *discount rate* $\delta \leq 1$ to avoid infinite payoffs. On the one hand, it takes into account the probability that another round occurs after the current round. On the other hand, it represents the economic fact that the current value of a future income is less than its nominal value. The *present value* N_k^t of the payoffs in round t is the discounted sum of all previous rounds.

$$N_k^t = \delta \cdot N_k^{t-1} + n_k^t \tag{1}$$

3.2 Cooperativeness of the Players

Number of requests received

We first count how many requests player k is expected to receive in round t. Let $|p_l^t|$ be the number of players who meet the placing policy p_l^t of some player l in round t. The reciprocal value $1/|p_l^t|$ then is the probability that a random player who meets the placing policy p_l^t receives a request from player l. We denote this probability as the expected number of requests exp_l^t such a player receives from player l in round t.

$$exp_l^t = \begin{cases} \frac{1}{|p_l^t|} & \text{if } |p_l^t| > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

We now sum up exp_l^t over all players l from which player k might receive requests because he meets their placing policies p_l^t. This is the expected number of requests rec_k^t player k receives in round t.

$$rec_k^t = \sum\nolimits_{l \in \{I \setminus \{k\} \,|\, k \in p_l^t\}} exp_l^t \tag{3}$$

Number of requests processed

Next, we count how many requests player k is expected to process in round t. He accepts requests from player l if and only if this player meets his accepting policy a_k^t. We refer to the respective indicator function as $acc_{k,l}^t$.

$$acc_{k,l}^t = \begin{cases} 1 & \text{if } l \in a_k^t, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

If we multiply $acc_{k,l}^t$ by exp_l^t, we get the expected number of requests which player k processes for player l. We now sum up this product over all players l from which player k receives requests because he meets their placing policies p_l^t. This is the expected number of requests $proc_k^t$ player k processes.

$$proc_k^t = \sum\nolimits_{l \in \{I \setminus \{k\} \,|\, k \in p_l^t\}} acc_{k,l}^t \cdot exp_l^t \tag{5}$$

Cooperativeness

We define the *cooperativeness* of a player as the share of incoming requests he processes. E.g., a player with cooperativeness $r = 0.5$ processes half of the requests. The cooperativeness r_k^t of player k in round t is the quotient of the number of requests processed $proc_k^t$ and the number of requests received rec_k^t.

$$r_k^t = \begin{cases} \frac{proc_k^t}{rec_k^t} & \text{if } rec_k^t > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

The *mean cooperativeness* R_k^t of player k in round t is the average over all previous rounds. Since players may change their behavior during the game, recent interactions are more important than old ones. We use an exponential moving average with smoothing factor $\alpha \leq 1$ to assign more weight to recent interactions.

$$R_k^t = \alpha \cdot r_k^t + (1 - \alpha) \cdot R_k^{t-1} \tag{7}$$

3.3 Payoff of the Players

Benefit of a successful request

To compute the benefit b_k^t of player k, we first determine whether this player sends out a request in round t. Player k sends requests to all players l which meet his placing policy p_k^t. We refer to the respective indicator function as $sent_k^t$.

$$sent_k^t = \begin{cases} 1 & \text{if } |p_k^t| > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

Next, we determine whether this request is successful or not. Player l accepts requests from player k ($acc_{l,k}^t = 1$) if and only if player k meets his accepting policy a_l^t. If we multiply $acc_{l,k}^t$ by exp_k^t, we get the expected number of requests which player l processes for player k. We now sum up this product over all players l to which player k sends requests because they meet his placing policy p_k^t.

$$succ_k^t = \sum\nolimits_{l \in p_k^t} acc_{l,k}^t \cdot exp_k^t \tag{9}$$

Note that the expected number of successful requests $succ_k^t$ is at most 1. The benefit b_k^t of player k is the product of b and the number of successful requests sent out in round t.

$$b_k^t = b \cdot succ_k^t \tag{10}$$

Cost of processing requests

The cost c_k^t depends on the number of requests player k processes in round t. While player k receives requests from all players l whose placing policies p_l^t he meets, he only processes requests from players who meet his accepting policy a_k^t. This is exactly the definition of $proc_k^t$ in Equation (5). In addition, player k has to pay fee f if he sends out a request in round t. We use the indicator function $sent_k^t$ from Equation (8).

$$c_k^t = c \cdot proc_k^t + f \cdot sent_k^t \tag{11}$$

Payoff

The payoff n_k^t of each player k in round t is the difference between the benefit b_k^t of a successful request and the cost c_k^t of sending and processing requests.

$$n_k^t = b \cdot succ_k^t - c \cdot proc_k^t - f \cdot sent_k^t \tag{12}$$

In every round t of the game, each player k would ideally choose the combination of placing policy p_k^t and accepting policy a_k^t which maximizes his payoff. We denote the optimal payoff of player k in round t as $\hat{n}_k^t$:

$$\hat{n}_k^t = \underset{p_k^t \subseteq I \setminus \{k\},\, a_k^t \subseteq I \setminus \{k\}}{\text{maximize}} \quad n_k^t \tag{13}$$

It is hard to determine the optimal payoff analytically if discriminating strategies are present [2]. Often, pairs of discriminating strategies perform equally well against each other, so that their frequencies drift randomly. However, the success of other strategies depends on the frequencies of the discriminating strategies. To remove this restriction, we will resort to numerical simulations.

4. EVOLUTIONARY ANALYSIS

In evolutionary games, players are not assumed to be rational or able to think ahead. Strategies are simple behavioral programs which specify the actions of each player. In this section, we will first explain the strategies used in our analysis and then describe the evolutionary process in detail.

4.1 Strategies

In a previous experiment of ours [7], subjects have formulated plaintext policies which we then used to program agents playing the Service Game. A *cutoff strategy* based on the cooperativeness of the players was the most successful strategy for provider selection. Hence, in this paper, we will analyze the evolutionary stability of the cutoff strategy for provider selection. In each round t of the game, player i sends a request to player j if the mean cooperativeness R_j^{t-1} of player j exceeds a threshold value ρ_i:

$$p_{i,j}^t = \begin{cases} 1 & \text{if } R_j^{t-1} \geq \rho_i, \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

We will eventually compare the results of our study with previous studies on reciprocity [2]. In these studies, the three basic strategies 'always cooperate', 'always defect', and 'tit-for-tat' have prevailed. Thus, in this paper, we use these strategies as well for service decisions. Needless to say, there are many other strategies conceivable for both provider selection and service decisions. Nevertheless, the strategies we use capture the most important aspects of cooperation [3].

We implement the strategies for service decisions as *stochastic reactive strategies* [11]. They are described by two parameters (p, q), which are the probabilities to cooperate after a cooperation resp. a defection by the co-player in the previous round. Always cooperate is given by $(1, 1)$, always defect by $(0, 0)$, and tit-for-tat by $(1, 0)$. In each round t, the probability that player i accepts a request from player j is the expected value of the stochastic reactive strategy. Recall that $a_{j,i}^{t-1}$ denotes the service decision of player j regarding player i in round $t - 1$.

$$a_{i,j}^t = p \cdot a_{j,i}^{t-1} + q \cdot (1 - a_{j,i}^{t-1}) \tag{15}$$

A strategy can be represented as a 'chromosome' describing the action of a player in each different context of the game. We encode the strategy of each player i by a binary string of 24 bits. The first 8 bits represent the threshold value ρ_i of the cutoff strategy for provider selection. The second and third 8 bits represent the probabilities p_i and q_i of the stochastic reactive strategy for service decisions. We use a Gray coding of the binary strings to avoid the representational bias in binary encoding.

4.2 Evolutionary Process

We update the strategies through an evolutionary process modeled by a genetic algorithm (GA). GAs mimic a population of strategies acting in a well-defined environment which evaluates the performance of each strategy. New populations are formed by selecting the better performing strategies and modifying them through genetic operators. The GA then subjects the resulting offspring to competition with other strategies in the population. Successful strategies are allowed to reproduce, while unsuccessful strategies become extinct. GAs have frequently been used in economics to characterize a form of social learning [1, 10, 14]. Selection can be interpreted as learning by imitation, recombination as learning by communication, and mutation as learning by experiment. The combination of these three operators results in a very powerful optimization algorithm.

Algorithm 2 The genetic algorithm

```
Create initial population $m_0$
Evaluate $m_0$
for round $t = 1$ to $t_{max}$ do
    Rank $m_{t-1}$
    Select from $m_{t-1}$ into $m_t$
    Recombine $m_t$
    Mutate $m_t$
    Evaluate $m_t$
```

The GA used in this paper is shown in Algorithm 2. First, an initial strategy population m_0 is created randomly. Each strategy is then tested against the environment (composed of the other strategies) and receives a performance score (the payoff). Given the performance scores, a fitness value is assigned to each strategy. We use a rank-based fitness function with a selective pressure of 2 and linear ranking, giving the most fit strategy a fitness value of 2 and the least fit strategy a fitness value of 0. Strategies from the parent population m_{t-1} are selected for reproduction using a stochastic universal sampling routine. The strategies m_t selected for reproduction are recombined using a single-point crossover function with probability .7. Having produced the offspring, mutation may now be applied with probability $.7/L_{ind}$, where $L_{ind} = 24$ is the length of a chromosome. Finally, the offspring m_t is evaluated and the new performance scores are calculated. The GA terminates when the maximum number of rounds t_{max} is reached. We verify that the population has converged by analyzing its variance.

5. EXPERIMENTS

To analyze the evolution of strategies and the emergence of cooperation under provider selection, we formulate the following research questions:

1. How does provider selection change policy-based helping scenarios?
2. Which states of the game are evolutionary stable, and how efficient are the resulting equilibria?
3. How do the parameters of the game influence stability and efficiency?

Regarding the first question, we expect provider selection to change the game substantially. This is because cooperative providers are likely to receive many requests, making cooperation expensive. With the second question we investigate whether the equilibria identified in previous studies

Table 1: Parameter values used in the experiments

Experiments		f	c	b	δ	α
E1	Random matching	0	2	20	.9	.5
	Provider selection	0	2	20	.9	.5
E2	No fee	*0*	2	20	.9	.5
	Low cost	1	2	20	.9	.5
	High cost	5	*10*	20	.9	.5
E3	Low discount	1	2	20	*.1*	.5
	Medium discount	1	2	20	.5	.5
	High discount	1	2	20	*.9*	.5
E4	Low smoothing	1	2	20	.5	*.1*
	Medium smoothing	1	2	20	.5	.5
	High smoothing	1	2	20	.5	*.9*

still hold for the Service Game. We expect the equilibria to be less efficient because, according to a previous study of ours [7], players are less cooperative in a game with provider selection than in a game with random matching. The third question addresses the parameters of the Service Game (f, c, b, δ, α). According to previous studies on reciprocity [2], we expect the cost-to-benefit ratio c/b to have a significant influence on the efficiency of the game. While a high discount factor δ causes a low rate of inflation, a low smoothing factor α indicates a long memory of interactions. We expect both to increase the efficiency.

5.1 Experimental Design

To answer the research questions, we have designed and conducted a series of experiments. In these experiments, we explore all parameters of the game separately (i.e., we vary one parameter at a time). In Experiment *E1*, we compare random matching with provider selection, to address the first research question. We tackle the second question by analyzing the equilibria of the game and their efficiency. Regarding the third question, we compare different ratios of fee/cost and cost/benefit in Experiment *E2*, different discount rates in Experiment *E3*, and different smoothing factors in Experiment *E4*. Table 1 serves as a summary. Finally, we identify lessons learned from our experiments that are of general interest.

Each simulation consists of a population of 100 agents, which played the Service Game for 100 rounds. In each round of the game, each player plays against every other player specified by his placing policy. The payoff then is the expected value of the play, i.e., the average over the individual games. Payoffs were calculated using the equations in Section 3.3. Under each of the conditions, 100 repetitions were conducted to allow for stochastic variations. The choices of simulation parameters (e.g., population size) and algorithmic components (e.g., selection function of the GA) were guided by considerations of robustness and computational constraints. Note that genetic algorithms are extremely robust to actual parametric and algorithmic choices. All the results reported in the next section have been confirmed using a variety of different simulation parameters and algorithmic components.

5.2 Methodology

The frequencies of the three strategies for service decisions (cooperate, defect, and tit-for-tat) are given by x, y, and z with $x + y + z = 1$. Thus, the three strategies form a *strategy simplex* which describes the composition of the population. The strategy simplex can be imagined as a 2D triangle in the 3D coordinate system (x, y, z). The three vertices $x = 1$, $y = 1$, and $z = 1$ of the strategy simplex describe 'pure' populations consisting solely of the strategies cooperate, defect, and tit-for-tat, respectively. The edges $x = 0$, $y = 0$, and $z = 0$ describe 'dual' populations consisting of the two strategies specified by its vertices. Finally, the points (x, y, z) within the simplex describe 'mixed' strategy populations.

By definition, the number of points in the strategy simplex is infinite. Thus, we have to create a random sample of the strategy simplex. In theory, it is possible that the random sample misses important aspects of the game. In the following experiments, we have chosen a large sample size of 500 points to minimize the possibility of error. To avoid clutter in the figure, we will depict only 100 points. We run the GA for each point of the sample and analyze how the population evolves. The strategies are updated through the evolutionary process described in Section 4. In the following analyses, we will average over the resulting strategies in each population. The key figures to quantify the efficiency are the mean cooperativeness R and the total payoff N of the players. Both figures are strongly correlated because requesters can only make profit if the providers cooperate. We focus on the cooperativeness because it is, by definition, normalized to the interval $[0, 1]$ and thus allows for a direct comparison.

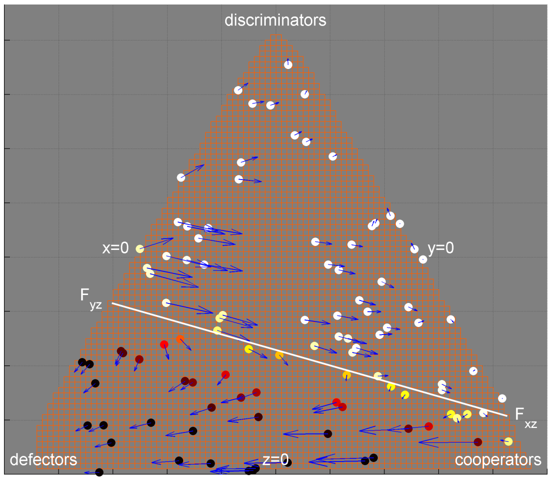
6. RESULTS

In this section, we first examine a system with random matching to replicate the effects of existing studies [2] on reciprocity. It serves as baseline for the following analysis of provider selection. Our experiments indicate that there are significant differences between random matching and provider selection.

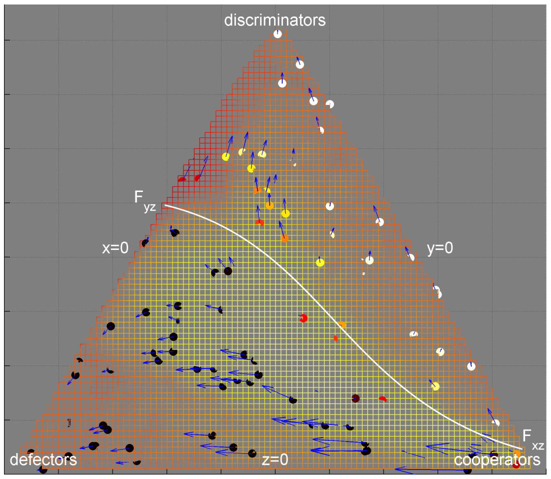
6.1 Random Matching

In experiment *E1*, we have analyzed how the matching of the players affects the evolution of the strategies and the efficiency of the game. Figure 1a shows the resulting simplex for random matching. To visualize the results, we have projected the strategy simplex onto a 2D coordinate system. The three vertices $x = 1$, $y = 1$, and $z = 1$ were mapped to the coordinates $(1, 0)$, $(0, 0)$, and $(1/2, \sqrt{3}/2)$, respectively. Recall that each point of the simplex represents one population of strategies. The coordinates specify the frequencies of the strategies, i.e., the initial composition of the population. To visualize the efficiency of each population, we depict the average cooperativeness as the color of the corresponding point. Light colors (white to yellow) represent cooperative populations, dark colors (red to black) uncooperative ones.

During the evolutionary process, the composition of the population changes because successful strategies prevail. The arrows in each point visualize the direction which the population shifts to. E.g., the points in the center of the simplex shift to the edge $y = 0$. A strategy population is in an evolutionary stable state if its composition does not change during the evolutionary process [9]. A fixed point keeps its position, i.e., the population is evolutionary stable. A closer look at the data reveals that the vertex $y = 1$ consisting solely of defectors is evolutionary stable. Furthermore, all points on the



(a) Random matching



(b) Provider selection

Figure 1: Strategy simplices for random matching and for provider selection.

edge $y = 0$ are evolutionary stable. I.e., any mixture of cooperators and discriminators are in equilibrium. All points on the edge $z = 0$ are unstable. There is a line of unstable points connecting a fixed point F_{yz} on the edge $x = 0$ with another fixed point F_{xz} on the edge $y = 0$. We call this line the *boundary line*. While the states between the defectors vertex and the boundary line are inefficient (cooperativeness $R \approx 0$), the states between the line and the discriminators vertex are efficient ($R \approx 1$). These findings are consistent with theoretical results in [3].

6.2 Provider Selection

From now on, we focus on systems with provider selection. Figure 1b shows the resulting strategy simplex. Apparently, the system with random matching has a higher fraction of cooperative (white) states than the system with provider selection. In fact, the mean cooperativeness is .637 (±.431) for random matching and only .402 (±.435) for provider selection. A t-test confirms that the difference is significant for a confidence level of .01.

Observation 1. Games with provider selection are less cooperative than games with random matching.

This result is expected because, according to their placing policies, requesters prefer cooperative providers. Thus, cooperative players have high workloads, making cooperation expensive. In contrast, random matching balances requests uniformly between providers.

An analysis of the evolutionary stability shows that the system with provider selection has the same stable states as the system with random matching. Again, the vertex $y = 1$ consisting solely of defectors is evolutionary stable. Along the edge $y = 0$, any mixture of cooperators and discriminators is stable. Finally, a line of unstable states (the boundary line) connects a fixed point F_{yz} on the edge $x = 0$ with a fixed point F_{xz} on the edge $y = 0$.

Observation 2. Populations consisting of discriminators and defectors form a bistable community.

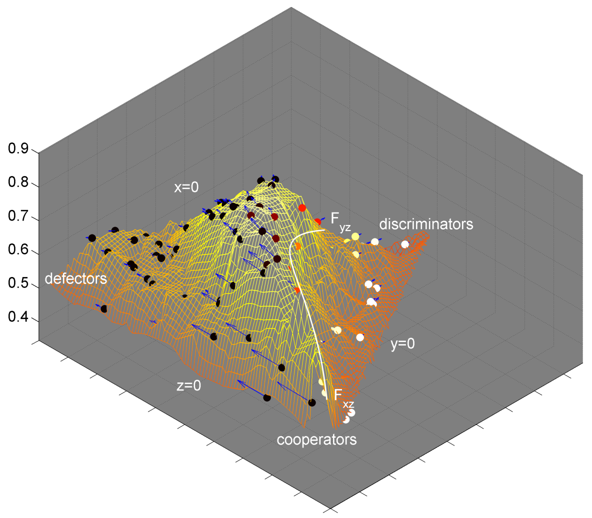
While the strategies for service decisions are defined by their coordinates in the strategy simplex, the cutoff strategy for provider selection is defined by its threshold values ρ (see Equation 14). For each population of strategies, we have computed the mean and the variance of the threshold values. We visualize the average threshold values as an additional dimension (the z-axis) of the strategy simplex. 'Higher' points represent greater threshold values and thus stricter cutoff strategies. Figure 2 shows the resulting strategy simplices.

Interestingly, the location of the boundary line strongly influences the threshold values of the cutoff strategies (height). The boundary line divides the simplex into two areas. In the area between the defectors vertex and the boundary line, the cutoff thresholds reach their maximum values, and the system is inefficient ($R \approx 0$). The threshold maximum can be visualized as a 'range' of strict cutoff strategies. On the other side, between the boundary line and the discriminators vertex, the threshold values are much lower. Visually, the boundary line forms the 'rim' of the cutoff range. If we descend the cutoff range towards the discriminators vertex, the system becomes more efficient. In the 'valley' along the edge $y = 0$, the system is highly efficient ($R \approx 1$).

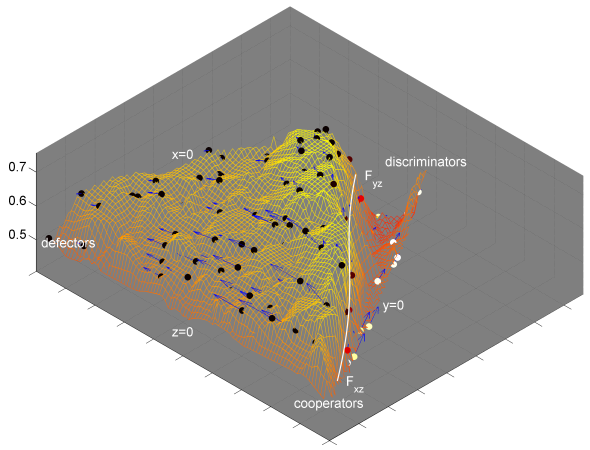
6.2.1 Benefit, Cost, and Fee

In experiment *E2*, we have investigated how the parameters benefit, cost, and fee affect the efficiency. First, we have conducted an analysis of variance (ANOVA) to test whether the means of the three treatments (no fee, low cost, high cost) are all equal or not. The test indicates that at least one sample mean is different from the other two ($F = 60.89$) with a significance level of .01. Next, we have analyzed the ratio f/c of fee to cost by comparing the no-fee and the low-cost treatment. The statistics do not show any significant difference. Thus, we leave aside the no-fee treatment in the remaining analysis.

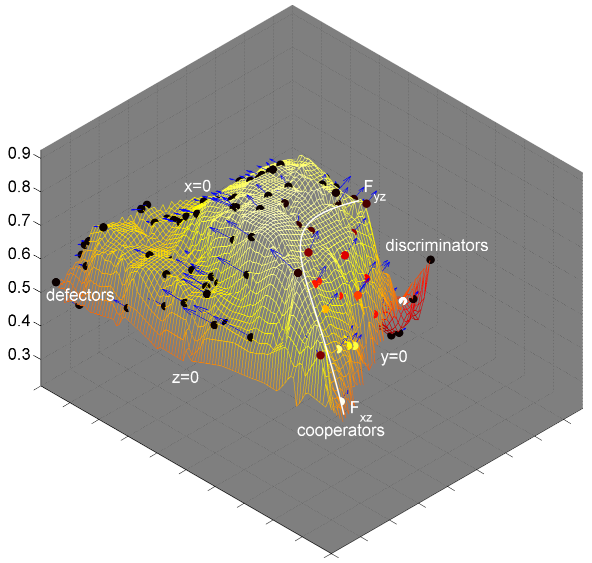
The ratio c/b of cost to benefit can be analyzed by comparing the low-cost and the high-cost treatment. Figures 2a–b show the resulting strategy simplices for both treatments. The mean cooperativeness is .37 (±.43) for the low-cost and only .148 (±.316) for the high-cost treatment. A t-test confirms that the difference is significant for a confidence level of .01. We observe that the ratio c/b also affects the location of the boundary line. As the cost-to-benefit ratio increases,



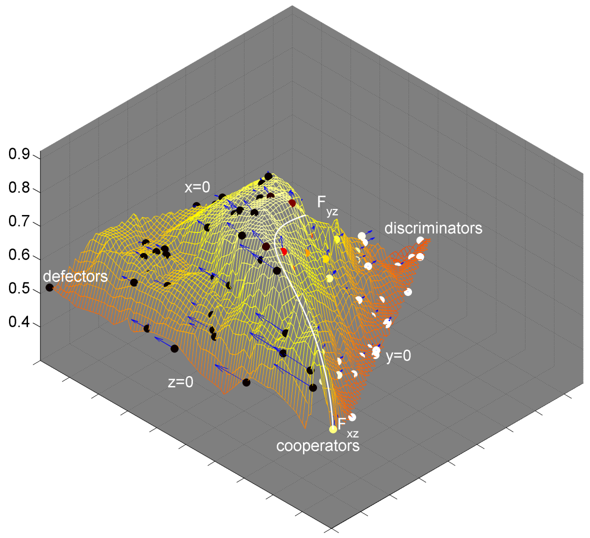
(a) Low cost/high discount: f=1, c=2, b=20, δ=.9



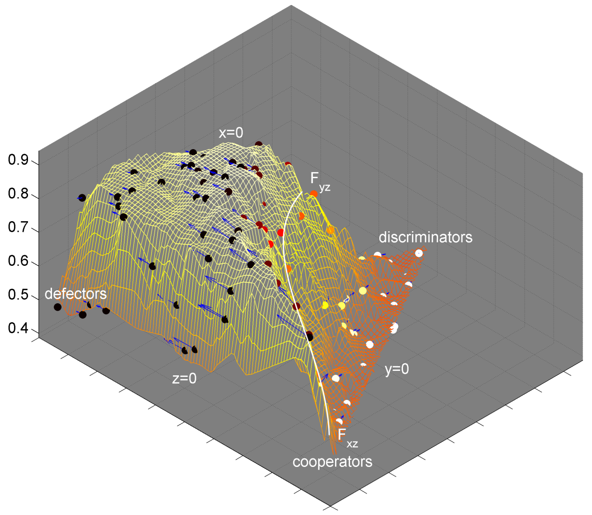
(b) High cost: f=5, c=10, b=20



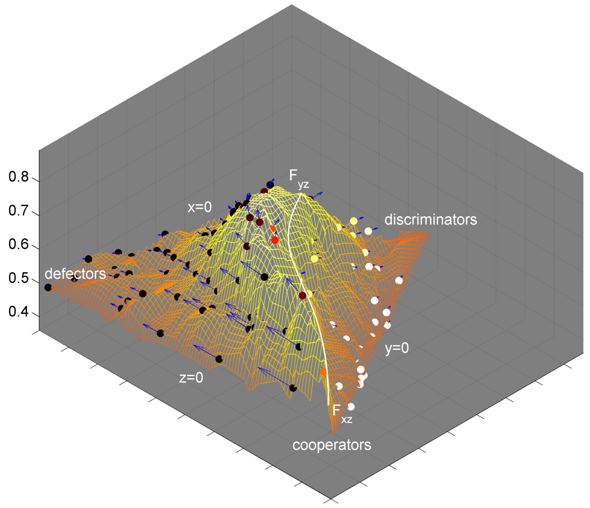
(c) Low discount rate: δ=.1, α=.5



(d) Medium discount/smoothing: δ=.5, α=.5



(e) Low smoothing factor: δ=.5, α=.1



(f) High smoothing factor: δ=.5, α=.9

Figure 2: Strategy simplices for different game parameters under provider selection.

the fixed point F_{yz} moves closer towards the discriminators vertex and thus the efficiency decreases.

Observation 3. The lower the ratio of cost to benefit, the more efficient the system.

An important lesson learned is that the benefit of a successful request should be at least one order of magnitude higher than the cost of processing. A fee for sending requests does not affect the efficiency of the system.

6.2.2 Discount Rate and Smoothing Factor

In experiments *E3* and *E4*, we have investigated how discount rate and smoothing factor affect the efficiency. First, we have compared low, medium and high discount rates δ. A low discount rate indicates a high rate of inflation and thus a decline of the present value of the payoffs. For the results of low, medium, and high discount rates see Figures 2c, d, a. The mean cooperativeness is .092 ($\pm$.213) for the low discount and .370 ($\pm$.43) for the high discount treatment. A t-test confirms that the difference is significant for a confidence level of .01. Apparently, the discount rate affects the 'height' of the cutoff range. The lower the discount rate, the higher the cutoff range and thus the stricter the strategies.

Observation 4. A higher discount rate (i.e., a lower inflation) results in a more efficient system.

Finally, we compared low, medium, and high smoothing factors α. A higher factor assigns more weight to recent interactions and thus causes a shorter memory. For the results of low, medium, and high smoothing factors see Figures 2e, d, f. The mean cooperativeness is .422 ($\pm$.412) for the low smoothing and .362 ($\pm$.429) for the high smoothing treatment. A t-test confirms that the difference is significant for a confidence level of .01. Apparently, the smoothing factor affects the 'width' of the cutoff range. The lower the factor is, the wider the cutoff range.

Observation 5. A lower smoothing factor (i.e., a longer memory) increases the efficiency of the system.

Thus, a second lesson learned is that reputation systems for multiagent systems should consider the complete history of interactions.

7. CONCLUSIONS

In settings without payments between individuals, reciprocity is the basis for cooperation. Examples for such settings are peer-to-peer systems and social search. In existing models for reciprocity, individuals are either matched randomly, or the same pairs of individuals interact repeatedly. However, in realistic settings, individuals can choose whom to interact with. In this paper, we have investigated how efficient reciprocity is under provider selection. To do so, we have developed a formal model for reciprocity in multiagent systems. Strategies are updated through an evolutionary process based on a genetic algorithm. This lets us incorporate the notions of bounded rationality, learning, and adaptation into the analysis.

We have designed and carried out a series of experiments to study the evolution of strategies and the emergence of cooperation. Our results show that cooperation is more expensive in a system with provider selection than in a system with random matching. Thus, existing models for reciprocity overestimate the efficiency of real-world systems where both direct and indirect reciprocity may occur in combination. Further, populations consisting of discriminators and defectors form a bistable community.

8. ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 216287 (TAS^3 – Trusted Architecture for Securely Shared Services).

9. REFERENCES

[1] R. Axelrod. The evolution of strategies in the iterated prisoner's dilemma. In *Genetic Algorithms and Simulated Annealing*. Pitman, 1987.

[2] H. Brandt, H. Ohtsuki, Y. Iwasa, and K. Sigmund. A survey of indirect reciprocity. *Mathematics for Ecology and Environmental Sciences*, 2007.

[3] H. Brandt and K. Sigmund. The good, the bad and the discriminator–errors in direct and indirect reciprocity. *Journal of theoretical biology*, 239(2), 2006.

[4] K. K. Fullam and K. S. Barber. Learning trust strategies in reputation exchange networks. In *Proc. of the 5th international joint conference on Autonomous agents and multiagent systems*, 2006.

[5] Y. Gal and A. Pfeffer. Modeling reciprocal behavior in human bilateral negotiation. In *Proc. of the 22nd national conference on Artificial intelligence*, 2007.

[6] D. Hales and B. Edmonds. Evolving Social Rationality for MAS using "Tags". In *Proc. of the 2nd international joint conference on Autonomous agents and multiagent systems*, 2003.

[7] C. Hütter, J. Z. Yue, C. von der Weth, and K. Böhm. Strategic provider selection in a policy-based helping scenario. In *Proc. of the 12th IEEE Conference on Commerce and Enterprise Computing*, 2010.

[8] O. Leimar and P. Hammerstein. Evolution of cooperation through indirect reciprocity. *Biological sciences*, 268(1468), 2001.

[9] J. Maynard Smith. *Evolution and the Theory of Games*. Cambridge Univ. Press, 1982.

[10] J. Miller. The coevolution of automata in the repeated Prisoner's Dilemma. *Journal of Economic Behavior & Organization*, 29(1), 1996.

[11] M. Nowak and K. Sigmund. The evolution of stochastic strategies in the prisoner's dilemma. *Acta Applicandae Mathematicae*, 20(3), 1990.

[12] M. A. Nowak and K. Sigmund. Evolution of Indirect Reciprocity by Image Scoring. *Nature*, 393, 1998.

[13] T. Papaioannou and G. Stamoulis. Effective use of reputation in peer-to-peer environments. In *Proc. of the IEEE International Symposium on Cluster Computing and the Grid*, 2004.

[14] T. Riechmann. Genetic algorithm learning and evolutionary games. *Journal of Economic Dynamics and Control*, 25(6-7), 2001.

[15] S. Sen. Believing others: Pros and cons. *Artificial Intelligence*, 142(2), 2002.

[16] R. L. Trivers. The Evolution of Reciprocal Altruism. *The Quarterly Review of Biology*, 46(1), 1971.