Can You Do Me A Favor?

Keith Sullivan, Sean Luke, and Brian Hrolenok

Department of Computer Science, George Mason University
4400 University Drive, Fairfax VA 22030 USA
{ksulliv2, sean, bhroleno}@cs.gmu.edu

Abstract. Multiagent systems often require coordination among the agents to maximize system utility. Using the notion of favors, we propose a technique, *flexible reciprocal altruism*, which determines when one agent should grant a favor to another agent based on past interactions. The desired rate of altruism is controllable, and as a result the loss associated with granting unmatched favors is bounded and amortized over all past interactions. In flexible reciprocal altruism the desired acceptable loss is independent of the cost and value of the favors. Experiments show that our technique performs well with different cost/value tradeoffs, numbers of agents, and load.

1 Introduction

In a multi-agent system environment, the agents typically have limited resources (e.g., sensors, communications) that restrict the problems they can solve. If the agents coordinate, then the system can be made more effective. In particular, system performance generally improves if the agents can coordinate both locally and globally. Example applications include cooperative target observation [1], foraging [2], and peer-to-peer systems [3]. One useful approach to agent coordination is to consider the agents as providing *favors* to the other agents.

Most of the existing work on favors has naturally followed from some version of *reciprocal altruism:* one agent is willing to incur a cost *now* to provide a value to another agent in return for (hopefully) receiving value in the future from the other agent, who in turn will incur some cost. Existing work in reciprocal altruism in multiagent systems has focused to date on rational agents, trust, social laws, and reputation. We instead tackle the issue from the viewpoint of *amortized risk:* the degree to which an agent is willing to go out on a limb for another agent is restricted in such a way that if the second agent never reciprocates again, the first agent's loss, amortized over *all previous interactions* with the second agent, is bounded.

Why would an agent behave like this? We argue that in many multiagent systems tasks, a moderate number of agents are essentially working towards the common good, and have a reasonable expectation of similar behaviors from other agents. However, the agents do not yet know exactly the nature of the other agents. The other agents could be dishonest, of course, but often instead they may have a poor perception of their own abilities; or they may simply be swamped with work; or there may be noise in the system. In the latter situations, an optimistic view of one's neighbors is called for. We believe there is a broad array of possible real-world applications which yield this situation: for example, cooperative "swarm-style" unmanned vehicular foraging and observation; or peer-to-peer networks; or distributing tasks over heterogeneous grid computing systems.

Bounded amortized risk yields a very simple, almost simplistic, decision rule which we call *flexible reciprocal altruism*. In a nutshell: the sum total degree to which an agent A is willing to provide help to another agent B is simply a linear function of the help A has received from B in the past.

As discussed later, this is not Tit-for-Tat. One of the consequences of flexible reciprocal altruism is that as the agents cooperate back and forth, they begin to be willing to provide larger and larger favors to one another, and furthermore, we may control this rate of growth. In Tit-for-Tat the size of favors remains fixed, as is the version in [4].

2 Related Work

The social science and economic communities have devoted considerable attention to cooperation, in particular, the nature of altruism and why it evolved [5, 6]. Continuing the same line of reasoning, several researchers examined how reciprocity and altruism effect the development of social agents [7–10].

More closely related to our work, Trivers examined how reciprocal altruism evolved in nature [11]. Using Triver's work as motivation, Axelrod developed some of the original theory of cooperation in a game-theoretic setting [12]. He developed an evolutionary model based on the probability that two agents would interact again in the future, and showed how stable behavior arrises, when agents exhibit deterministic reciprocity towards each other. Sen extended this work to probabilistic reciprocity, using publicly available discount factors to encourage sharing of resources [4, 13–15]. That work, in various guises, generally bases the probability of agent A assisting agent B directly on the historical *difference* in cost and benefit to A of interacting with B. This is related to our approach, which will instead increase the degree to which A helps B based on the *ratio* of historical cost and benefit; thus as A and B interact more, A is willing to help B to a larger and larger extent. We elaborate on a portion of this work [4] in Section 3. Hazard further broadened Sen's work by using private discount factors [16].

Altruism is also studied in cooperative game theory [17], which deals with coalition formation. Coalitions are groups of agents where the benefit received by an individual agent is higher when acting with the group than when acting alone. Shapley showed that under certain conditions the core of such a game is non-empty [18]. In other words, there exists a coalition structure such that no agent has incentive to change (similar to the idea of Nash equilibria). Fuzzy coalition theory seeks a middle ground between coalitions and self-interested agents [19, 20]. Economists have studied reciprocity and favors using cooperative game theory and fuzzy coalition theory, focusing on the economy that develops when favors are associated with a cost and a benefit [21–24]. A similar approach was used to improve wireless sensor networks between two organizations [25].

Closely related to reciprocity and cooperative game theory is the notion of trust and its extension in the form of reputation. Since interactions form the basis of a multi-agent system, the multi-agent community has devoted considerable effort towards understanding trust in large scale systems [26, 27]. Hand in hand with trust is the idea of reputation [28, 29]. While trust and reputation are significant research areas within multi-agent systems, our flexible reciprocal altruism does not use either mechanism per se; instead it relies on historical pairwise interactions.

3 Model Description

In flexible reciprocal altruism, an agent grants favors by considering the degree to which it has had favorable interactions with the grantee agent in the past. If the grantee agent has provided many significant favors in the past to the grantor, the grantor is willing to offer the grantee more unmatched favors in the future. Imagine that agent i is determining whether to grant agent j a favor. Let v_{ji} be the sum total *value* to agent i of the favors which agent j has granted agent i up to this point. Let c_{ij} be the sum total *cost* to agent i of the favors which agent i has granted agent j up to this point. Agent i will grant the proposed *favor* if:

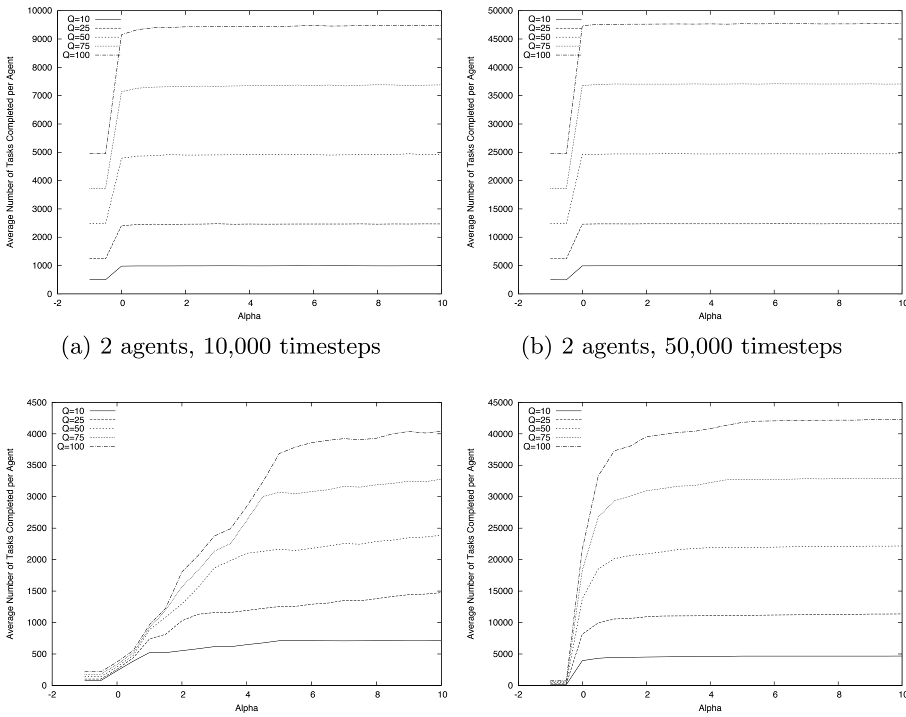
$$c_{ij} - v_{ji} + \text{cost}(\mathit{favor}) \leq \alpha v_{ji} + \beta.$$

The idea here is that the degree to which agent i will go out on a limb for agent j (that is, $c_{ij} + \text{cost}(\mathit{favor}) - v_{ji}$) is no more than some constant (α) times how well agent j has treated agent i in the past (that is, v_{ji}). α is essentially a measure of risk tolerance: highly altruistic agents may have a high value of α, whereas risk-averse agents will have a low value. An initialization constant, $\beta > 0$, determines the initial level of altruism, i.e., how many favors should an agent grant initially before requiring the other agent to reciprocate. Without β, agents would never grant favors to grantees with which they have had no history (that is, when $v_{ji} = 0$). Rearranging, we get:

$$\beta + (1 + \alpha) v_{ji} \geq c_{ij} + \text{cost}(\mathit{favor}). \quad (1)$$

This is our decision equation: a favor will be granted so long as it does not increase c_{ij} to greater than $\beta + (1 + \alpha) v_{ji}$. This decision equation is not based on rational agents per se: rather, it is based on amortized altruism. Imagine that agent i and j have had a history of interactions, and then agent j cheats i and walks away. To agent i this is acceptable if the *amortized ratio of cost to benefit over all interactions with* j does not exceed some amount. Clearly from the previous equation, $c_{ij} = O(v_{ji})$, and (rearranging yet again) the augmented cost to benefit ratio is:

$$\frac{c_{ij} + \text{cost}(\mathit{favor}) - \beta}{v_{ji}} \leq 1 + \alpha. \quad (2)$$



(a) 2 agents, 10,000 timesteps

(b) 2 agents, 50,000 timesteps

(c) 64 agents, 10,000 timesteps

(d) 64 agents, 50,000 timesteps

Fig. 1. Average number of completed tasks for 2 and 64 agents and various task queue lengths (Q).

As the agents gain a transaction history and c_{ij} and v_{ji} both grow, β becomes inconsequential and the ratio of cost to benefit approaches $1 + \alpha$.

Tit-for-Tat and the Reciprocal Norm It's worthwhile to compare this rule to others. Using our rule, if an agent A is being asked by another agent B for a string of favors, it will provide them up to its level of risk tolerance (α). Initially the allowed strings will be small, but ultimately they may be very large if needed. Because we can tune α, the rate of growth of allowed strings is not dependent on the cost and value of favors.

Now consider a simple variation of Tit-for-Tat which disregards the relative costs and values of the tasks to the agents. Here, agent A will only grant B a favor if the number of favors A has granted B in the past is less than or equal to the number B has granted A. If B needs a string of favors in a row, A will provide only enough of them to achieve parity plus one favor, and no more

We take liberties with the simple variation of Tit-for-Tat to consider costs and values in the tasks. In this "Expanded" version, an agent A will grant B a favor only if the *cost* to A of the favors A has granted B in the past is less than or equal to the *value* to A of the favors A has received from B. Assuming that all tasks have the same cost and the same value, this is equivalent to Equation 1 with $\alpha = 0$. In this model the strings of favors agents offer one another will grow without bound if the cost of a task is less than the value of the task. However this growth rate is entirely determined by the ratio of cost to value, since α is disregarded. Furthermore, if the cost and value are the same, then this model reduces to the simple Tit-for-Tat scenario earlier.

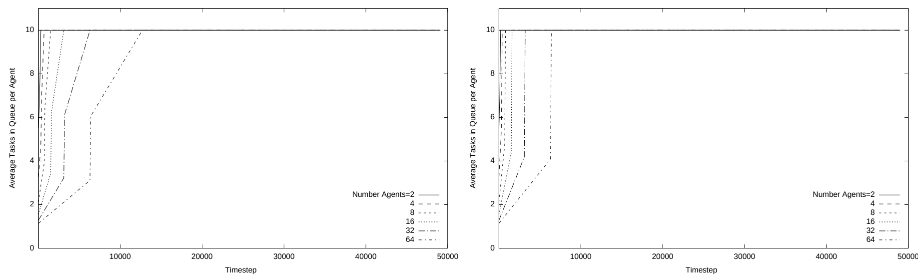
A more elaborate system described in [4] permits consideration of various task types with different costs and values, expectation of future distribution of tasks by type, and a probability model of other agents providing future favors. The authors suggest a reduced version in which both the future distribution of tasks and the future favor probability model are estimated using the results from the immediately previous time step (we may generalize this to some window of previous time steps). If we employ a single task type with a given cost and value, as is done in the experiments in this paper, the model may be simplified to $Pr_{A,B} \overset{?}{<} Pr_{B,A}$, where $Pr_{i,j}$ is the recent probability that, when asked, i has given j a favor. We believe this is a manifestation of the so-called *reciprocal norm*, that is, A gives B favors because he feels indebted to B. We would augment this to $Pr_{A,B} \overset{?}{\leq} Pr_{B,A}$ to enable granting when the two are evenly matched. An approach along these lines, like Tit-for-Tat, fixes the length of favor strings if the two agents are evenly matched.

We note that our method is also in some sense more robust to noise once the agents have warmed to one another. Imagine if agent A believes he has given agent B a favor or two, but B did not receive them due to noise. In the three systems described above, ultimately the situation may arise where each agent believes the other owes him, and refuses further interactions. In our approach, depending on how many interactions A and B have had with one another, this situation can only arise after a large number of noise-caused failures.

Free Riders Because of its optimism, our approach is susceptible to free-riding agents ("leeches") in two ways. First, if an agent A meets an unknown agent B for the first time, A will always agree to offer B a small initial favor no greater than β. Were there an infinite number of agents, B could wander about forever, getting one favor from each agent. Second, a smart agent B could ratchet up the favors he steals from A by building up a transaction history with A, then asking for a large number of favors, then walking away.

It's important to note that, from the perspective of A, the first situation is in some sense worse. Consider the case where the cost and value of favors are both 1. For B to prime A to give out an unmatched string of N favors, B would have had to provide A with some $\frac{N-1}{\alpha}$ favors in turn (and vice versa). But if some N rogue friends of B's each asked A for one favor, A would offer it to all of them without any reciprocation.

In a finite-sized group, free-riders of the first kind are capped automatically: they'll run out of people to cheat. Free-riders of the second kind can extract any single amount they wish from the altruistic agents, but to extract more they will need to give back more. The total amount they can cheat the system



(a) $\alpha = 0.0$, 50,000 timesteps (b) $\alpha = 1.0$, 50,000 timesteps

Fig. 2. Average queue length per agent. Maximum queue length was 10.

is unlimited, but the *ratio* of loss, with respect to the value they provided the system, is bounded. Importantly, *this ratio may be set by the altruistic agents:* if they are risk averse they may cut α clear to 0, in which their loss is never more than β (here, 1). Or they can cut α clear even further, down to -1, at which point they never grant more than a single favor to a requester, and ignore him after that.

Favor Brokers Though our approach allows agents to warm to one another relatively quickly regardless of the ratio of cost to value, it also makes possible extensions which promise even faster warming. We are particularly interested in the notion of *favor brokers* which can act as a bridge between two agents who otherwise know little about one another. Consider two agents A and C who do not know one another well. A needs a significant favor but C ordinarily would not provide it because of limited transaction history between the two. However, broker B has a significant history with both A and C. A asks B for a favor, and B agrees to broker it by in turn asking C to perform the favor. C agrees to do the favor for B. A receives the value and C the cost of the favor; but C believes that C has done B, not A, a favor ; and A likewise believes that B, and not C, has done A a favor. Note that neither A nor C have changed in opinion about one another.

The idea behind brokerage is to provide a special channel whereby a population may offer favors despite little prior history; or for two alien populations to offer favors to one another. Though this approach is related to notions of reputation and trust (for example [29]) it is *different* in an important way: B is not introducing A and C to one another. At the end of the day, A and C have no additional transaction history. B has not used its reputation to convince C that A is worthwhile; rather B has assumed responsibility as a middle-man for the transaction.

Though it would be interesting for future study, we do not at this time permit transitivity in brokerage.

4 Experiments

Our experimental problem is an abstraction of common basic factory-floor models. Each agent has a queue of a certain *length* (how many tasks it can hold), and each timestep it removes a single task from its queue, if there is any, and performs it. As time passes, new tasks stochastically arrive for each agent, sometimes many at a time. If the agent's queue is not yet full, it will put the tasks in its own queue. Otherwise the agent will ask others for favors: to put the tasks in their queues instead. If an agent cannot curry a favor, and his queue is full, the task will be dropped on the floor and be lost.

When an agent places a task into his queue (to perform it in the future) he immediately incurs a *cost*, but the task's owner receives a *value*. Costs are less than or equal to values: and in our experiments all tasks have the same cost and the same value. Thus granting a favor costs the grantor in two ways. First, when a grantor performs a grantee's task, the grantor incurs the cost but the grantee receives the value. Second, by accepting tasks, the grantor runs the risk of filling his own queue such that newly arriving tasks of his own are dropped on the floor.

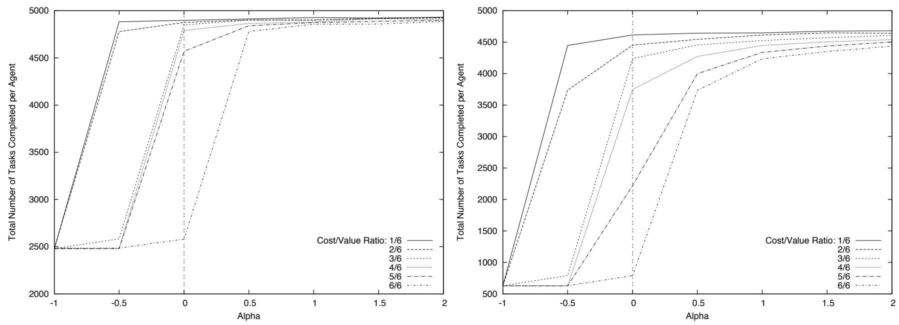
Tasks arrive following a Poisson distribution with mean of 100 timesteps, and normally round robin selection determines which agent receives the tasks. When asking for a favor, agent A asks a random agent B, and if B does not grant the favor, then A chooses another random agent to ask, then another, and so on. If ultimately no agent grants A's favor request, then the task is lost. α usually ranged from -1 to 10 in steps of 0.5. Runtime was set to either 10,000 or 50,000 timesteps. All experiments were done in the MASON simulator [30] and the results were averaged over 50 independent runs. Except in Experiment 3, task value was set to 3 and task cost to 2.

Our primary statistic was number of tasks performed per agent in the allotted time. We chose this, rather than total value minus total cost, because it is insensitive to the particular task costs and values chosen (in particular, if task cost and value are the same, then total value minus total cost will always be zero). Another statistic was the average number of tasks completed per agent in a given timeframe.

Experiment 1: Task Load, Number of Agents, and Degree of Altruism We first set out to examine the dynamics of the system: what happens with increasing load (shorter queue lengths), more agents, and varying amounts of altruism (α). Figure 1 shows the results. In the left column we show the results for 10,000 timesteps, which emphasizes the warm-up period until the agents were offering large favors to one another; and in the right column we show 50,000 timesteps, which is closer to the steady state of the system.

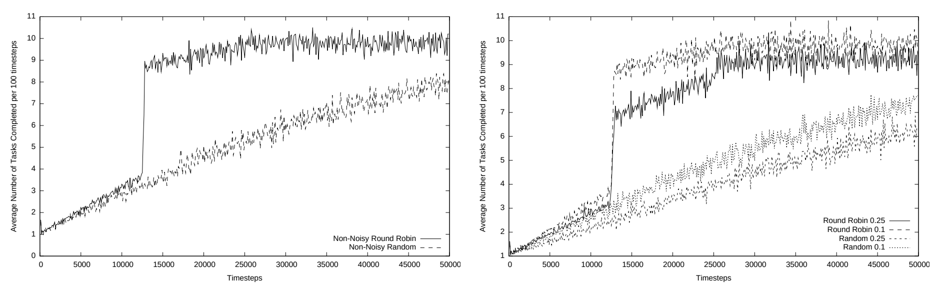
As would be obvious, smaller queue lengths resulted in more tasks being dropped on the floor, and thus lower numbers of tasks completed per agent.

Figure 1 also demonstrates that the more altruistic agents (with larger values of α) are able to warm up to one another more rapidly. Agents with $\alpha = -1$ (approximately greedy agents who only provide just one favor per grantee for the



(a) 2 agents, queue length 50 (b) 8 agents, queue length 50

Fig. 3. Total number of tasks per agent (10,000 timesteps), with different ratios of cost/value for each task. "Expanded Tit-for-Tat" ($\alpha = 0$) is marked with a vertical line to distinguish it.



(a) No Noise (b) Different Noise Levels

Fig. 4. Average number of tasks completed per agent per 100 timesteps for round-robin and random task distribution, $\alpha = 1.0, \beta = 2.0$, queue length = 10, 125 agents

entire duration of the run) cannot improve. "Expanded Tit-for-Tat" style agents, with $\alpha = 0$, are also able to warm to one another because their decisions are based on the ratio of value to cost rather than on the specific number of favors granted. But for agents with larger α values, the rate of warming increases even more, resulting in more tasks competed per agent.

Last, but most importantly, it would seem intuitive that having more agents would allow for more favor opportunities, and this in turn should improve the average number of tasks performed. But in fact this is not the case: as he spreads his favors over more agents at random, a grantee takes longer to warm up to any given grantor. As a result, the total number of tasks performed is reduced until the agents have adequately warmed to one another. Ultimately, however, the steady state results converge to the same values regardless of the number of agents.

Experiment 2: Queue Saturation To verify that larger numbers of agents were in fact taking longer to warm up, we examined how full agents' queues were during the task. Figure 2 shows the number of tasks waiting in a queue, per agent, for $\alpha = 1$ (other α values were similar). The maximum queue length was 10. Note that as the number of agents increases, there are fewer interactions, and so the agents are idle until much longer in the run; however, at the steady state, eventually all the queues are filled.

Experiment 3: Tit-for-Tat Experiment 1 showed that "Expanded Tit-for-Tat" ($\alpha = 0$) would in fact warm to larger and larger favors. However, this is mostly due to the cost/value ratio involved. By default we had set this ratio to $\frac{2}{3}$. But if the cost and value of a task were the same, then Tit-for-Tat can do scarcely better than approximately greedy agents ($\alpha = -1$). This is shown in Figure 3. As the cost/value ratio approaches 1, Tit-for-Tat starts performing poorly. In fact, the more agents involved, the worse Tit-for-Tat performs, since each agent must make up favors over more agents without a warm-up mechanism. Larger values of α (that is, more altruistic agents) are much less sensitive to this ratio.

Experiment 4: Task Allocation and Noise Next we studied how changing which agent receives the tasks effects performance. We examined two methods for choosing which agent receives tasks: round robin selection (our previous default) and random selection. In this experiment, we also added noise to the system by increasing the probability that the grantor does not receive any value. In other words, the grantee will think it granted a favor while the grantor will think the favor was not granted.

We set the number of agents to 125, $\alpha = 1.0$, $\beta = 2$, and the queue length to 10. For the noisy experiments, we set the noise level at 0.1 and 0.25 (a noise level of 0.1 meant that 10% of the time, the grantor would not receive any value).

Figure 4(a) shows that changing how agents receive tasks alters performance. In the round-robin case, after all the agents have received the tasks once (around timestep 12,500), all the agents will then grant one another a favor; thus the spike in the graph. If instead we chose agents at random to receive tasks, the agents require additional time to build a history, thus resulting in poorer performance. Figure 4(b) shows how noise effects system performance. As expected, increasing the noise level caused performance to drop. However, our model formulation overcomes the noise, although it requires more time to do so.

Experiment 5: Initially Altruistic Societies This experiment studied one possible method to decrease the warmup period. In flexible reciprocal altruism, β controls how *initially* altruistic the society is: as β increases, agents are willing to grant more initial favors before expecting the other agent to reciprocate. In other words, increasing β decreases the warmup period.

Figure 5 shows how the length of the warmup period changes with β, given $\alpha = 1.0$, maximum queue length of 10, and 125 agents. We ran the experiment for 50,000 timesteps but truncated the figure to 20,000 timesteps to emphasize the warmup period; after 20,000 timesteps all the curves are statistically the

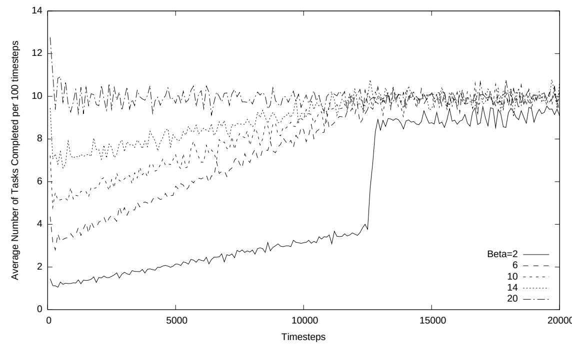


Fig. 5. Average number of tasks completed for different levels of initial altruism. $\alpha = 1.0$, queue length = 10, and 125 agents.

same. Note that for a very altruistic society (high β) the average throughput is maximal from the beginning.

Experiment 6: Favor Brokers Finally, we studied how introducing favor brokers influences performance. Recall that a *broker* is an intermediary who can ask a favor of one agent on behalf of another, despite possible distrust between the two. In a society with low altruism, brokers (and their trusted status) offer a potential way to increase interactions amongst the agents, and thus decrease the warmup period. Figure 6 shows the average number of tasks completed per 100 timesteps per agent, for $\alpha = 1.0$, $\beta = 2.0$, a maximum queue size of 10 and 50 agents. Increasing the percentage of brokers in the population increases the number of interactions between agents, thus causing quicker convergence even in a society that is not very altruistic. Changing α and the number of agents does not significantly alter the results.

This experiment, along with Experiment 5, suggests that either increasing the global level of altruism, or increasing the number of brokers within the society, will decrease the warmup period. While both techniques could be combined, the problem domains where such an arrangement is sensible may be limited.

5 Conclusions and Future Work

We have presented a simple formula which enables cooperating agents to warm to one another, offering larger and larger favors based on past experience. Though at any time an agent can cheat another agent and walk away, the second agent's amortized loss is bounded. This optimistic approach, which we have termed flexible reciprocal altruism, allows agents to optimize the collective performance of the system, and is particularly apropos to environments with moderate numbers of agents, significant and highly variable task load, and moderate noise. Our technique adapts faster and more flexibly than Tit-for-Tat and similar tech-

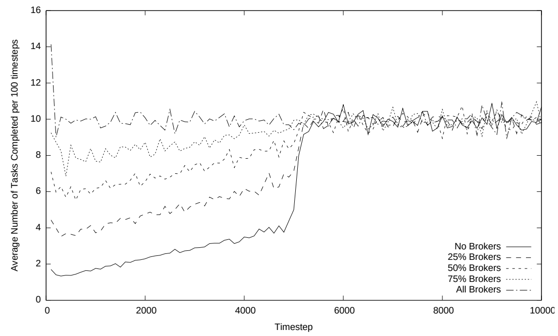


Fig. 6. Average number of tasks completed per 100 timesteps for varying percentage of brokers in the population. $\alpha = 1, \beta = 2$, queue length = 10, and 50 agents.

niques, and can handle situations they cannot: such as when the cost and value of a task are the same.

For future work: our existing approach lacks a time-discounting procedure: agents have perfect memory of all previous favor transactions. We have not yet examined the effects of different ways of doing this: for example, what would be the result of having longer memories for costs than for values?

We may also examine possible approaches to transitivity in favor brokerage, and using brokers to introduce entire disjoint populations to one another. Last, we have not experimented with per-agent values of α, or different α values on a pairwise basis. It might also be helpful to consider dynamic α values, producing nonlinear functions or ones designed to cut off freeloaders more effectively.

Acknowledgments This work was supported in part by NSF grant 0916870.

References

1. Luke, S., Sullivan, K., Balan, G., Panait, L.: Tunably decentralized algorithms for cooperative target observation. In: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, Utrecht Netherlands (July 2005)
2. Gheorghe, M., Holcomb, M., Kefalas, P.: Computational models of collective foraging. Biosystems **61**(2–3) (July 2001) 133–141
3. Camorlinga, S., Barker, K., Anderson, J.: Multiagent systems for resource allocation in peer-to-peer systems. In: Proceedings of International Symposium on Information Communication and Technologies. (2004) 1–6
4. Saha, S., Sen, S., Dutta, P.S.: Helping based on future expectations. In: Proceedings of Autonomous Agents and Multiagent Systems. (2003)
5. Dugatkin, L.A., Wilson, D.S., III, L.F., Wilkens, R.T.: Altruism, tit for tat and 'outlaw' genes. Evolutionary Ecology **8**(4) (1994)
6. Lehmann, L.: The evolution of cooperation and altruism–a general framework and a classification of models. Evolutionary Biology **19**(5) (2006)
7. Briggs, W., Cook, D.: Flexible social laws. In: Proceedings 14th International Joint Conference on Artificial Intelligence. (1995) 688–693

8. Grimaldo, F., Lozano, M., Barber, F.: A multiagent framework to animate socially intelligent agents. In: Innovations in Hybrid Intelligent Systems. Springer (2008)
9. Lerman, K., Shehory, O.: Coalition formation for large-scale electronic markets. In: Proceedings of Fourth International Conference on Multi-Agent Systems. (2000)
10. Matsubayashi, K., Tokoro, M.: A collaboration mechanism on positive interactions in multi-agent environments. In: Proceedings of International Joint Conference on Artificial Intelligence. (1993)
11. Trivers, R.L.: The evolution of reciprocal altruism. The Quarterly Review of Biology **46**(1) (1971) 35–57
12. Alexrod, R.: The Evolution of Cooperation. Basic Books (1984)
13. Saha, S., Sen, S.: Reciprocal negotiation over shared resources in agent societies. In: Proceedings of Autonomous Agents and Multiagent Systems. (2007)
14. Sen, S.: Reciprocity: A foundational principle for promoting cooperative behavior among self-interested agents. In: Proceedings of the Second International Conference on Multiagent Systems. (1996) 332–329
15. Sen, S.: Believing others: Pros and cons. Artificial Intelligence (2002)
16. Hazard, C.J.: ¿Por favor? Favor reciprocation when agents have private discounting. In: Proceedings of the 2008 AAAI Workshop on Coordination, Organizations, Institutions and Norms. (July 2008) 9–16
17. von Neumann, J., Morgenstern, O.: Theory of Games and Economic Behavior. Princeton University Press (1953)
18. Shapley, L.S.: Cores of convex games. International Journal of Game Theory **1**(1) (1971) 11–26
19. Branzei, R., Dimitrov, D., Tijs, S.: Models in Cooperative Game Theory: Crisp, Fuzzy, and Multi-Choice Games. Springer (2005)
20. Mareš, M.: Fuzzy Cooperative Games: Cooperation with Vague Expectations. Physica-Verlag (2001)
21. Abdulkadiroğlu, A., Bagwell, K.: Trust, reciprocity and favors in cooperative relationships. In: Proceedings of University of Maryland, Department of Economics, Workshop in Industrial Organization and Microeconomic Theory. (January 2005)
22. Fung, K.K.: Doing well by doing good: A market for favors. Cato Journal **15**(1) (1995)
23. Hauser, C., Hopenhayn, H.: Trading favors: Optimal exchange and forgiveness. In: Society for Economic Dynamics Annual Meetings. (2004)
24. Möbius, M.M.: Trading favors. Manuscript (May 2001)
25. Miller, D.A., Tilak, S., Fountain, T.: "Token" equilibria in sensor networks with multiple sponsors. In: Proceedings of International Conference on Collaborative Computing: Networking, Applications and Worksharing. (2005)
26. Ramchurn, S.D., Huynh, D., Jennings, N.R.: Trust in multi-agent systems. The Knowledge Engineering Review **19**(1) (2004) 1–25
27. Wang, Y., Singh, M.P.: Formal trust model for multiagent systems. In: Proceedings of International Joint Conference on Artificial Intelligence. (2007)
28. Pujol, J.M., Sangüesa, R., Delgado, J.: Extracting reputation in multi agent systems by means of social network topology. In: Proceedings of Autonomous Agents and Multiagent Systems. (2002) 467–474
29. Sabater, J., Sierra, C.: Reputation and social network analysis in multi-agent systems. In: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems, New York, NY, USA, ACM (2002) 475–482
30. Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., Balan, G.: MASON: A multiagent simulation environment. Simulation **81**(7) (July 2005) 517–527