Improving Cooperation in Spatially Distributed Agents

Sara McGrath, John Anderson, and Jacky Baltes
Department of Computer Science, University of Manitoba, Winnipeg, Canada R3T2N2
ummcgrat,andersj,jacky@cs.umanitoba.ca

Abstract

Agent trust and its effect on cooperation is an important concern in multi-agent systems, both in improving cooperation in application areas such as electronic marketplaces, and understanding its effects on low-level agent interaction. We present a mechanism for facilitating cooperation in multi-agent systems based on that of Dutta and Sen [4], that adds spatial awareness and removes the assumption of global knowledge of other agents. This approach employs local methods to propagate the existence of other agents as well as trust information. We evaluate this mechanism in comparison to that of Dutta and Sen, and illustrate that it improves performance in a spatially oriented environment, both in terms of the quality and quantity of work performed.

1 Introduction

The ability to trust other agents is an obvious factor in allowing cooperation between agents to ensue. The nature of trust and its effect on cooperation have thus proven to be important topics in both multi-agent systems (MAS) and game theory. While there has been a great deal of work in these areas from a theoretical standpoint (e.g. the seminal iterative prisoner's dilemma (IPD) work of Axelrod [2] and related works that have followed it in trust and coalition formation), application-oriented multi-agent systems research has also been inspired by real-world problems of trust. The modern electronic marketplace, for example, has provided many challenges in identifying both rogue agents and potentially good trading partners, and policing fraud and deception. Trust and reputation management is also an important component of peer-to-peer (P2P) networking. Agents in these systems must be able to trust each other (here in terms of receiving valuable data from an agent one agrees to share data with), or the system will collapse. These tend to use Prisoner's Dilemma (PD)-like techniques as part of an interaction policy between agents. BitTorrent [3], for example is a P2P system that enforces a reciprocative [8, 2] or tit-for-tat-like policy on file sharing: peers will only share with peers who share with them.

Recent work on trust and cooperation in multi-agent systems can be divided into two categories: high level work that focuses on particular elements of trust or specific application areas (e.g. [5]), and low-level work on basic agent interaction in problems that epitomize a number of areas (e.g. [1, 8, 7, 4]). Within the latter realm, one important issue is that of spatial locality and its affect on stable cooperation between agents. While there has been some work on both the effects of building up trust in neighbourhoods of agents [7, 4], and including mobility [1], much work on spatial locality has emphasized evolution: how population distributions change when agents can alter their strategies based on the success of others (e.g. [2] and later works).

We are interested in studying the spatial aspects of trust and cooperation in low-level agent interaction while considering the issue of scalability. In order to ultimately be applicable to the real world, the approach used to choose agents to interact with must not make broad assumptions, such as global knowledge of other agents, that cannot be supported in large domains. The more such constraints are discarded, the greater the likelihood that work will be of use to higher-level multi-agent systems research, and to immediate applications. In this paper we present an approach to managing trust and cooperation in spatially-oriented multi-agent systems. We illustrate an implementation of this approach, and compare its performance to that of Dutta and Sen [4], the closest and best-performing prior work to this. We begin with a review of related work in this area.

2 Related Work

As stated above, much work in MAS has emerged from Axelrod's IPD studies [2]. An IPD involves a standard PD scenario, which has been shown by Ax-

elrod to subsume a great many real-world problems in areas from commerce to biology, in a setting where agents will encounter one another repeatedly. Agents thus have the ability to respond to the prior behaviour of other agents in each new encounter. Memory of prior interactions is the most obvious addition to a primitive agent model in order to deal successfully with IPD situations, and significant exploration of aspects of memory, including limiting agent memory, has ensued [1, 8]. At a higher level, Esfandiari and Chandrasekharan [5] began looking at simple ways to gain and spread trust through a network of agents. Meanwhile, Dutta and Sen [7] continued looking at increasingly complex means of not only encouraging emergent cooperation (with inherent memory), but proactively adapting to fluctuations in supply and demand of skills while continuing to cooperate.

Beyond memory, the issue of spatial locality and its effect on cooperation has also been explored. Axelrod [2] looked at the evolution of spatially distributed societies as agents were allowed to change strategy in response to interactions with others, and Nowak and May [6] later extensively studied the ratios between payoffs in a PD scenario necessary for cooperation to emerge. In an iterative setting, Armstrong and Durfee [1] examined the effect of agents that could move, but used spatial locality and the possibility of movement only to examine the effects of memory.

Many of the lower-level approaches to agent interaction make strong assumptions about the nature of agents: global awareness of all other agents in the environment, restriction of interest to only one product in a marketplace, or participation in only one task or group, for example. Each of these assumptions is limiting in terms of the applicability of results to the real world. In recent work, Dutta and Sen [4] have examined the issue of supporting stable cooperation among groups of agents that deals with a number of these assumptions, and is the closest work to that described here. We explore their approach in the following subsection.

2.1 Dutta and Sen: Forming Stable Partnerships

The cooperative domain used in Dutta and Sen's recent work [4] involves the distribution of expertise among agents. There are several types of abstract tasks in the domain, with each agent being expert in one and only one task type. An expert agent can complete a task with high quality and in little time; a non-expert will require much time and deliver a lower quality result. The actual values of time and quality are drawn from normal distributions depending on the agent's expertise at that task type. They denote the cost of a task i for agent k as C_i^k, and as C_{ij}^k if task j it is completed by agent k for agent$_i$. Each of M agents are each randomly assigned N tasks without regard to type. The simulation ends when all agents have completed all of their tasks. Cooperation here is of obvious benefit; if agents can recruit others with greater expertise for tasks than they possess themselves, greater performance will result. An agent asked for help may not immediately have a reciprocal request, and such help must be returned later. This temporal distance allows for agents to be exploited.

Dutta and Sen's basis for cooperation lies in each agent maintaining cost balance information on other agents: how much each owes the agent or is owed by the agent in terms of work performed. Dutta and Sen describe a number of agent types based primarily on the probability that an agent will assist another agent when requested, given the current balance. This probability, $Pr(i, j, k)$ that agent i will help agent k to perform task j, is based on the cost of the task, how much more or less it will cost the agent than its average prior tasks have, and how the agent (and possibly other agents) view the requestor:

$$Pr(i,k,j) = \frac{1}{1 + exp\frac{C_{ij}^k - \beta * C_{avg}^k - OP_i}{\tau}} \quad (1)$$

OP_i is the balance between other agents and agent i, and β (0.5) and τ (0.75) are parameters that control the shape of the curve produced by Equation 1. With these settings, the initial probability of helping another agent is approximately 0.5.

Agents use estimates to keep track of who is good at particular tasks to make appropriate choices for assistance. Every agent keeps track of all others expected time and quality for task types, starting with neutrality and updating as it interacts with others, weighting each new episode according to a learning factor α (this value was not stated in [4]: we employed the common value of 0.1 in our reproduced implementation).

Dutta and Sen define several agent types based on variations of these concepts. *Earned Trust Reciprocative* agents, which we will term *helpful* here for convenience, are agents that assist using probability formula (1). They are reciprocative based on their own balances, but also those of others: when an agent requests help, inquiries about the balance of that agent are made to all agents with which a positive balance exists. They respond honestly to similar requests from others, allowing these agents to exploit one another's knowledge. *Individual Lying Selfish* agents, which we

will term *selfish* agents, do not cooperate with requests for assistance, but still try to exploit others by asking for assistance themselves. Further, these agents always report balances with others as negative values if they are in fact positive, thus damaging the reputations of helpful agents.

Dutta and Sen's implementation employs 3 task types, 100 agents, and 100 to 1000 tasks per agent. Each turn, each agent receives one task of its own and will either perform the task itself, or get another agent to perform the task for it. Agents may perform an unlimited number of tasks for other agents each turn. Every agent has a list of all other agents, kept sorted based on the estimated cost of completing this turn's task's type. The agent asks each agent in list order for help on its task. If it reaches itself before any agent assists it, it does the task itself.

3 Considering Spatial Distribution

The implementation described in the previous section is one of the better to be found in the literature in terms of eliminating unrealistic assumptions. However, there are two important aspects that Dutta and Sen have omitted, which we address in this paper. First, there is an assumption here that spatial distribution does not matter: agents simply know everyone else and can deal with everyone. Second, maintaining a list of all agents in the system, let alone keeping it sorted by estimated task cost, is not possible in larger societies of agents. In this section, we describe an approach and implementation that removes these assumptions.

Without these assumptions, agents must discover potential partners over time, in addition to who good partners are. When we consider the element of spatial distribution added to the concepts detailed in section 2.1, a logical approach is to begin with spatial neighbours. Each agent in our approach begins knowing the existence of only its immediate neighbours. Such agents also require mechanism for discovering distant agents. Asking neighbours for the identities of other agents is a logical approach once the spatial nature of the domain is taken into account. This general concept has had success in referral systems [10], as well as in P2P systems [9]. In our approach, agents may request new partner information from current partners each turn, and requests on a turn are allowed until known agents are exhausted or new information is received. Responding agents share only their partners' identifiers (IDs), and so each agent initially knows no details of new agents it discovers. This also makes for a more realistic scenario: in large systems where agents can join and leave often, treating existing agents similarly to new ones allows the approach to scale, as opposed to maintaining additional information about previously existing, but unknown, agents.

The agent types we define are based on those defined by Dutta and Sen [4]. *Helpful* agents are willing to share information on other agents probabilistically. We employ a variant of the probability formula (1) – removing the task costs – to determine if an agent is willing to share the IDs of its partners. Once a helpful agent decides to share partner IDs, it only shares nicely – that is, it will share all (and only) partners with a balance > 0. In respects other than sharing partners, our helpful agents are identical to Dutta and Sen's, in order to facilitate comparison. *Selfish* agents behave similarly to those in [4] (i.e. uncooperative, lying), but must also have a selfish policy for sharing information about new potential partners. Here, selfish agents are willing to spread knowledge of the existence of other agents, but *only bad ones*: those with a balance < 0.

As stated above, agents can request partner lists until new information is received. However, we must also have a strategy for choosing which agent to begin asking for partners from. Agents assume that helpful agents will share better partner lists, and so need not ask for partners of agents with negative balances. Further, agents need to ask as many agents as possible for their partners, in order to explore the space of agents. Thus, we do not want to always ask the first agent on the list of partners, because unlike the complete lists of [4], this position will not change. We could ask the person with the highest balance, but this may not change much, leading to a local maximum. As we need some randomization to explore the space of potential partners, all agents in our approach randomly choose a known agent, and if that agent has a positive balance, ask it to share its partner list.

This approach was implemented using Swarm, a common MAS simulator, to model the agents and domain. We evaluated the effects of introducing the concept of physical space and the removal of global agent knowledge in comparison to an implementation of Dutta and Sen's approach according to pseudocode supplied in [4]. The next section describes the results of this evaluation.

3.1 Evaluation

We began by reimplementing the approach of Dutta and Sen [4], and attempting to replicate their results. We employed the same values for β and τ in formula

(1), and used a learning parameter α of 0.1, using the same 3 task types, 100 agents (50% selfish, 50% helpful), and 500 tasks per agent, with each assigned to an expertise by a uniform random distribution.

We found that while helpful agents performed at least as well as the results shown in [4], and with a similar linear fit, selfish agents performed more poorly. This led to some interesting observations in their results. We calculated the time and quality results for doing all tasks alone (i.e. no help) as a low baseline vs. optimal baseline of having expert help on both task types in which an agent was not an expert. Since tasks are equally distributed, doing all tasks yourself results in average times of two HIGH and one LOW for every three tasks (since lower expertise results in higher completion time), while quality should be two LOW and one HIGH (since lower expertise results in lower quality results). Thus, we should see

$$\text{time}_{avg} = (2{*}\text{HIGH}{+}\text{LOW})/3 = (20+1)/3 = 7$$
$$\text{quality}_{avg} = (\text{HIGH}{+}2{*}\text{LOW})/3 = (2+10)/3 = 4$$

Similarly, if we obtain expert help on all tasks, all times should be LOW, and all quality HIGH, leading to an average time of 1, and an average quality of 10. These can be used for guidelines as to how many tasks agents are doing themselves; averages around a time of 7 and quality of 4 (a *base* average) indicates they are mostly working by themselves, where 1 and 10 (an *optimal* average) indicate a considerable amount of help. Because learning in this environment is very simple, after the first few turns, agents are almost certain to only ask experts for their help. Once an agent begins to ask non-expert agents for help, the help probability calculations ensure they are unlikely to receive any, and will have to do the task themselves. Thus, we would expect agents who are being helped regularly to have optimal (or near-optimal) averages. Selfish agents, because they will not help and should not be helped in return, should have averages around the base average.

Dutta and Sen [4] do not calculate the actual averages for their agents, instead plotting average time and quality totals for each agent. If, however, we calculate the actual averages from the number of tasks, the data shown in Figure 1 is obtained.

The values in these tables are created from previously averaged data, and so are not precise. However, even considering this inaccuracy, we can observe two interesting phenomena. First, agents are not learning to cooperate with each other as well (or as long) as expected. The results shown in [4] only show the first 100 tasks onward, but by that time, learning has

		Reciprocative		
Tasks	Total time	Avg Time	Total Qual	Avg Qual
100	400	4	700	7
200	800	4	1400	7
400	1500	3.75	2800	7
800	3200	4	5600	7
1000	3700	3.7	7000	7
		Selfish		
Tasks	Total time	Avg Time	Total Qual	Avg Qual
100	0	0	1000	10
200	500	1.25	1100	5.5
400	1800	4.5	1200	3
800	4500	5.75	1400	1.5
1000	6000	6	1400	1.4

Figure 1: Dutta and Sen's average times and quality for selfish and reciprocative agents.

largely converged. Average times don't change, implying that helpful agents aren't avoiding exploitation after the first 100 tasks have been accomplished. Second, selfish agents' average quality totals change very little between 100 and 1000 tasks (from 1000 to 1400), and most of the change occurs before the 400th task (where agents are at 1200). This implies that the 50 selfish agents were able to complete up to 900 tasks each with a net quality increase of approximately 400, an average quality of approximately 0.45, with average task cost approaching 0 as the number of tasks approaches 1000. As the lowest quality is drawn from the LOW distribution of 1, this seems highly unlikely. A more reasonable explanation is that selfish agents are doing every other task, and getting another agent to perform the other tasks for them. Dutta and Sen's results only show averages of the total task times and qualities per agent, so there is no way to know if reciprocative agents are performing more tasks than would seem appropriate. They do look at the average savings accumulated by both selfish and reciprocative agents for both 20 and 40 tasks, with varying percentages of agents. At 50% selfish agents, with 20 tasks, reciprocative agents have average cost savings of about 100 and selfish about 30, while with 40 tasks, reciprocative agents have average cost savings of 400, and selfish still about 30. These figures would suggest that reciprocative agents are gaining greater savings with other reciprocative agents (as the selfish ones are not saving much), but contradicts the data in Figure 1.

After considering these possibilities, we ran our simulation with only selfish agents, and only reciprocative agents. As expected, purely selfish agents stayed at the base averages, while purely reciprocative agents quickly moved closer to the optimal averages, indicating that each of these agent types behaved as they

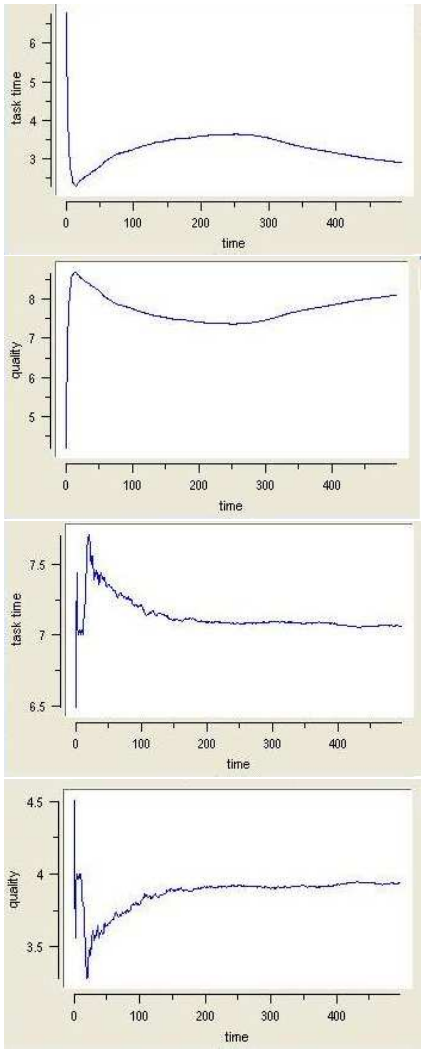

Figure 2: Average time and quality for only helpful (top) and only selfish (bottom) agents, from a reproduction of the experiments of [4].

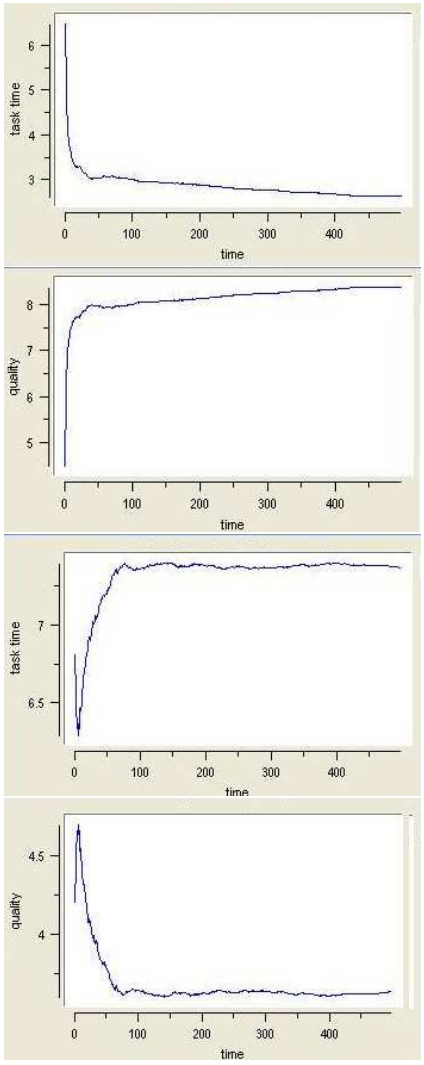

Figure 3: Average time and quality for helpful (top) and selfish (bottom) agents with local agent knowledge in the vertical, horizontal, and diagonal directions.

should in our implementation (Figure 2). We obtained better results when we removed the exploration period specified in [4] which forced each agent to ask every agent for help once. Helpful agents form stronger bonds more quickly, and stabilize at higher performance levels. As accurate replication of [4] was our intent, only the original baseline results are shown.

In evaluating the extensions described in Section 3, we distributed 100 agents (50% of each type) on a square spatial layout grid with no empty space, and allowed each to know of the existence of the (up to) 8 neighbours surrounding any agent. The resulting performance of this configuration (Figure 3) shows that under more challenging conditions - considering space and removing the assumption of knowledge of all agents - helpful agents do somewhat better than that those of Dutta and Sen [4]. Further, selfish agents do more poorly, indicating less exploitation in the system.

There is still much room for improvement in the spacial considerations described in Section 3, as there are still situations where agents do not become known to one another. Consider the agent distribution in Figure 4, where two light (helpful) agents highlighted by a dashed ellipse are surrounded by dark (selfish)

agents. They will know each other, but no other agents (since the selfish agents surrounding them will never pass on knowledge of any nearby reciprocative agents). Their performance is thus limited.

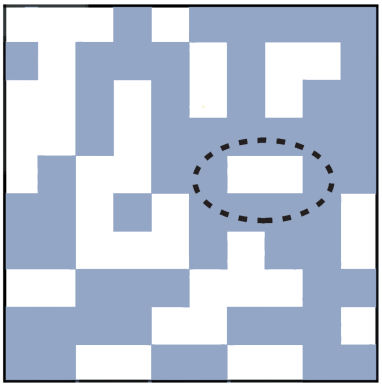

Figure 4: World map of helpful (light squares) and selfish (dark) agents, with an area of significance highlighted.

4 Conclusions and Future Work

In this paper, we compared the previous work of Dutta and Sen to an approach we developed that is spatially aware and removes the assumption of knowledge of all agents. Agents no longer know of every other agent through a global mechanism, but discover who is in the world themselves, resulting in a more distributed and much more scaleable algorithm.

In replicating Dutta and Sen's implementation, we noted interesting implications of their previous results, as selfish agents appear to break down cooperation instead of being refused cooperation, forcing helpful agents to work on their own. Averaging agent performance gave a clearer picture of agent behaviour, demonstrating that learning was very basic in this domain, and worked best without much exploration.

There are still many opportunities for future work. In this domain and in [4], agents are able to complete as many tasks as they accept in one turn. This is clearly unrealistic; future experiments should explore agents only being able to complete a few tasks a turn, and could look at the implications for bottlenecks, and scarcity of "good" agents in this case. Balancing the workload among the agents would become more important, as agents could be easily overworked and unavailable if all agents asked the same ones for help.

Further, the agent types in this experiment (and in Dutta and Sen's work) are still reasonably simple. More complex agents (e.g., agents that lie some portion of the time) may have a profound impact on the system. Similarly, agents could be selfish but truthful, or helpful but lying. In such cases (as in the real world) it would likely be more helpful to share partner lists based on a measure of perceived honesty instead of a balance of work shared. It would also be interesting to add new agents over time.

References

[1] Aaron A. Armstrong and Edmund H. Durfee. Mixing and Memory: Emergent Cooperation in an Information Marketplace. In *Proceedings of ICMAS-98*, pages 34–41, 1998.

[2] R. Axelrod. *The Evolution of Cooperation*. Penguin Books, England, 1984.

[3] Bram Cohen. Incentives Build Robustness in BitTorrent. In *Proceedings of the 1st Workshop on Economics of Peer-to-Peer Systems*, Berkeley, 2003.

[4] Partha Sarathi Dutta and Sandip Sen. Forming stable partnerships. *Cognitive Systems Research*, 4(3):211–221, 2003.

[5] B. Esfandiari and S. Chandrasekharan. On How Agents Make Friends: Mechanisms for Trust Acquisition. In *In Proceedings of the ICMAS-01 Workshop on Deception, Fraud and Trust in Agent Societies*, pages 27–34, 2001.

[6] M. Nowak and R.M. May. The spatial dilemmas of evolution. *International Journal of Bifurcation and Chaos*, 3:35–78, 1993.

[7] Sabyasachi Saha, Sandip Sen, and Partha Sarathi Dutta. Helping based on future expectations. In *Proceedings of AAMAS-03*, pages 289–296.

[8] Sandip Sen. Reciprocity: a foundational principle for promoting cooperative behavior among self-interested agents. In *Proceedings of ICMAS-96*, pages 322–329, 1996.

[9] Maggs Sripanidkulchai and Shang. Efficient content location using interest based locality in peer-to-peer systems. In *InfoCom 2003*, 2003.

[10] Pinar Yolum and Munindar P. Singh. Emergent properties of referral systems. In *Proceedings of AAMAS-03*, pages 592–599. ACM Press, 2003.