# Forming stable partnerships

Partha Sarathi Dutta and Sandip Sen
Mathematical & Computer Sciences Department
University of Tulsa
Tulsa, OK 74104
partha@euler.mcs.utulsa.edu, sandip@kolkata.mcs.utulsa.edu

**Abstract**

Autonomous agents interacting in an open world can be considered to be primarily driven by self interests. In this paper, we evaluate the hypotheses that self-interested agents with complementary expertise can learn to recognize cooperation possibilities and develop stable, mutually beneficial partnerships that is resistant to exploitation by malevolent agents. Previous work in this area has prescribed a strategy of reciprocal behavior for promoting and sustaining cooperation among such cognitive learning agents. We develop on that work by expanding the task cost metric to include both time of completion and quality of performance. Different task types are assumed and, in contrast to previous work, we use heterogeneous agents with varying expertise for different task types. This necessitates the incorporation of cognitive abilities, including an understanding of one's own capabilities and learning about other's capabilities within the reciprocity framework.

## 1 Introduction

Agent-based systems provide an effective paradigm for developing real world applications like electronic commerce, recommender systems and personal assistants. Agents deployed in these applications often interact in an open environment with other agents or humans [3, 4, 5, 10]. These interactions involve cooperation, collaboration or competition for resources to achieve specified goals of these agents. With increase in the complexity of agent interactions, it is imperative that the behavioral characteristics of agents acting in a group should be studied in detail and suitable interaction strategies developed that optimize system performance. Agents, in this context, should be considered as cognitive entities and multi-agent interactions should be studied with agents using elements of cognition like perception, modeling and learning.

We have been interested in agent strategies for interactions with other agents that can promote cooperation in groups of self-interested agents. Our approach is different from other researchers who have designed effective social laws that can be imposed on agents [14]. We assume that typical real-world environments abound in *cooperation possibilities*: situations where one agent can help another agent by sharing work such that the helping cost of the helper is less than the cost saved by the helped agent. In systems with frequent interaction possibilities, agents can benefit by exchanging help and developing stable, mutually beneficial relationships. As agent system designers, we can define rules of interactions that enable the agents to learn to benefit from such cooperation possibilities. We prescribe reciprocative behavior as a stable, sustainable approach that creates, promotes and sustains cooperative groups in such environments. This approach not only encourages group formation similar, self-interested agents, but also helps ward off exploitative tendencies of selfish agents in the society [2]. Reciprocal behavior becomes more viable and stable if the helping agent takes into account the opinion of other agents before extending any favor [12].

A restrictive assumption of the previous work on reciprocity was the incorporation of only a single cost metric, time, used by the agents. In real-life scenarios, agents have to be cognizant of multiple objectives like time, quality, dependability, etc. when evaluating the benefit of interaction with another agent. As a first step to handling such scenarios, we expand on the set of cost metrics by including both time and quality in an agent's evaluation scheme. The *time* attribute refers to the absolute time units required for completing a particular task, and the *quality* attribute is a measure of the effectiveness of executing a task. Agents are assumed to have different expertise over different task types. The time and quality of performance of an agent working on a task will depend on the expertise level of that agent on that particular task type.

A second restriction in the previous work was the explicit assumption that all agents had identical capabilities, i.e., agents were homogeneous in their expertise over different task types. We assumed this to focus on the design and evaluation of help-giving behaviors of the agents. Having established a basic competence of probabilistic reciprocity based agents in recognizing mutually cooperative relationships and effectively neutralizing exploitative behavior [12], we now turn our attention to the practical problem of variation in agent expertise. We assume that different agents have different skill sets, which make them more effective in accomplishing some tasks compared to others. We endow our agents with cognitive capabilities for perceiving and utilizing cooperation possibilities and learning the capabilities of themselves and others through repeated task performance and interaction.

The goal of this work is to evaluate whether self-interested agents can learn to recognize agents with complementary expertise and develop a self-sustaining relationship through exchange of help. This requires agents to have a memory of the history of interactions with other agents. The current work can be described as an augmentation of the basic probabilistic reciprocity model with the concept of selecting partners. Additionally, such help exchange procedures must be resistant to exploitation by malevolent agents who do not reciprocate help-giving actions. Our hypothesis is that when combined with cognitive characteristics like appropriate learning schemes, probabilistic reciprocity based strategies will enable the development of stable, mutually beneficial groups of self-interested agents with complementary skill sets.

# 2 Reciprocity

Our probability-based reciprocal behavior is designed to enable an agent make decisions about responding favorably to or refusing a request for help from another agent. A reciprocal agent, $r$, decides to help or not with some probability that depends on the history of interaction with the agent $h$ asking help. The probability with which $r$ helps $h$ increases as the net benefit earned by $r$ over all past interactions with $h$ increases, i.e., $r$ is more inclined to help someone from whom it had received more help in the past. This is the basic hypothesis of the reciprocity principle. We represent that benefit in terms of the *balance* that $r$ has with $h$. While helping, an agent incurs a cost (in terms of time or other resource spent). This is reflected as a decrease in the balance that the helping agent has with the helped by an amount equal to the incurred cost. In contrast, if an agent receives help, it saves the cost it would have incurred doing the task on its own. This saving is recognized by an increase in balance that it has with the helping agent. From the above discussion we can infer that higher the balance that $r$ has with $h$, higher is the probability of $r$ helping $h$. We also wanted our decision procedure to allow the agents to incur some initial cost to identify possibly beneficial relationships. For example, at the outset all agents will have zero balances with other agents. If every agent conservatively avoids giving help at this stage, then cooperation possibilities in the environment will not be availed. So, we want our agents to be willing to risk incurring a moderate amount of initial cost to help other agents with the hope that such costs will be later recuperated many times over by other agents reciprocating such behavior.

Now we present the actual probability function that is used by our agents. The probability that agent $k$ will help agent $i$ to do task $j$ is given by

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

where $C^k_{avg}$ is the average cost of tasks performed by agent $k$; where $OP_i$ is a measure of the balance that other agents have with agent $i$, (the role of this parameter is further discussed in Section 3.2); $C^k_{ij}$ is the cost of agent $k$ to do the task $j$ of agent $i$; $\beta$ and $\tau$ are the only two constant parameters used in the function, where $\beta$ is used to set the cost an agent is ready to incur to help an unknown agent with the hope of initiating a long-term cooperative relationship and $\tau$ is used to set the shape of the probability curve. This is a sigmoidal function where the probability of helping increases as the balance increases and is more for less costly tasks. Note, that with initial opinion of zero, an agent has a 0.5 probability of helping another agent where the cost of helping is $\beta * C^k_{avg}$. We include the $C_{avg}$ term because while calculating the probability of helping, relative cost should be more important than absolute cost. The term $OP_i$ is calculated from individual balance of past help, $B_{ki}$, that agent $k$ has with agent$i$, due to the previous helps given to and received from agent $i$. We will define We define $B_{ki} = S_{ki} - W_{ki}$, where $S_{ki}$ and $W_{ki}$ are respectively the savings obtained from and extra cost incurred by agent $k$ from agent $i$ over all of their previous exchanges (note that, in general, $B_{ik} \neq -B_{ki}$). From Equation 1 we see that as the cost of requested help increases and the opinion of an agent decreases, that agent is less likely to receive help from an another agent. But as the opinion of an agent in a group improves (which happens if that agent has a net positive balance with other agents), reciprocative agents will be more inclined to help that agent for the same helping cost.

In this framework, when agent $i$ asks for help from agent $k$ for its $j^{th}$ task, a cooperation possibility occurs if $C^k_{ij} < C^i_j$, where the right hand side of the inequality is agent $i$'s cost for task $j$. This is equivalent to saying that the cost incurred by the helper agent in doing the task of the asking agent is less than what the asking agent incurs itself. We assume that the cost incurred by an agent in doing a task type in which it is an expert is less than that for other task types. Hence, it is more likely that an agent $i$ asking for help from agent $k$ for a task of type $x$, will satisfy the cooperation possibility condition and get help if $i$ is a non-expert on task type $x$ while $k$ is an expert. If agents have varying expertise over different task types and interact over extended periods, it is reasonable to conjecture that agents with dissimilar expertise will cooperate with each other more often if there are sufficient tasks of all types. This would lead to stable and mutually beneficial group-relationships among agents with complementary capabilities.

In this paper we aim at extending the basic probabilistic reciprocity framework. We evaluate the importance of "quality of performance" as another cost metric and the applicability of learning other agents' capabilities in developing stable, cooperative partnerships among self-interested agents.

# 3 Problem domain

We evaluate our hypothesis using simulations in a task completion problem domain. In the following we enumerate the characteristics of our domain and delineate the major steps in the simulation.

- Total number of agents = $N$.
- Each agent $i$ has to complete $T$ tasks.
- There are $k$ different task types.
- Each agent $i$ is an"expert" in one of the task types and a "non-expert" in the others. Being an "expert" in a task type "x", an agent "i" can complete tasks of type "x" with less time and high quality.
- Time to complete and the quality of performance for a task of type "x" by an agent "i" are drawn from different Normal distributions depending on its expertise on "x".
  - If "i" is an expert in task type "x" then, $\mathtt{time} \in \mathtt{N}(\mu_{\mathtt{high}}, \sigma)$ and $\mathtt{quality} \in \mathtt{N}(\mu_{\mathtt{low}}, \sigma)$.
  - If "i" is not an expert in task type "x" then, $\mathtt{time} \in \mathtt{N}(\mu_{\mathtt{low}}, \sigma)$ and $\mathtt{quality} \in \mathtt{N}(\mu_{\mathtt{high}}, \sigma)$.
- The cost incurred by an agent "i" to complete a task of type "x" is calculated in terms of the "time" and "quality" metrics. $\mathtt{cost} \propto \frac{\mathtt{time}}{\mathtt{quality}}$, where `time` and `quality` are drawn from distributions as mentioned before.

Each agent is assigned the same number of tasks at the start of the simulation. At this point the agents ask for help from one another. When help is given, the agent who is helped updates its estimate of the helper agent's expertise, i.e., the time and quality of performance of the helper agent. With more interactions, therefore, agents develop better models of the capabilities of other agents. This biases their help asking behavior - for a given task type, an agent is more likely to ask help from those agents who are expected to produce higher quality results in less time. We have used the strategy where an agent selects another agent randomly for the first "F" tasks it has to complete. This allows it to explore and build estimates of the capabilities of other agents by interacting with them. Ideally, "F" should be chosen such that within its first "F" tasks, an agent is likely to interact with every other agent for at least one task of each type (out of a total of $K$ types). We attempt to achieve that by selecting an agent randomly to ask for help as well as by distributing uniformly the task types in the agents' initial task set. Hence, for the remaining $T-F$ tasks, the agent can use its estimates of other agents about their time to complete and quality of performance of every task type and select preferentially on the basis of these estimates [1]. For each task type, an agent completes the first task of that type that it gets as its current task. This is to ensure that each agent has a starting, albeit preliminary, model of its own capabilities. The modeling of other agents' time and quality metrics are better with more interactions, and hence, with more tasks to complete. The different activities described so far have been summarized in the following pseudocode.

```
Global:
   A <- set of N agents

//methods used to simulate an agent's (member i of the set A) behavior.
main()
{
  //assing T tasks to agents i.
  tasks_i <- {task_i1, ... ,task_iT}; // where type(task_ij) in {1, ...,K}
                                   //            for all j in 1, ...,T

  //for each task in bag, do the following.
  while(|tasks_i| > 0)
  {
    currtask_i <- selectCurrentTaskFrom(tasks_i);

    if(firstTimeEncountered(type(currtask_i))
    {
      completeTask(currtask_i); //complete task on its own
      updateEstimate(i);      //update own estimates of time and quality
    }
    else
    {
      helped <- false;
      if(taskNo < EXPLORE_RANGE) //explore
      {
        while(!helped and !allAgentsVisitedIn(A\i))
        {
          j <- selectAgentRandomlyFrom(A\i);  //randomly choose an agent to ask
          helped <- askForHelp(i,j,currtask_i);  //ask for help for the current task
```

[1] It should be noted that agents ask help only for tasks that were assigned initially and not for those they acquire while accepting another agent's request for help

```
        }
      }
      else  //exploit using the learned models of other agents
      {
        //sort agents in order of their estimated costs for the current task type.
        sortAgentsInOrderOfCosts(type(currtask_i));
        while(!helped and !allAgentsVisitedInOrderFromSortedList)
        {
          j <- selectAgentFromSortedList;
          helped <- askForHelp(i,j,currtask_i);
        }
      }

      if(!helped)
      {
        completeTask(currtask_i); //complete task on its own
        updateEstimate(i);      //update own estimates of time and quality
      }
      else
        updateEstimate(j);      //update helper agent's time and quality estimates
    }
    remove currtask_i from tasks_i;
  }
}//end main()

boolean askForHelp(i,j,currtask_i)
{
  if(cooperationPossibility(i,j,currtask_i)) //if cost incurred by j to complete
                                             //currtask_i is less than cost
                                             //incurred by i
  {
    pr <- probabilityOfHelp(i,j,currtask_i); //probability of j helping i
                                             //calculated using Equation 1
    if(rand() < pr)
    {
      updateBalance(i,j,currtask_i);  //i records help from j
                                      //j records helping i
      return true;
    }
    else
      return false;
  }
  return false;
}
```

**Agent pseudocode:** The pseudocode for a single agent is presented. The entire simulation runs until bags of all agents become empty.

The simulation ends when every agent has finished all of their assigned tasks, either by doing the tasks or by getting other agents to help with some of the tasks. If an agent is not an expert in a particular task assigned to it, it will try to get help from another agent, who is an expert in that particular task type. But to convince the other agent to help with a task, our agent would have to prove itself to be useful to that agent. To achieve this, agents have to form models of their own performance and that of others for different task types.

## 3.1 Partner selection and partnership formation

We have incorporated simple reinforcement learning schemes to update agent performance estimates. Agents have estimates of their own abilities to do the different task types. Estimates are of two types: *time estimate*, which reflects the expected time for completing the task, and *quality estimate*, which reflects the expected performance level of an agent for that task. Agents also keep estimates of every other agents' abilities.

The time and quality estimates are used by the agents to compute the cost of processing a particular task. In the current formulation, the cost of completing a task is directly proportional to the time and inversely proportional to the quality of performance. For a given task type, an expert takes less time to produce higher quality work than a non-expert. Hence, the design of the cost function in the above manner ensures that the cost of completing a task of a certain type is less for an expert in that type compared to a non-expert. When asking for help, agents compute the cost $C1$, it expects to incur by doing that task itself. The estimated cost $C2$ that a prospective helping agent incurs for that work is also computed. Help is obtained only if $C2 < C1$. This condition corresponds to a "cooperation possibility".

Initially, agents have neutral estimates about their own abilities and that of other agents. To obtain accurate estimates about their own abilities, agents must themselves perform tasks of different types. When an agent performs a task, it requires a certain time and achieves a certain quality of performance. These values are used by the agents to estimate its own expertise level. When an agent helps another, the helped agent updates its estimate of the helper using the time and quality of performance values that the helper agent requires to complete the particular type of task. We use a simple reinforcement scheme to update the time and quality estimates is given below:

$$t_{ij}^{n+1} \leftarrow (1-\alpha)t_{ij}^{n} + \alpha t_{ij},$$

where $t_{ij}^{n}$ is the time taken by agent $i$ to do task $j$ on the $n^{th}$ time instance.

We use a similar update policy for the quality of performance ($q_{ij}$). More interactions between the same pair of agents provide each with the opportunity of developing more accurate models of one another. This information is used to select an agent for asking help. As a consequence of developing the learned model of others' abilities, an agent $i$ who specializes in task type $T_1$ when given a task of type $T_2$, where $T_1 \neq T_2$, will decide in favor of asking help from an agent $j$ specializing in task type $T_2$, rather than agents who specialize in other task types. The knowledge that agent $j$ is capable of delivering the task with lesser time and with higher quality is obtained from the reinforced models that the asking agent has developed over time. As a consequence, agents with complementary expertise interact with each other more often than agents with similar expertise. Thus, given sufficient interaction opportunities and cooperation possibilities, agent partnerships develop where self-interested agents have complementary skill sets.

### 3.2 Agent types

The characterize the behavior of reciprocative agents by the level of confidence they employ on other agents. We mentioned that agent $A$ can receive help from agent $B$ only if there is a cooperation possibility, i.e., the cost incurred by $A$ to do that task is more than incurred by $B$ for helping $A$. However, in deciding whether to help or not, a reciprocative agents can ask *some* other agents for their *opinion* about the agent asking for help. The agents from whom a help-giving agent $x$, asks for opinion are only those with whom $x$ has a favorable balance of help. It is a reasonable policy to believe in those who have *earned trust* based on past interactions. The opinion received by $x$ from another agent $y$ is the cumulative balance that $y$ has with the agent seeking help. Higher the balance, more is the value of the opinion metric. A raise in the opinion value increases the probability of helping (refer to Equation 1. An agent who has a favorable record of help giving behavior with the agent giving opinion, therefore, is more likely to be helped. Reciprocative agents who decide to help or not based on such collaborative opinion are called *earned trust based reciprocative agents*. While calculating the probability of helping agent $i$, a learned-trust based reciprocative agent $k$ uses calculates the $OP_i$ term in Equation 1 as $\sum_{j \neq i \wedge B_{kj} > 0} B_{ji}$.

Selfish agents by definition do not extend help under any circumstances. The selfish agents that we use in our simulations are characterized by the additional property of *lying* when they are asked for their opinion. They lie about the balance they hold at any time with the agent who is currently waiting to receive help. If agent $x$ is asking agent $y$ whether or not to help agent $z$, then $y$ reports a negated balance with $z$ if $y$ is a liar. This means that liars "bad-mouth" helpful agents. When such an agent, $j$, is asked for its balance with another agent $i$, it reveals $B'_{ji}$ as

$$\begin{aligned} B'_{ji} &= (-B_{ji}), \text{ when } B_{ji} > 0 \\ &= B_{ji}, \text{ otherwise.} \end{aligned}$$

## 4 Experimental results

To examine our claim that the ability to estimate other agents' expertise enhance the performance of agents and boost partnership formation among agents with complementary expertise, we ran a series of experiments. In our experimental setups we have used three different task types. The "high" and "low" mean values are used to define the distributions from which we draw time and quality of performance of experts and non-experts are taken to be 10.0 and 1.0 respectively, with the standard deviation being 0.05.

We have considered two task cost metrics, time of completion and quality of performance. Our first experimental setup focuses on examining the relative performances of the reciprocative and selfish agents on those two cost metrics. We record the average time taken and the average quality of performance of the reciprocative and selfish agents for completing all tasks. We ran experiments with a total of 100 agents with half reciprocative and half selfish

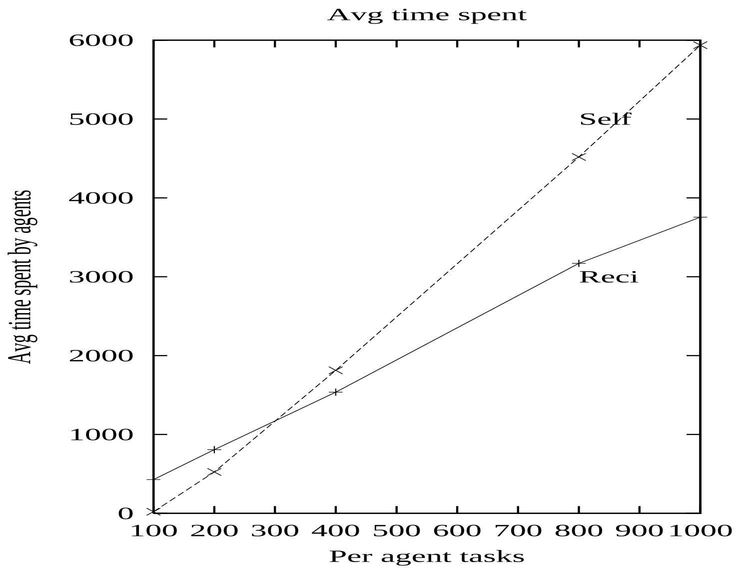


Figure 1: Average time spent by agents.

in nature. The tasks per agent was varied over 100, 200, 400, 800 and 1000 in different sets of experimental runs and the results in each set are averaged over 10 runs. The average time taken and the average quality of performance to complete all tasks for each agent type were recorded. Figures 1 and 2 show the results.

Figure 1 shows the relative performance of individual lying selfish and earned-trust based reciprocative agents in terms of the average time spent in performing all tasks. We see from the graphs that selfish agents outperform reciprocative (take lesser time to complete tasks) until the per agent task reaches a value of 300. As the task value is increased, reciprocative prove to dominate the performance of selfish agents by being to complete all tasks in lesser time. As noted earlier, the increase in the number of tasks allow the reciprocative to have more interactions with other agents. This, in turn, helps in their learning process of identifying potential helpful partners, and in particular, those with non-selfish behavior and having complementary expertise and detect the exploiters. Thus, when coupled with the strategy of earned trust that the reciprocative employ, i.e., rely on the opinion of only those agents who have been helpful in the past, learning leads to dominating performance in most cases.

Figure 2 shows the quality of performance of the same set of 100 agents with equal number of earned-trust based reciprocative and individual lying selfish. The graphs follow a pattern similar to that in figure 1. Reciprocative, initially fail to extract the advantage by learning other agents' models since the number of tasks is not large enough for learning to produce good models and using them to form fruitful partnerships. With increasing tasks, agents interact with agents more often which help them better identify others' expertise and learn more robust models of other agents. Hence, with increasing tasks reciprocative can get their tasks completed at a higher quality by obtaining help from partners of complementary expertise in emerging partnerships.

In the results presented so far we have attempted to verify our claim that agents do

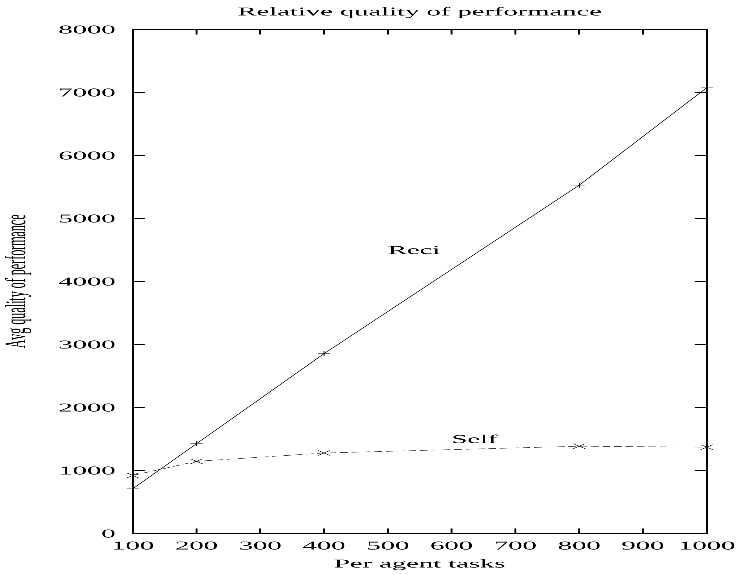


Figure 2: Average quality of performance of agents.

learn to identify partners with complementary expertise by examining and explaining the consequences or external effects of the strategies. In our next experimental setup, we show the results that fully establishes the fact. We present, in this experimental setup, information regarding the group/partnership formation among agents. Stability of a partnership is determined by the frequency of interaction and the net savings earned by such partnerships.

Table 1 shows the number of helps (interaction frequency) and total savings generated by the different groups or partnerships of agents of different expertise in a population of size 30 of earned trust based reciprocative with 2000 tasks per agent. The interaction frequency of a "group" (like $(1,1)$) is defined as the total number of interactions between all agents with expertise in task type 1. Since there are 3 task types, there can be a total of 6 groups that are enumerated in the table. The "savings" of such a group is the total savings earned over the interactions that define the group size. ¿From Table 1, we find that both the interaction frequencies and savings earned by the groups of agents having complementary expertise are larger than the corresponding values of groups of similar agents. For example, looking at groups $(3,3)$ and $(2,3)$, we find that the group size in the $(2,3)$ case is almost three times larger than in the $(3,3)$ case. Similarly, the savings earned by the $(2,3)$ group is an order of magnitude larger than the savings earned by the homogeneous group $(3,3)$. This corroborates our hypothesis that when agents are endowed with the ability to learn others' expertise, they develop mutually beneficial partnerships with agents of complementary expertise. The benefit is particularly obtained in groups of heterogeneous agents, i.e. agents with varying expertise.

In the previous sets of experiments we have shown that reciprocative perform better on both the dimensions, time and quality, that define task cost. Savings earned by an agent gives in a nutshell how well it performed in accomplishing its assigned tasks. So, in our last set of experiments we studied the variations in average savings earned by both selfish and reciprocative agents after each agent has completed all of its tasks. We used a mix of

Table 1: Group interaction information of earned trust based reciprocative

| Group | Number of helps | Saving |
|---|---|---|
| (1,1) | 4470 | 2987.03 |
| (2,2) | 4409 | 3023.38 |
| (3,3) | 4584 | 3560.69 |
| (1,2) | 12946 | 130703.34 |
| (1,3) | 12744 | 128684.23 |
| (2,3) | 12976 | 135528.07 |

individual lying selfish and earned-trust based reciprocative agents. Figures 3 and 4 show the results obtained by varying the percentage of selfish agents in the population under two different values of "tasks per agent", 20 and 40 respectively. The percentage of selfish agents in the population was varied from 50% to 80% in steps of 10%. In both the figures we find reciprocative agents outperform selfish agents, because the average savings earned by the reciprocative agents are more than those of the selfish agents for all selfish percentages in the population. In both cases the savings earned by reciprocative decrease with increase in the percentage of selfish population. For the same percentage of selfish population, however, the savings earned by the reciprocative are higher with more tasks per agent. From these observations, we can conclude that, over sustained interactions, reciprocative agents can learn to identify potential helpful agents and their expertise, which enables them to enter into mutually beneficial partnerships. This means that if the group composition stays the same or changes slowly, agents can significantly benefit from learning about other agents' abilities and help-giving attitudes and using those to make help-giving decisions. Savings earned, reflect the effectiveness and stability of agent partnerships. Lower values of savings earned by the selfish agents indicate that the reciprocative agents have effectively shunned the exploitative selfish agents. The performance of reciprocative agents, however, decreases with increase in the number of the selfish agents. This is due to the fewer reciprocative agents with which stable mutually beneficial partnerships can be formed. With increase in the task level, however, the savings of reciprocative agents for the same number of selfish agents in the population are more because a higher number of tasks per agent allows agents to receive help from few reciprocative agents for a much longer time. Therefore, with 60% selfish population, the average savings of reciprocative is about 76 when there are 20 tasks per agent and about 320 with 40 tasks per agent.

# 5 Related work

Using agent partnerships is an interesting and much explored approach to solve complex, distributed tasks. Assigning groups of agents to do a task or multiple tasks has the advantage of complementary individual agent expertise being used to complete different parts of the global problem.

However, work in the area of partnership formation in agent societies has focused on

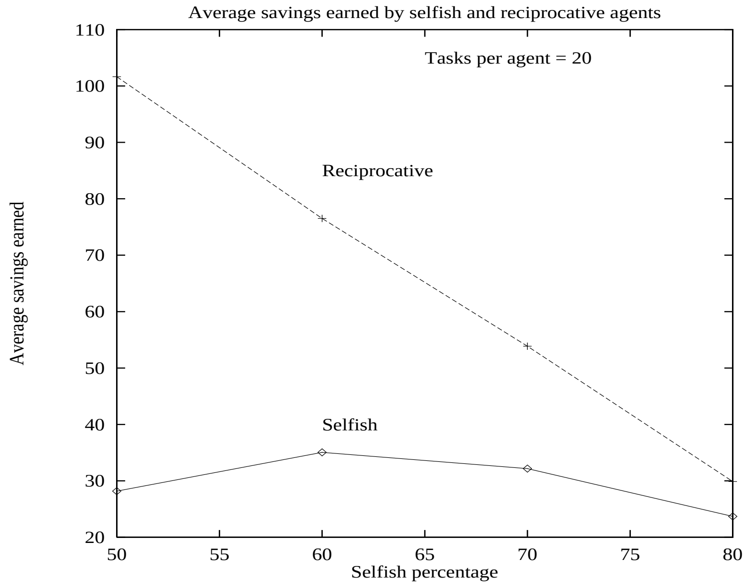


Figure 3: Average savings of agents with 20 tasks per agent.

cooperative agents [13]. A related work on partnerships in agent societies takes self-interested agents into account [11]. But it does not consider the possible heterogeneity in performance of a task between different agents. Our work is different from these because it takes into consideration self-interested agents that have different performance levels for different task types. Additionally, we have used a cognitive component for our agents, the capability of learning other's expertise. Development of such learned models about other agents' expertise engenders a favorable bias towards those with complementary expertise when asking for help. Mutual understanding of capabilities evolve cooperative groups in the society of self-interested agents. Learning of cooperative behavior has also been addressed in [7]. However, there the learning uses an off-line learning module which generates situation-action pairs using a genetic algorithm. To determine the fitness of the evolved situation-action pairs during off-line learning an agent-model is required. Our work uses only online learning and agents do not have to store a priori models of other agents. However, in an attempt to learn behaviors of all other agents, our approach can become time consuming in situations when there are a large number of agents in the environment.

Though psychologists and socio-economists have widely studied individual cognitive behaviors [1, 8], evaluating the influence of cognitive components like memory and learning on the evolution and profitability of agent behaviors has been relatively scarce in the multi-agent systems community. An integrated approach, combining the concepts of social science, cognitive science, and multi-agent systems to elucidate the characteristics of cognitive agents and their capacities is presented in [15]. The impact of socio-cultural influences on individual agent cognition has been the main focus of that work. That behaviors like cooperation and competition are based on an agent's individual sociality has been advocated in [6]. In [9], a decision framework for socially conscious agents has been demonstrated where agents tune their behaviors depending on resource availability and social responsibilities. Therefore, agents in these strands of research can be identified as "socially self-interested". In our

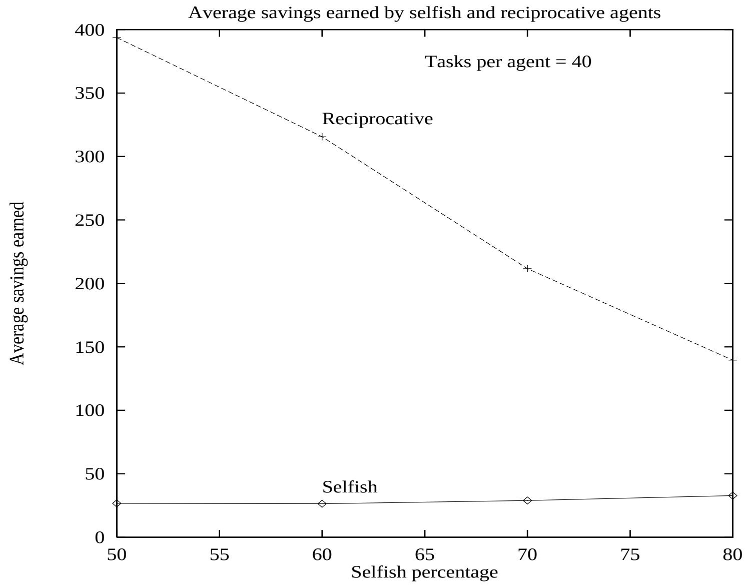


Figure 4: Average savings of agents with 40 tasks per agent.

work, however, we study the effects of cognitive aspects like learning to recognize skills, on the individual profitability of inter-agent interactions and stability of agent partnerships.

# 6 Conclusions and future directions

Adoption of reciprocal behavior in a society of self-interested agents generates incentives that promote cooperative behavior. In this paper we hypothesized that in an agent society where agents differ in their expertise, incorporation of cognitive behavioral components, like learning, will help generate cooperative groups of agents with complementary expertise. Our simulations confirmed this hypotheses. These groups are mutually beneficial to the agents because they save more time by giving the tasks to other agents who have the pertinent expertise and accomplish their tasks with better quality. Such cooperative group formation is stable since exploitative agents are effectively shunned.

Agent behaviors can change over time given changing perceptions. One assumption in this work has been that agents have fixed behaviors. A more realistic scenario would be for an agent to have the freedom of changing its behavior when it deems appropriate. Such behavior adoption leads to an evolutionary process with a dynamically changing composition of agent group behaviors. We plan to investigate whether such evolving agent behavior might create a society of optimally co-adapted agents. It would also be interesting to study the effect on the partnerships evolving among agents when the distributions of tasks of different types are non-uniform, i.e., when agents have different number of tasks of different types.

**Acknowledgments** This work has been supported in part by an NSF CAREER award IIS-9702672.

# References

[1] A. Barnes and P. Thagard. Emotional decisions. In *Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society*, pages 426–429, 1996.

[2] Anish Biswas, Sandip Sen, and Sandip Debnath. Limiting deception in groups of social agents. *Applied Artificial Intelligence, Special Issue on Deception, Fraud, and Trust in Agent Societies*, 14(8):785–797, 2000.

[3] Jeffrey M. Bradshaw. *Software Agents.* AAAI Press/The MIT Press, Menlo Park, CA, 1997.

[4] Communications of the ACM, july 1994, volume 37, number 7, 1994. Special Issue on Intelligent Agents.

[5] Communications of the ACM, march 1999, volume 42, number 3, 1999. Special Issue on Multiagent Systems on the Net and Agents in E-commerce.

[6] Christiano Castelfranchi. Modeling social actions for ai agents. *Artificial Intelligence*, 103:157–182, 1998.

[7] Joerg Denzinger and Micheal Kordt. Evolutionary on-line learning of cooperative behavior with situation-action-pairs. In *Proceedings of Fourth International Conference on MultiAgent Systems, ICMAS 2000*, pages 103–110, Los Alamitos, CA, 2000. IEEE Computer Society.

[8] Bruce Edmonds. Modeling socially intelligent agents. *Applied Artificial Intelligence*, 12:677–699, 1998.

[9] Lisa Hogg and Nicholas R. Jennings. Variable sociability in agent-based decision making. In *Proceedings of the Sixth International Workshop on Agent Theories Architectures and Languages (ATAL-99)*, pages 276–289, 1999.

[10] Michael N. Huhns and Munindar P. Singh. *Readings in Agents.* Morgan Kaufmann, San Francisco, CA, 1997.

[11] K. Lerman and O. Shehory. Coalition formation for large-scale electronic markets. In *Proceedings of the Fourth International Conference on Multi-Agent Systems*, pages 167–174, 2000.

[12] Sandip Sen, Anish Biswas, and Sandip Debnath. Believing others: Pros and cons. In *Proceedings of the Fourth International Conference on Multiagent Systems*, pages 279–285, Los Alamitos, CA, 2000. IEEE Computer Society.

[13] Onn Shehory and Sarit Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence Journal*, 101(1-2):165–200, 1998.

[14] Yoav Shoham and Moshe Tennenholtz. On the synthesis of useful social laws for artificial agent societies (preliminary report). In *Proceedings of the National Conference on Artificial Intelligence*, pages 276–281, San Jose, California, July 1992.

[15] Ron Sun. Cognitive science meets multi-agent systems: a prolegomenon. *Philosophical Psychology*, 14(1):5–28, 2001.