Robust Agent Communities

Sandip Sen, Sabyasachi Saha, Stéphane Airiau, Teddy Candale,
Dipyaman Banerjee, Doran Chakraborty, Partha Mukherjee and Anil Gursel

Department of Mathematical & Computer Sciences
The University of Tulsa, USA
sandip@utulsa.edu

Abstract. We believe that intelligent information agents will represent their users interest in electronic marketplaces and other forums to trade, exchange, share, identify, and locate goods and services. Such information worlds will present unforeseen opportunities as well as challenges that can be best addressed by robust, self-sustaining agent communities. An agent community is a stable, adaptive group of self-interested agents that share common resources and must coordinate their efforts to effectively develop, utilize and nurture group resources and organization. More specifically, agents will need mechanisms to benefit from complementary expertise in the group, pool together resources to meet new demands and exploit transient opportunities, negotiate fair settlements, develop norms to facilitate coordination, exchange help and transfer knowledge between peers, secure the community against intruders, and learn to collaborate effectively. In this talk, I will summarize some of our research results on trust-based computing, negotiation, and learning that will enable intelligent agents to develop and sustain robust, adaptive, and successful agent communities.

1 Introduction

Humans are often referred to as social animals. What this implies is that societal relations and interactions are important and essential in our lives. Various modalities and temporal horizons of social interactions enrich our life and supports our material and spiritual pursuits. As agent researchers, we envision autonomous, intelligent agents as augmentations of our natural selves that can relieve us of some of our chores and responsibilities. While some of these tasks can be achieved individually, without input from others, a significant proportion of such tasks would require our agents to interact with agents of our peers or other members of our society. Such interactions may involve sharing of resources, give-and-take of information and expertise, collaborative efforts for a common cause, etc. Just as we humans thrive in our societies, these intelligent agents will contribute and benefit from artificial agent societies. Hence, *agents must be social entities.* Until that happens, our agents will necessarily be limited in potential, bereft of the opportunities provide by the social scaffolding that we enjoy, and constrained by the limited local resources, knowledge, and capabilities.

But before the vision of a vibrant, productive agent society becomes a reality, significant research investments in codifying and formalizing social protocols, norms, and mechanisms must be made. Our MultiAgent SysTEms ReSearch (MASTERS) research group at the University of Tulsa have been studying, for more than a decade, mechanisms, algorithms, and techniques to foster social collaboration and coordination mechanisms ranging from negotiation schemes to trust-based reasoning, predictive and proactive coordination protocols to multiagent learning algorithms. In particular, we emphasize the value of a stable environment for agents where they choose relationships to enter in, resources to share, commitments to make and keep, agents to interact with, etc. While monetary transactions are necessary for one-time interactions or in unstable societies with significant flux in agent compositions, our work focuses on environments where agents are resident for significant time periods and hence interact repeatedly with the same agents. Thus we can utilize rich interaction modalities and principles including trust, belief, context, history, future expectations, shared values, etc. that allow for some of the same richness of interaction that we enjoy in human societies. We are primarily interested in leveraging history of interactions to identify trustworthy partners to engage in long-term relationships. We are actively working on developing principles and mechanisms that allow agents to build and sustain communities. Our vision is that of robust agent communities that form, grow, and flourish from proactive collaboration, that benefit from each other's resources and expertise, that encourages active cooperation to develop, sustain, secure, and enrich its inhabitants.

In this invited talk, I present representative results from some of these investigations. As it is not possible to do justice to all of these research, I have chosen to highlight the primary motivation, technical approach and summary contribution from a select few recent research efforts. I refer the reader to our website (`http://www.mcs.utulsa.edu/~sandip`) for a list of our papers which would provide a more thorough and representative overview of our research.

We will overview the following subset of our research areas:

- In Section 2 we present results from our work on trusted, reciprocal relationship maintenance in agent communities and its applications in P2P networks.
- In Section 3 we present a multiagent learning scheme that solve well-known social dilemma problems like the Prisoner's dilemma,
- In Section 4 we present a framework by which one agent in the community can teach classification knowledge to another agent without knowledge of the latter's knowledge representation or learning algorithms.
- In Section 5 we present results on emergence of social norms when populations of agents interact repeatedly,
- In Section 6 we outline a study that involves the use of simple learning strategies to identify optimal partnerships in a large population.
- In Section 7 we overview a protocol for negotiating fair and efficient allocation of multiple, indivisible resources.
- In Section 8 we present strategies that allow agents to bid for bundles of items from concurrent auctions selling individual items.

• In Section 9 we present techniques that allow agents to share referrals about service providers that allow the agent community to settle in states where all agents are satisfied with their current service provider.
• In Section 10 we outline techniques for improving security and data integrity in sensor networks by detecting malfunctioning nodes.
• In Section 11 we summarize a probabilistic reasoning mechanism for detecting deviations from team plans by a team member.

2 Reciprocal relationships

Our work on reciprocal relationships allow self-interested agents to leverage complementary expertise in stable agent communities. An agent can ask for help for a task from an expert in that task type. The help from the expert saves the requesting agent significant cost which is greater than the cost incurred by the expert agent while helping. In another situation, the role of the helping and the helped agents may be reversed. Thus, if the environment has sufficient cooperation possibilities, self-interested agent will find it prudent to enter into reciprocal relationships. Our work allows agents to develop, nurture, and sustain such relationships while avoiding exploitative agents who receive help but do not reciprocate.

In our early work on reciprocity [1, 2], we had a restrictive assumption that agents have fixed strategies for deciding whether or not to help other agents. More specifically, agents were either reciprocative, selfish (never returned help), philanthropic (always helped) or individual (never gave or received help). Agents with specified strategies interacted repeatedly over a sustained period of time and their effectiveness was calculated as a function of the total cost incurred to complete all assigned tasks and the agents never changed their expertise.

A more realistic scenario would be to give an agent the freedom of choosing from one of several help-giving strategies and to change its strategy as dictated by the environmental conditions. An agent may be inclined to adopt a strategy if agents using that strategy is observed to be performing better than others. Such a strategy adoption method leads to an evolutionary process with a dynamically changing group composition of agent strategies [3]. In [4], we present a mathematical model to capture the dynamics of the agent population. Using this analytical model, one can predict the population distribution of future given the initial environmental settings. Given the initial strategy profile in the population and the assigned task load, we have constructed decision surfaces using which a rational agent can choose the most beneficial strategy for the long run.

2.1 Resisting Free Riding and Collusion in P2P networks

Peer-to-peer (P2P) systems enable users to share resources in a networked environment. P2P systems are vulnerable to problems including free-riding users who utilize resources without contributing in turn, collusion between groups of users to falsely promote or malign other users, and zero-cost identity problem that allows nodes to obliterate unfavorable history without incurring any expenditure. We used a reciprocity-based mechanism to tackle these problems [5]. We assume each node in the P2P network is managed by a self-interested agent. Our mechanism rewards an agent by enhancing its probability to receive resources it requested only when the node itself shares its own resources with other nodes in the system. This motivates nodes to contribute resources instead of free-riding.

In a P2P network, the tasks can be mapped into resources that an agent requires at a particular instance of time. Every agent has expertise in resource type $T \in \Upsilon$ where Υ is the set of all such resource types. Agents request resources of types in which they are not experts from other agents. The probability that an agent has a particular resource of a given type is much higher if an agent is an expert in that resource type than when it is not. An agent helps another agent if it provides a resource that is requested from it. Reciprocative agents return help, selfish agents try to free-ride. The expected utility of agent m for interacting with agent o requesting a resource type τ at time T is

$$E_T(m,o,\tau) = \sum_{t=T}^{\infty} \gamma^{t-T} [\sum_{x\in\Upsilon} (D_m^t(x) \mathrm{Pr}_{m,o}^t(x) cost_m(x)) - \sum_{x\in\Upsilon} (D_{o,m}^t(x) \mathrm{Pr}_{o,m}^t(x) cost_m(x))],$$

where $cost_i(x)$ is the expected cost that i incurs to procure a resource of type x by itself, γ is the time discount, and Υ is the set of different area of expertise. The evaluation of the expected utility of agent m helping agent o considers all possible interactions in future and for all types of resources. In the above equation, $D_m^t(x)$ is the expected future distribution of resource types that agent m may require at time instance t, and $D_{o,m}^t(x)$ is the expected future distribution of resource types that agent o may ask from m at time instance t. We define $\mathrm{Pr}_{i,j}^t(x)$ as the probability that agent j will share a resource of type x, when requested by agent i at time step t.

For a sufficiently large agent population, interaction between any two given agent may be infrequent, and it can take a long time to ensure enough interaction among agents to build up informative interaction histories. To alleviate this problem, we propose to use a reputation mechanism. In this reputation framework, when an agent m is asked for help by another agent o, m requests other agents, $\mathcal{C}$, who have interacted with o before to share their experiences about o. Upon request, the $\mathcal{C}$ agents report their complete interaction history with o to m. The helping agent, m, then uses this information to compute a more accurate probability of o's help-offering behavior for different resource types by weighing its personal experience with o and the average of the probabilities reported by the $\mathcal{C}$ agents. Therefore, the $\mathrm{Pr}_{m,o}^T(x)$ term in the previous equation is replaced by $\mathrm{Pr}_o^T(x)$, the reputation of o for providing help for task type x. $\mathrm{Pr}_o^T(x)$, is calculated as $\mathrm{Pr}_o^T(x) = (1-\alpha)\mathrm{Pr}_{m,o}^T(x) + \alpha \frac{\sum_{a\in\mathcal{A}-\{m,o\}} \mathrm{Pr}_{a,o}^T(x)}{|\mathcal{A}|-2}$ where $\mathrm{Pr}_{a,o}^T(x)$ is the opinion about o reported by a. These opinions are averaged from all agents except the interacting parties and the weight α on others opinion is an inverse function on the number of times m has asked o for help.

Agents, however, may collude to disrupt this mechanism by reporting good opinions about other colluders to third parties. We propose a Bayesian update scheme to discriminate between truthful and lying agent. In this approach, reciprocative agents assumes every one to be truthful and then uses a Bayesian update technique to judge the truthfulness of each agent based on its interaction experience with those about whom reputation was reported. Subsequently, the opinions reported by an agent is weighted by its estimated truthfulness. Experimental results showed that our mechanism effectively removes the free-riding, zero-cost identity and collusion problem in a P2P network.

2.2 Reciprocity between super-peers

Super-peer networks have been proposed to address the issue of search latency and scalability in traditional peer-to-peer (P2P) networks. In a super-peer network, instead of having a fully distributed systems of peer nodes with similar or comparable capabilities, some nodes that possess considerable computing power and resources are designated as super-peers. We address the problem of mutual selection by super-peers and client peers. In particular, we evaluate alternative decision functions used by super-peers to allow new client peers to join the cluster of clients under it. By formally representing and reasoning with capability and query distributions, we develop peer-selection functions that either promote concentration or diversification of capabilities within a cluster, and evaluate their effectiveness of different peer-selection functions for different environments where peer capabilities are aligned or are independent of their queries. Super-peers are responsible to find other peers which can provide an answer to a query, either by using peers from its pool of clients, or by requesting help from other super-peers. Our goal is to dynamically build the network of super-peers from a fully distributed network and ensure that peers are contributing to the community. Super-peers use a reciprocity mechanism to ensure that there are no free-riders in the system [6]. Each super-peer also ensures that all its client peers are contributing by enforcing load balancing within its cluster of client peers.

3 Learning to solve social dilemmas

Agents in a society are often confronted by social dilemmas. We formulate such social dilemma as general-sum game repeatedly played by self-interested agents and use a learning approach to solve such problems. Most exist learning mechanisms developed for game playing assume complete and perfect information, i.e., players can observe the payoffs received by all the players. This may not be possible in a large number of real environments and we assume players can observe the actions of all other players but not their payoffs. Rather than convergence to any Nash equilibrium strategy profile, we prefer Pareto-Optimal outcomes that also generate a Nash Equilibrium payoff (POSNE) for repeated two-player, n-action general-sum games. We introduce the Conditional Joint Action Learner

(CJAL) which learns the conditional probability of an action taken by the opponent given its own actions and uses it to decide its next course of action [7].

We assume repeated play of a stage game by a set S of 2 agents where each agent $i \in S$ has a set of actions A_i. We use the following notations: $E_t^i(a_i)$ is the expected utility of an agent i at iteration t for an action a_i, $Pr_t^i(a_i)$ is the probability that agent i plays action a_i at iteration t, and $Pr_t^i(a_j|a_i)$ is the conditional probability that the other agent, j, will play a_j given that the i^{th} agent plays a_i at iteration t. The joint probability of an action pair (a_i, a_j) at iteration t is given by $Pr_t(a_i, a_j)$.

A *CJAL* learner is an agent i who at any iteration t chooses an action $a_i \in A_i$ with a probability proportional to $E_t^i(a_i) = \sum_{a_j \in A_j} U_i(a_i, a_j) Pr_t^i(a_j|a_i)$, where a_j is the action taken by the other agent. These expectations are learned by using Q-learning:$E_t^i(a_i) = \sum_{a_j \in A_j} Q_t^i(a_i, a_j) * \frac{n_{t-1}^i(a_i, a_j)}{n_{t-1}^i(a_i)}$, where $n_t^i(a_i) = \sum_{a_j \in A_j} n_t^i(a_i, a_j)$ is the number of times agent i has played action a_i until iteration t and $Q_t^i(a_i, a_j)$, the estimated payoff from joint action (a_i, a_j) is updated after the $(t-1)th$ interaction as

$$Q_t^i(a_i, a_j) \leftarrow Q_{t-1}^i(a_i, a_j) + \alpha(U_i(a_i, a_j) - Q_{t-1}^i(a_i, a_j)),$$

where $0 < \alpha \leq 1$ is the learning rate.

We empirically show that under self-play and if the payoff structure of the Prisoner's Dilemma game satisfies certain conditions, a CJAL learner, using a random exploration strategy followed by a completely greedy exploitation technique, will successfully resolve the Prisoner's dilemma and produce cooperation. We also experimentally demonstrated the convergence of CJAL using limited exploration in self-play to POSNE outcomes on a representative testbed containing all structurally distinct two-player conflict games with ordinal payoffs. Though CJAL was not explicitly designed to optimize measures like social welfare, fairness (measured by the product of player payoffs) and success in converging to POSNE outcomes, it outperforms well-known existing multiagent learning algorithms like JAL and WOLF-PHC on these metrics.

4 Agent-Teaching-Agent (ATA)

Few researchers have addressed the problems of one, knowledgeable, agent teaching another agent. We investigate how an agent can use its learned knowledge to train a peer agent in its community with a possibly different internal knowledge representation. The knowledge being transferred is a concept description, a boolean-valued function that classifies input examples as members or nonmembers of the target concept. We assume that the trainer agent does not have access to the internal knowledge representation of the trainee agent, but can evaluate its concept recognition abilities by asking it to categorize selected exemplars and non-exemplars of the target concept. The trainer agent also do not have access to the original training set from which it learned its concept description. We have developed an Agent Teaching Agent (ATA) framework which focuses on incremental selection of training examples by the trainer to expedite the learning of the trainee agent [8, 9].

We envisage an iterative training procedure in which alternatively the trainer selects a set of training and testing exemplars, the trainee trains using the training set and then classifies the testing set, the trainer observes errors made by the trainee in classifying the instances in the last testing set and accordingly generates the next training and testing sets. This iterative process converges when error of the trainee falls below a given threshold. We now present these iterative training steps in an algorithmic form:

Procedure Train-Agent(Trainer,Trainee,Domain-Info){
Select initial training set N_0 and initial testing set T_0 from Domain-Info
$i \leftarrow 0$
repeat
  Train trainee agent on training set N_i
  Let M_i be the instances in T_i misclassified by trainee after training on N_i
  $T_{i+1} \leftarrow T_i \cup newTestInstances(M_i)$ and Domain-Info
  $N_{i+1} \leftarrow N_i \cup newTrainingInstaces(M_i)$
  $i \leftarrow i + 1$
until $|M_i| < threshold$}

This procedure needs to be fleshed out to realize an actual implementation. In particular, we have to specify procedures for selection of the initial training and testing sets, N_0 and T_0, and the generation of the next test set T_{i+1} based on the mistakes, M_i, made by the trainee on the current test set. We have developed these procedures for instance based and decision tree learners to work on problems with real-valued attributes. When selecting the initial training and testing instances, the goal is to select the most discriminating examples that help identify regions of the input space that do and do not belong to the target concept. For example, if a hyperplane separates instances of the target concept from non-instances, then points close to and on both sides of that hyperplane should be selected as initial training and testing set members. When selecting the next set of training and testing instances, the goal is to first isolate the mistakes made on the previous test set, and for each of these instances, find a few neighboring points, use some of them as part of the training data and the rest as part of the test data in the following iteration. The true classification of these points will not be known in general, and only their estimated classification, based on the concept description knowledge previously acquired by the trainer, can be used.

In our initial experiments with instance-based and decision tree learners as training and trainee agents we found that incremental training results in rapid improvement in classification performance of the trainee agent over the entire test set. The final accuracy is comparable to the accuracy of the trainer's knowledge or the trainee's knowledge if it had access to the original training set. Particularly interesting results were obtained with the Haberman data set obtained from the UCI repository: IB2, an instance-based learner, acting as a trainer can train C4.5, a decision-tree learner, to have better testing accuracy than itself. The trainer can, via the iterative training process, produce a more competent trainee! We have used this framework to train new agents joining a team of experts [8].

5 Social learning of norms

Behavioral norms are key ingredients that allow agent coordination where societal laws do not sufficiently constrain agent behaviors. Whereas social laws need to be enforced in a top-down manner, norms evolve in a bottom-up manner and are typically more self-enforcing. While effective norms and social conventions can significantly enhance performance of individual agents and agent societies and have merited in-depth studies in the social sciencethere has been little work in multiagent systems on the formation of social norms.

We have recently used a model that supports the emergence of social norms via learning from interaction experiences. In our model, individual agents repeatedly interact with other agents in the society over instances of a given scenario [10]. Each interaction is framed as a stage game. An agent learns its policy to play the game over repeated interactions with multiple agents. We term this mode of learning *social learning*, which is distinct from an agent learning from repeated interactions against the same player. We are particularly interested in situations where multiple action combinations yield the same optimal payoff. The key research question is to find out if the entire population learns to converge to a consistent norm.

The specific social learning situation for norm evolution that we consider is that of learning "rules of the road". In particular, we have considered the problem of which side of the road to drive in and who yields if two drivers arrive at an interaction at the same time from neighboring roads [10]. When two cars arrive at an intersection, a driver will sometimes have another car on its left and sometimes on its right. These two experiences can be mapped to two different roles an agent can assume in this social dilemma scenario and corresponds to an agent playing as the row and column player respectively. Consequently, an agent has a private bimatrix: a matrix when it is the row player, one matrix when it is the column player. For this problem, we use a bimatrix where both players get a high value (4) if they choose the same action and a low payoff (-1) otherwise. Note that either action combinations (0,0) or (1,1) would be equally desirable. Each agent has a learning algorithm to play as a row player and as a column player and learns independently to play as a row and a column player. An agent can observe opponent action but not their payoff. The goal is then for all agent to develop a norm of choosing the same action consistently.

Each agent is paired in each time period for interaction with a randomly selected agent from a subset of the population. An agent is randomly assigned to be the row or column player in any interaction. We assume that the stage game payoff matrix is known to both players, but agents cannot distinguish between other players in the population. Hence, each agent can only develop a single pair of policies, one as a row player and the other as a column player, to play against any other player from the agent population.

In our initial experiments, any two agents in the population had an equal probability of interaction. We observed that social learning was successful in generating consistent norms in the population. The main conclusions of this study was as follows:

• The number of interactions required to evolve a consistent norm increases with the population size and the number of actions, m, available to each agent.
• The number of interactions required to evolve a consistent norm varies with the learning algorithm used by the agents. If different agents used different learning algorithms (heterogeneous learning population), the convergence rate is in between the rates for homogeneous populations using the constituent learning algorithms.
• Different norms producing equal payoffs emerged equally often over different runs. However, when we introduced non-learners, i.e., fixed-strategy agents who always chose a given action (for example, always driving on the right), only a handful of additional non-learners following a given norm compared to others led to the corresponding norm emerging significantly more often in the population.
• If the population was segregated with very infrequent interactions between agents belonging to different sub-populations, different norms could emerge in different sub-population. It was surprising to see divergent norms emerging even when 25% of the interactions were across sub-populations.

In a more recent paper [11], we have enhanced the interaction model to study spatial interaction effects on norm emergence. In this enhanced model, the agents are distributed over space where each agent is located at a grid point. An agent is allowed to interact only with agents located within its neighborhood composed of all agents within a distance D of its grid location (we have used the Manhattan distance metric, i.e., $|x_1 - x_2| + |y_1 - y_2|$ is the distance between grid locations (x_1, y_1) and (x_2, y_2)). We vary D to allow for different neighborhood sizes. We have experimented with a society of 225 agents placed on a 15 by 15 grid and using the WoLF-PHC learning scheme.

We present in Figure 1 the dynamics of the average payoff of the population over a run when all agents are learning concurrently. We conclude that a norm has emerged in the population when the average payoff of the population reaches 3.5. From Figure 1 we observe that the smaller the neighborhood distance, the faster the emergence of a norm. This is because, for a given number of iterations, the agents interact more often with a particular neighbors for smaller neighborhoods. This means that the impact an agent has on another agent is larger when the neighborhood size is small. In addition, an agent with few neighbors will encounter few different behaviors from its neighbors, and it is *a priori* easier to coordinate with a small set of agents rather than a larger one. the decreasing interaction frequency between pairs of learners increases the time for exploration of the behavior space and thereby influences the learning patterns of the agents in the network.

6 Finding partners

Human and artificial agents routinely make critical choices about interaction partners. The decision about which of several possible candidates to interact with has significant importance on the competitiveness, survivability, and overall utility of an agent. We assume that an agent has time and resource constraints

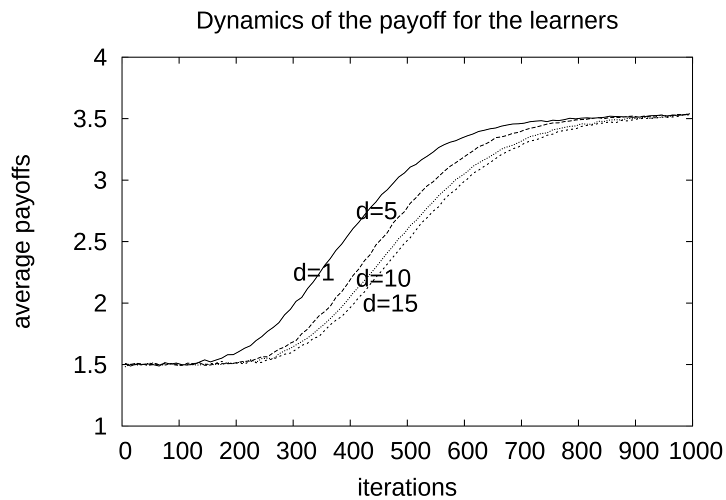

Fig. 1. Influence of neighborhood size on learning rate. All agents are learning.

that limit its participation to only a fixed number, k, of relationships or interactions with other agents in a particular time period. Therefore, in a given time period, an agent is free to choose to interact with any k other agents from a society of N agents. A bilateral relationship is established in a time period, however, if both agents choose to do so. The goal of this research is to investigate the extent to which well-known, simple learning schemes can identify and sustain mutually beneficial relationships in these conditions [12].

A number of multiagent learning algorithms have been developed recently that converge to equilibrium in repeated play. We, however, believe that it is much more likely that simple, single-agent reinforcement learning techniques will be used by a large majority of the agents that in open real-world environments. Therefore, we use Q-learning as the learning algorithm used by our population of agents. We do not know of any other research that has attempted such massively concurrent learning by a large number of utility maximizing agents using single-agent reinforcement learning techniques where the agent utilities are closely coupled. Not only is the likelihood of convergence of such interlinked learning to effective selections unclear *a priori*, no weak guarantees about performance can also be provided. Our experiments, however, show that independent Q-learning by concurrent learners with sufficient exploration is surprisingly robust in identifying most of the mutually beneficial relationships in the society.

7 Negotiating fair allocation of resources

We study the problem of autonomous agents negotiating the allocation of multiple indivisible resources. It is difficult to reach optimal outcomes in bilateral or multi-lateral negotiations over multiple resources when the agents' preferences for the resources are not common knowledge. Self-interested agents often end up negotiating inefficient agreements in such situations. We have developed a protocol for negotiation over multiple indivisible resources which can be used by rational agents to reach efficient outcomes [13]. Our proposed protocol enables the negotiating agents to identify efficient solutions using systematic distributed search that visits only a subspace of the whole solution space.

We represent a negotiation scenario for allocation of multiple indivisible resources as a 3-tuple $\langle \mathcal{A}, R, \mathcal{U} \rangle$, where $\mathcal{A} = \{1, 2\}$ is the set of agents, $R = \{r_1, r_2, \ldots, r_H\}$, $H \geq 2$, is the set of H indivisible resources whose allocation are being negotiated, and $\mathcal{U} = \{U_1, U_2\}$ is the set of utility functions, where U_i is the utility function of agent i. Each resource is considered as a negotiation issue. The negotiating agents must agree on the allocation of the resources.

We assume that the issues or resources are ordered, e.g., lexicographically. We conceptualize the allocations of the resources as a binary *negotiation tree*. The root node represents a null allocation to the agents and then each successive level represents allocation of the next resource in the order chosen. The left and right branches at the l^{th} level imply that the l^{th} resource will be allocated to agent 1 and 2 respectively. Each leaf node at level H represents one possible allocation of the resources and the path from the root node to that leaf node lists the allocation of all the resources. A negotiation tree is created by the negotiating agents in a distributed, top-down process starting at the root node. At any level, agent 1 can only create the right child of a node in the previous level of the tree. Similarly, agent 2 can only create the left child nodes. Each agent, however, may choose not to create any of the nodes it can create, and such a node will be marked as *black* node and it will be pruned from the negotiation tree.

At each node of the negotiation tree, each agent has a *best possible agreement (BPA)* which is the allocation where the resources until the current level are allocated according to the path from the tree root to this node and the remaining resources are allocated to this agent. An individually rational agent will prune a node whose BPA utility is less than the utility it can receive otherwise.

The *Protocol to reach Optimal agreement in Negotiation Over Multiple Indivisible Resources (PONOMIR)* consists of three phases. The first phase consists of a primary allocation procedure using any one of *strict alteration* or *balanced alteration* protocol to produce a default allocation L. The second phase consists of distributed formation of the negotiation tree by the negotiating agents where agents prune nodes as mentioned above. If no nodes are created at level $l < H$, the final allocation is L. Otherwise L and the nodes at level H make up the probable agreements Q at the end of the second phase. In the third phase, agents reach the final Pareto optimal solution by exchanging offers from Q. Agents take turn in making offers from Q, and the recipient removes agreements from Q that are dominated by the received offer. When Q cannot be reduced further, an agreement is picked randomly from it.

Our goal was to develop protocols that lead rational agents to Pareto optimal agreements and to increase fairness as much as possible. As a measure of fairness, we use *egalitarian social welfare*. PONOMIR is not strategy-proof and does not guarantee Pareto optimal agreements if agents are arbitrarily risk seeking. The rational behavior of the agents, who have no prior knowledge of the preferences of the other agents, depends on their risk attitudes. We assume that such agents will be *cooperative-individually rational* which means that i) an agent will not take any risky action that can lead to an agreement which produces less utility than what it is already assured of, and, ii) if there exists two agreements which produces same utility to it but different utility to the opponent, the agent will agree to accept any of the agreement proposed by the opponent. PONOMIR guarantees Pareto optimal agreements if the participating agents are cooperative-individually rational. The agreements reached also guarantees at least as much egalitarian social welfare as the agreements reached by the existing protocols.

8 Bidding for bundles in auctions

Agents with preferences for bundles of items are faced with a difficult computational problem when each of the several electronic auctions sell only one item. While an optimal bidding strategy is known when bidding for item bundles in sequential auctions, only suboptimal strategies are known for simultaneous auctions. We investigate a multi-dimensional bid improvement scheme, motivated by optimization techniques, to derive optimal bids for item bundles in simultaneous auctions [14].

We consider multiple sealed-bid auctions offering items from the set $\mathcal{I}$. A valuation function ϑ expresses the bidder's preferences for bundles or subsets of items from the set $\mathcal{I}$, i.e., the bidder is willing to pay up to $\vartheta(I)$ for a bundle of items $I \subseteq \mathcal{I}$. Each item i is available only in the single-item single-unit auction a_i. We do not specify the particular auction type but make the *exogenous price* assumption: the bids of our bidder do not influence the auction closing prices. An auction is modeled by the probability distribution F_i of the closing prices of the item being offered in that particular auction. We assume these distributions to be continuous, independent, and known by the bidder. In practice approximate price distributions can be learned from observing electronic markets. When an auction closes, a closing price $p_i \in [\underline{p_i}, \overline{p_i}]$ is drawn from the distribution F_i. The bidder gets the item if it has placed a bid b_i greater than or equal to the closing price, i.e., if $p_i \leq b_i$, and the winning payment is equal to the closing price p_i. All auctions run in parallel and their closing times are not known by the bidder. The bidder place bids represented by $B = (b_1, \ldots, b_N) \in \mathcal{B}$ where $\mathcal{B}$ is the bid domain for all auctions. Replacing a bid is not allowed in this model.

Once all the auctions close, the bidder can compute its utility $\alpha(B, P)$ where $P = (p_1, \ldots, p_N)$ represents the closing prices of all auctions. The set of acquired items $\mathcal{I}_{ac}(B, P)$ is calculated as $\mathcal{I}_{ac}(B, P) = \{i \in \mathcal{I} \; s.t. \; p_i \leq b_i\}$ and the corresponding utility received by the bidder is $\alpha(B, P) = \vartheta(\mathcal{I}_{ac}(B, P)) - \sum_{i \in \mathcal{I}_{ac}(B, P)} p_i$. The expected utility is then $\bar{\alpha}(B) = E_P[\alpha(B, P)]$ which can be calculated as specified in Proposition 1:

Proposition 1 (Expected utility)

$$\bar{\alpha}(B) = \sum_{I \subseteq \mathcal{I}} \left\{ \left(\prod_{i \in I} F_i(b_i) \right) \left(\prod_{j \notin I} (1 - F_j(b_j)) \right) \vartheta(I) \right\} - \sum_{i=1}^{N} \int_{\underline{p_i}}^{b_i} p_i \, f_i(p_i) \ dp_i,$$

where $F_i(b_i) = Pr\{p_i \leq b_i\} = \int_{\underline{p_i}}^{b_i} f_i(p_i) \, \mathrm{d}p_i$ and f_i is the pdf of F_i. Our research objective is to find a bid vector B^* which maximizes the expected utility $\bar{\alpha}$: $B^* = \underset{B \in \mathcal{B}}{\operatorname{argmax}} \ \bar{\alpha}(B)$. Assume that the bidder has decided by some means to bid the vector B. If B is sub-optimal, there is at least one item whose bid can be improved, i.e., there exist i and δ_i such that $\bar{\alpha}(B) < \bar{\alpha}((b_i + \delta_i) \vee B_{-i})$ where $B_{-i} = (b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_N)$ and $b_i' \vee B_{-i} = (b_1, \ldots, b_{i-1}, b_i', b_{i+1}, \ldots, b_N)$. By repeating this process, we can realize the best improvement possible for the item i, which is equivalent to maximizing the function $b_i \mapsto \bar{\alpha}(b_i \vee B_{-i})$.

Definition 1 (Optimal bid for item i) *$\beta_i(B_{-i})$ is the optimal bid for item i given bids for item $j \neq i$ is fixed: $\beta_i(B_{-i}) = \underset{b_i \in [\underline{p_i}, \overline{p_i}]}{argmax} \ \bar{\alpha}(b_i \vee B_{-i})$.*

Proposition 2 (Optimal bid for item i)

$$\beta_i(B_{-i}) = \sum_{\substack{I \subseteq \mathcal{I} \\ i \in I}} \prod_{\substack{j \in I \\ j \neq i}} F_j(b_j) \prod_{l \notin I} (1 - F_l(b_l)) \vartheta(I) - \sum_{\substack{J \subseteq \mathcal{I} \\ i \notin J}} \prod_{j \in J} F_j(b_j) \prod_{\substack{l \notin J \\ l \neq i}} (1 - F_l(b_l)) \vartheta(J).$$

To implement this solution approach, an initial bid vector B is chosen and N components of this bid vector are repeatedly improved in any predetermined order. Improving the bid for item i involves replacing b_i by $\beta_i(B_{-i})$: $B \leftarrow \beta_i(B_{-i}) \vee B_{-i}$. We refer to this improvement as *single improvement* and the sequence of N improvements as *N-sequential improvement*. In the bid domain $\mathcal{B}$, the single improvement can be regarded as going from B to the hyper-surface $b_i = \beta(B_{-i})$ by moving parallel to the b_i-axes. The process is stopped when no further improvement can be made. We refer to this sequential bid-improvement process as a Multi-Dimensional Bid Improvement (MDBI) scheme.

The MDBI algorithm produces optimal bid vectors with infinite random restarts. As this is infeasible in practice, we evaluated variants of the scheme with finite number of restarts and carefully chosen starting bid vectors. A surprising result from these experiments is that good solutions can be achieved without restarts when items are substitutable, complementary, or are non-related. In general, small number of restarts can be used to approximate optimal expected utility in all cases. A desirable property of our algorithm, in contrast to existing schemes, is its approximately linear time complexity. Experimental results show that the MDBI scheme scales up effectively with larger number of items.

9 Selecting service providers based on referrals

Agents searching for high-quality services can use either their own interaction experience or referrals from peer agents. We assume that agents want a quality of service that exceeds an acceptable performance threshold. The performance of a resource depends on its intrinsic characteristics and is inversely correlated to the current workload it is handling. Individual agent satisfaction depends both on the resource selected and choices made by the other agents. We study efficient decentralized protocols for finding satisfying resources [15]. Locally optimal actions can increase the number of conflicts of interests where resources are shared. Referrals from other agents can help agents find more satisfying service providers. However, such referrals may cost the referring agent since the load of the referred provider may increase, with corresponding performance deterioration.

Framework: Let $\mathcal{E} =< \mathcal{A},\, \mathcal{R},\, perf,\, L,\, S,\, \Gamma >$ where: (i) $\mathcal{A} = \{a_k\}_{k=1..K}$ is the set of agents, (ii) $\mathcal{R} = \{r_n\}_{n=1..N}$ is the set of resources, (iii) $f:\ \mathcal{R}\times\mathbb{R} \rightarrow [0,\ 1]$, intrinsic performance function of a provider, (iv) $L = \mathcal{A} \rightarrow \mathbb{R}_+$, daily load assigned to agents, (v) $S:\ \mathcal{A} \times [0,1] \rightarrow [0,\ 1]$, satisfaction function for agents, (vi) $\Gamma = \{\gamma_1,\ \ldots,\ \gamma_K\}$, set of satisfaction thresholds of agents. If a set $\mathcal{A}_n^d$ of agents use the provider r_n on day d then the feedback received by every agent in $\mathcal{A}_n^d$ at the end of the day d is $perf = f\left(r_n,\, \sum_{a\in\mathcal{A}_n^d} L(a)\right)$. An agent $a_k \in \mathcal{A}_n^d$ evaluates the performance of r_n by the the satisfaction it obtained and is given by $s = S(a_k,\, perf)$. An agent is satisfied if $s > \gamma_k$.

Definition 2 (Distribution of agents over providers) *We call* **distribution of agents over providers** *the set $D = \{\mathcal{A}_n\}_{n=1..N}$ where $\mathcal{A}_n$ is the set of agents using resource r_n. The set of distributions is denoted by $\mathcal{D}$.*

Definition 3 (Γ-acceptable distribution) *A distribution D is said to be* **Γ-acceptable** *if each agent is satisfied by the resource they use in D. The set of Γ-acceptable distributions is denoted by $\mathcal{D}_\Gamma$.*

A Γ-acceptable distribution is an equilibrium concept and our goal is to develop mechanisms that enable a group of agents to converge to such a distribution. We present alternative strategies for selecting service providers.We evaluate three kinds of agents: agents who find providers on their own without using information from other agents (*No Referral* or *NR*), agents who use referral to locate providers and are trustful of the referrals received(*Referral (Truthful)* or *RT*).

Definition 4 (Entropy) *Given an environment where agents are identical and resource r_n has capacity C_n, we call entropy of a distribution D: $\mathcal{E}(D) = \sum_{n=1}^{N} \max\,(0,\ |\mathcal{A}_n| - C_n)$.*

Each Γ-acceptable distribution has zero entropy. The lower the entropy the better the distribution since less agents are unsatisfied.

We claim that when agents choose their actions based on local perspective only, the system is likely to move from a distribution with a low entropy to one with a higher entropy and vice and versa. Such oscillations can be controlled by limiting the number of agents moving simultaneously, K_{move}. We experimentally show K_{move} has a critical influence on the convergence speed. A high value for K_{move} leads to system oscillations and hence is undesirable.

Experimental results: 1. There exists a lower bound of the number of resources N^* for effective system performance. For any agent type, convergence speed is optimal when $N = N^*$.
2. For any agent type, performance in Zone I, i.e., for $N < N^*$ is worse compared to performance in Zone II, i.e., for $N \geq N^*$.
3. When $N \geq N^*$ for all strategies, i.e., in the range $N \geq 100$: *RT* converges faster than *NR*.
4. *NR* is more robust than other algorithms as it produces convergence for a much larger range of environments, e.g., only *NR* leads to convergence within the iteration limit for $N \leq 20$.

Interestingly, systems without referrals appear to be more robust in the sense they have satisfactory or reasonable performance even for extremely small number of providers, i.e., for more challenging environments. Referrals, however, do facilitate convergence when there are a significant number of providers.

10 Detecting malfunctioning nodes in sensor networks

Sensor network applications often require remote, distributed monitoring of inaccessible and hostile locations. These networks are vulnerable to both physical and electronic security breaches. Nodes in a sensor network can be hierarchically organized, where the non-leaf nodes serve as the aggregators of the data sensed at the leaf level and data collected from the entire network is available at the root node to users for monitoring the environment. Erroneous data, either from compromised or defective nodes, can influence the aggregated result and can undermine the effectiveness of the network for effectively reporting data about the environment. Current research on sensor networks use outlier detection mechanisms by a parent node to detect the erroneous nodes among its children. It is assumed that the data reported by the children of a node is sampled from the same distribution and they are of almost equal values. We believe that effective, distributed attack mechanisms will eschew egregious deviations and report smaller errors by a colluding group of compromised nodes. We have developed a distributed reputation framework that can learn to recognize such distributed attacks over repeated data aggregation periods [16].

The above approaches, however, will be ineffective for networks where the data sensed vary widely from one portion to another as the data sensed by different nodes are not from the same distribution. In such a scenario, we have used neural network based learning technique to predict data reported by different nodes. We train the nets offline from sufficient data collected after initial network deployment [17]. Subsequently, parent nodes monitor their children by calculating the differences between the value reported by a child node and that predicted by the net based on the data reported by that node's siblings. Each node incrementally updates the reputations of its child nodes based on those calculated differences. We have used robust schemes like Q_learning and a Beta-reputation scheme to detect the faulty/malicious nodes. We have incorporated different degrees of node's physical and geometrical features (e.g varying coordinates of anomalous nodes, the number of malicious nodes, the errors imparted, data pattern etc) into our experiments and demonstrated robust system performance under varying environmental conditions.

11 Detecting deviation from team plans

Effective decentralized control mechanism are required for multiple agents co-operating to achieve a common goal while limited by computing and communication limitations and possible security breaches. Multiagent planning techniques computing near-optimal joint-strategies that can handle intrinsic domain uncertainties. Uncertainties related to agents deviating from the recommended joint-policy, however, is typically not taken into consideration. We focus on hostile domains, where teams must quickly identify deviations from team plans by compromised agents. There is a growing need to develop techniques that enable the system to recognize and recover from such deviations. We have developed a distributed probabilistic intrusion detection scheme for detecting a particular type of deviations by team members [18].

The problem of decentralized control can be effectively modeled as a decentralized partially observable Markov Decision Process (DEC-POMDP). A DEC-POMDP is given by a tuple $< I, S, \{A_i\}, \{\Omega_i\}, O,$
$P, R, b_0 >$ where I is the finite set of agents indexed by $1 \dots n$, S is a finite set of states, A_i is a finite set of actions available to agent i and $A = \times_{i \in I} A_i$ is the set of joint actions where a $=< a_1, \dots, a_n >$ denotes a joint action, Ω_i is a finite set of observations available to agent i and $\Omega = \times_{i \in I} \Omega_i$ is the set of joint observations where $o =< o_1, \dots, o_n >$ denotes a joint observation, O is the observation function given by $O(s, a_1, \dots, a_n, o_1, \dots, o_n)$, the probability of observing the joint observation $(o_1, \dots, o_n)$ when transitioning to state s after taking joint action $(a_1, \dots, a_n)$, P is the set of Markovian state transition probabilities where $P(s, a, s')$ denotes the probability of taking action a in state s and reaching state s', R: $S \times A \rightarrow \Re$ is the common reward function, and b_0 is the initial belief state for all agents. We assume that the agent's observations are independent. Thus the observation function can be represented as $O = \times_{i \in I} O_i$ where $O_i(s, a_1, \dots, a_n, o_i)$ is the probability that agent i observes o_i given the joint-action $< a_1, \dots, a_n >$ resulted in state s. The decision problem spans over a finite horizon T. The policy for agent i, π_i is represented by a policy tree. Each node corresponds to an action and each edge corresponds to an observation that the agent makes at that time interval. We assume that a centralized planner computes the policy tree for each agent. The running belief state of agent i at time interval t is its estimate of the physical states and the observation histories of the other agents and is given by $RB_i^t : S \times \boldsymbol{o^t_{-i}} \rightarrow [0, 1]$ where $\boldsymbol{o^t_{-i}}$ are the

t'th observation histories of other agents. We define Bel_i^t as the set of all such possible combinations of physical states and observation histories that have a positive probability in RB_i^t: $Bel_i^t = \{b | RB_i^t(b) > 0\}$. The agents update RB_i^t and Bel_i^t with each execution step.

Each agent i maintains a set $V_i = \{R_i^j\}$ where R_i^j is the reputation of the jth agent as computed by agent i, $\forall j \neq i$, and is updated in each iteration by:

$$R_i^j \leftarrow R_i^j - \kappa(R_i^j) \times \sum_{\forall < s, \boldsymbol{o^t_{-i}} > \in Bel_i^{t^{max}}} (max_{o_j \in \Omega_j} O_j(s, < \pi_i(\boldsymbol{o_i^{t-1}}), \pi_{-i}(\boldsymbol{o_{-i}^{t-1}}) >, o_j)) - O_j(s, < \pi_i(\boldsymbol{o_i^{t-1}}), \pi_{-i}(\boldsymbol{o_{-i}^{t-1}}) >, o_j^t >))/|Bel_i^{t^{max}}|$$

where $Bel_i^{t^{max}} = \{b | RB_i^t(b) = max_{b' \in Bel_i^t} RB_i^t(b')\}$. $Bel_i^{t^{max}}$ is a subset of beliefs most convincing to i. Based on $Bel_i^{t^{max}}$, i reasons about the last observational transition that each of the other agents have made. Note, a simple malicious agent k would often fake observations and this inconsistency would gradually reflect in the Bel_i^t of i and result in a higher value for the numerator. This would result in a sharp drop of R_i^k. The function κ is monotonically decreasing with R_i^j and thus facilitates faster detection. We have shown the effectiveness of this scheme on the Tiger problem [18].

12 Conclusions

While the brief summaries presented here provide only coarse outlines of the research results, my website (`http://www.mcs.utulsa.edu/~sandip`) can be perused both to obtain complete papers with extended discussion on these topics as well as to obtain details on related research of key relevance to the topic of developing sustainable agent communities. In addition, our research group has worked on related areas on multiagent learning, trust-based computing, peer-level negotiation schemes, proactive information dissemination, cooperative security envelops, etc. that are key components of the set of technologies required to design, develop and implement self-interested social agents. Such agents must balance local needs with societal constraints to maximize long-term utility. In particular, they have to leverage complementary expertise, proactively seek out collaboration opportunities, and cooperatively avoid unforeseen problems and inefficiencies. Our ongoing work is focused on techniques and methods to make our vision of a vibrant, self-sufficient intelligent agent community a reality in the foreseeable future.

Acknowledgments: This work has been supported in part by an NSF award IIS-0209208, a DOD-ARO Grant #W911NF-05-1-0285, and a NASA Research Initiation Grant NCC5-586. We also thank our collaborator Prithviraj Dasgupta.

References

1. Sen, S.: Reciprocity: a foundational principle for promoting cooperative behavior among self-interested agents. In: Proceedings of the Second International Conference on Multiagent Systems, Menlo Park, CA, AAAI Press (1996) 315–321

2. Sen, S.: Believing others: Pros and cons. Artificial Intelligence **142**(2) (2002) 179–203
3. Sen, S., Dutta, P.S.: The evolution and stability of cooperative traits. In: Proceedings of the First Intenational Joint Conference on Autonomous Agents and Multiagent Systems, New York, NY, ACM Pres (2002) 1114–1120
4. Saha, S., Sen, S.: Predicting agent strategy mix of evolving populations. In: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, New York, NY, ACM Pres (2005) 1075–1082
5. Banerjee, D., Saha, S., Dasgupta, P., Sen, S.: Reciprocal resource sharing in p2p environments. In: Proceedings of the Fourth Intenational Joint Conference on Autonomous Agents and Multiagent Systems, New York, NY, ACM Pres (2005) 853–859
6. Airiau, S., Sen, S., Dasgupta, P.: Effect of joining decisions on peer clusters. In: Proceedings of the Fifth International Joint Conference on Autonomous Agents & Multi Agent Systems, New York, NY, ACM Pres (2006) 609–615
7. Banerjee, D., Sen, S.: Reaching pareto optimality in prisoner's dilemma using conditional joint action learning,. In: Working Notes of the AAAI-05 Workshop on Multiagent Learning,. (2005)
8. Chakraborty, D., Sen, S.: Teaching new teammates. In: Proceedings of the Fifth International Joint Conference on Autonomous Agents & Multi Agent Systems, New York, NY, ACM Pres (2006) 691–693
9. Kar, P.P., Sen, S.: The agent-teaching-agent framework. In: Proceedings of the Second International Joint Conference on Autonomous Agents & Multi Agent Systems, New York, NY, ACM Pres (2003) 1028–1029
10. Sen, S., Airiau, S.: Emergence of norms through social learning. In: Proceedings of the Twentieth International Joint Conference on Artificial Intelligence(IJCAI'07). (January 2007) 1507–1512
11. Mukherjee, P., Sen, S., Airiau, S.: Norm emergence in spatially constrained interactions. In: Working Notes of the Adaptive and Learning Agents Workshop at AAMAS'07,. (2007)
12. Sen, S., Gursel, A., Airiau, S.: Learning to identify beneficial partners. In: Working Notes of the Adaptive and Learning Agents Workshop at AAMAS'07,. (2007)
13. Saha, S., Sen, S.: An efficient protocol for negotiation over multiple indivisible resources. In: Proceedings of the Twentieth International Joint Conference on Artificial Intelligence(IJCAI'07). (January 2007)
14. Candale, T., Sen, S.: Multi-dimensional bid improvement algorithm for simultaneous auctions,. In: Proceedings of the Twentieth International Joint Conference on Artificial Intelligence,. (2007)
15. Candale, T., Sen, S.: Effect of referrals on convergence to satisficing distributions. In Dignum, F., Dignum, V., Koenig, S., Kraus, S., Singh, M.P., Wooldridge, M., eds.: Proc. 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), ACM (2005) 347–354
16. Basak, S., Sen, S.: Using distributed reputation management to preserve data integrity in sensor networks,. Journal of Autonomic and Trusted Computing (to appear).
17. Mukherjee, P., Sen, S.: Detecting malicious sensor nodes from learned data patterns. In: Working Notes of the Agent Technology for Sensor Networks workshop at AAMAS'07,. (2007)
18. Chakraborty, D., Sen, S.: Distributed intrusion detection in partially observable markov decision processes. In: Proceedings of the Sixth International Joint Conference on Autonomous Agents & Multi Agent Systems. (2007) (to appear).