Information Sharing in Large Scale Teams

Yang Xu, Michael Lewis
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA 15260, USA
yxu@sis.pitt.edu, ml@sis.pitt.edu

Katia Sycara, Paul Scerri
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
katia@cs.cmu.edu, pscerri@cs.cmu.edu

Abstract

Effective communication among agents in large teams is crucial because the members share a common goal but only have a partial views of the environment. Information sharing is difficult in a large team because, a team member may have a piece of valuable information but not know who needs the information, since it is infeasible to know what each other agent is doing. Although much related work has been done on efficient delivery of information, most work is based on assumptions which are not suited to large scale multiagent teams.

In this paper, we made two contributions. Firstly, we present a solution to sharing information that is applicable to large teams based on previous research [12]. A key to the solution is imposing a static network topology on the members of the team where each agent requiring communication to be only along very few links in that network. The key observation underlying this solution is that each piece of information is interrelated and the sender of a piece of information can "guess" who might need some information based on previously sent messages. Thus, when an agent has a piece of information, it can determine which of its neighbors in the network is most likely to either need the information or know who does, based on related messages previously received. Secondly, we investigate the influence of different types of team network topology on the efficiency of information sharing. Our results show that our algorithm works with various topologies but gets the best performance on a scale free network.

1. Introduction

Exciting emerging applications require hundreds or thousands of agents and robots to coordinate to achieve their joint goals. In domains such as military operations, space or disaster response, coordination among large numbers of agents promises to revolutionize the effectiveness of our ability to achieve complex goals. Such domains are characterized by widely distributed entities with limited communication channels among them and no agent having a complete view of the environment. Information relevant to team goals will become available to team members in a spontaneous, unpredictable and, most importantly, distributed way. The question we address in this paper is when a team member senses some information, how it can decide which team member to communicate that information to. In most applications for very large teams, broadcasting information is not suitable, desirable or feasible. Instead, the agent must attempt to target its information delivery to just the agents that need it. In a large team, each member has a limited model of what other members of the group know or even what many of them are doing. For example, an unmanned aerial vehicle (UAV) involved in a military operation may observe many features of a battlefield on route to an assignment. Many of its observations will be relevant to the plans of other combatants but the UAV will not necessarily know which group members require the information.

While the problem of what and whether to communicate has been extensively studied [4,5,14], previous work typically makes one of two assumptions, rendering it inappropriate for very large teams. One strand of researches assumes that it is feasible to have all agents communicate with some central control [6,9]. Both centralized algorithms and distributed approaches that rely on all agents communicating with one particular agent, e.g., a match maker or information broker, make this assumption. In very large groups such centralization is not feasible. A second strand of researches [14,18] relies on having accurate models of what other group members are doing, e.g., STEAM [13] relies on such information. However, in very large groups, agents will typically only have accurate knowledge of what a subset of other agents are doing (i.e., those agents with which it is currently closely coordinating). Hence, previous work has avoided the more difficult problem of distributed information delivery when the state of the team is not known.

Our solution for information sharing among large teams can perform distributed information sharing without the cost of maintaining accurate models of all the teammates. First, we impose a network topology on the team members analogous to the social networks that exist in human societies. The key characteristic of this network model is that information exchange is based on peer to peer communication. Specifically, we limit agents to communicating directly with only a small percentage of the overall team. This kind social information sharing model has been proved efficient in human groups on allowing information to efficiently reach many people by passing from friend to friend with very few "jumps". In this paper, analogous to human groups, agents between whom there is a direct communication channel are referred to as acquaintances and the resulting information exchange topology is referred to as an acquaintance network. The central reason why communication decision making is hard is not that it is fundamentally hard, but that no single agent has all the required information to make the decision where or to whom to send information. That is, an agent can easily know what information they need, but it will not know who has the information, while another agent has the information but does not know who needs it. By allowing the agents to simply forward the information to acquaintance in a better position to make the decision we spread the reasoning across the team, leveraging the knowledge of many agents. We also leverage the idea that information is always interrelated and a received piece of information can be useful in deciding where to send another piece of information if there is a relationship between two pieces of information. For example, when coordinating an agent group in urban search and rescue, if agent a tells agent b about a fire at 50 Smith St, when agent b has the information about the traffic condition of Smith St, sending that information to agent a is a reasonable thing to do, since a likely either needs the information or knows who does. By utilizing the interrelationship between pieces of information, agents can more quickly route new information through the acquaintance network. Moreover, agents do not model information, rather they model which of their acquaintances to send information to. It may take several hops for a message to get to an agent that needs the information. Since each piece of information informs delivery of other pieces and models are updated as the message moves, the volume of information to be shared among the team increases the amount of effort required per piece of information actually decreases. Moreover, since agents need to only know about their acquaintances, the approach scales as the number of agents in the team increases.

Furthermore, our research includes the success that humans have in getting information to those who need it to enhance information sharing among agent teams. We observe that in a human group, members typically maintain a small number acquaintances but can rapidly transmit information to any member of the group in a series of hops, a phenomena known as a *small world effect*. The most popular manifestation of this phenomena is the *six degrees of separation* concept, uncovered by the social psychologist Stanley Milgram [9]. Milgram concluded that there was a path of acquaintances with typical length six between any two people in the United States. This experiments showed that using very vague (and often incorrect) information about other members of the population, people will pass a message to someone better placed to find the intended recipient until the information reaches the desired recipient. Researchers from other fields have revealed that social structures, patterns and interconnections have a strong impact on the effectiveness of communication and cooperation among human society [2]. Our algorithm leverages this effect to get efficient information sharing in agent teams. And our experiment shows the small world effect [16] and power law distribution [1] of connectivity can greatly enhance the efficiency of information delivery.

This paper is organized as follows: firstly, the basic model of how teams are organized for efficient information is presented in section 2; Section 3 describes the algorithm the agent team uses to share information; In the section 4, we investigate each social network property and analyze how they potentially enhance information sharing. In the last section, we present the experiment results to identify our approach above.

2. Information Sharing in Large Scale Teams

We are developing information sharing algorithms for team with the following basic characteristics: there are large number of widely distributed team members with limited communication bandwidth and as a part of a large team, agents only coordinate closely with a subset of the rest agents of the team. Based on these characteristics, we can define a model for information sharing among large scale teams. Specifically, we define an agent team as a tuple with three elements: the

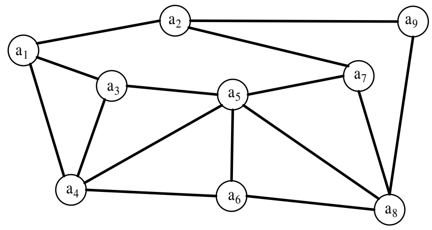

Figure 1. An Example of Team Organization where each agent only have a small number of acquaintances.

team members, the acquaintance network, the information to be shared. $A = \{a_1, a_2, ..., a_n\}$ is the agent team and consists of a large number of agents, normally more than 100. N is the definition of network topology agents use to share information with each other. We refer to agents with link between them as *acquaintances*. $N(t) = \bigcup_{a \in A(t)} n(a)$ defines the acquaintance network of the team, where $n(a)$ is defined as all the acquaintances of agent a. Note that the number of each agents' acquaintances is much less than the size of agent team $|A|$. A subset of a typical acquaintance network for a big team is shown as Figure 1. In the figure, each node represents an agent member in the team, and when pairs of agents are connected, they can sharing information with each other as acquaintances. Further description of the types of acquaintance networks will be discussed in Section 4.

The state of an agent a is written as S_a and is defined by a tuple $S_a =< H_a, K_a >$. H_a is the (potentially truncated) history of messages received by this agent. $I = \{i_1, i_2, \ldots, i_n\}$ is the alphabet of domain information that is to be shared. $i \in I$ denotes a specific piece of information, such as "There is an enemy at (12, 12)". $K_a \subseteq I$ is the local knowledge of the agent. If $i \in K_a$, we say agent a *knows* information i and $knows(a, i) = 1$, otherwise, $knows(a, i) = 0$. Typically, individual team members will know only a small fraction of all the team knows, i.e., $|K_a| << |I|$.

The objective for information sharing is to make as many agent know the information as can make use of it. The importance of the information i is calculated by determining the expected increase in utility for the agent with the information versus without it. That is $U(a, i) = EU(a, K_a \cup i) - EU(a, K_a)$, where $EU(a, K_a)$ is the expected utility of the agent a with knowledge K_a. When $U(a, i) > 0$, knowledge of i is useful to a, and the bigger value of $U(a, i)$, the more that information i is helpful for this agent. The reward for the team

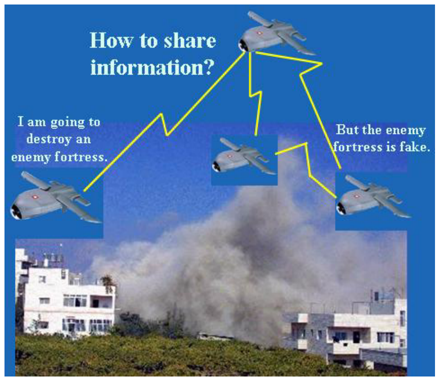

Figure 2. Information sharing difficulty in coordinating WASMs is that WASM who has a piece of information does not know who need it or where send it to.

can be defined as $R(i) = \frac{\sum_{a \in A} U(a,i) \times knows(a,i)}{\sum_{a \in A} knows(a,i)}$. That is, the agent team must get information to those for whom the information can be made best use of, while wasting as few resources as possible communicating information to agents that do not require it. Notice, that since this calculation is based on knowing the use of a piece of information to each agent, agents cannot compute this reward function locally.

3. Algorithm for Information Sharing in a Large Scale Team

Before discussing our model and algorithm in detail, we first provide an example to illustrate the basic idea. This paper is being done as a part of a research project on coordinating a large team of Wide Area Search Munitions (WASM). A WASM is a cross between a unmanned aerial vehicle and a standard munition and can perform functions including reconnaissance, search, battle damage assessment, communications relays and decoys in a hostile environment [12]. In the project, a team of WASMs typically has more than 100 members and is physically distributed. In such a domain, as shown in Figure 2, information exchange is important because the WASM sensing some information is not always the one who can make the best use of it. As a simple example, when a WASM detects enemy movement while on route to destroying another enemy, it must relay this information to an available team member to investigate further.

Leveraging the team network, our basic approach is when an agent has a piece of information to communicate, it forwards that information to the acquaintance most likely to actually need that information or know who will. Then the acquaintance perform the same reasoning when it gets the information. After passing through hopefully, a small number of team members, information arrives at a team member that needs it. The intuition is that each agent attempts to guess which of its acquaintance either require the information or are in the best position to get the information to the agent that requires it. Even though members of large teams will not have accurate, up-to-date models of the team, our hypothesis is that the models will be accurate enough to deliver the information in a small number of "hops".

To test the potential of the approach we ran an experiment where 800,000 agents are organized in a three dimensional lattice [12]. One agent is randomly chosen as the source of some information and another is randomly chosen as the sink for that information. A probability is attached to each link, indicating the chance that passing information down that link will get the information through the smallest number of links to the sink. In the experiment shown in Figure 3, we varied the probability of sending information down links that actually lead to an agent requiring the information (as opposed to sending it down links that moves the information further away) and measured the number of Messages (or "hops") required to get the information from the source to the sink. For example, for the "59%" setting, messages are passe along links getting closer to the sink 59% of the time and links further from the sink 41% of the time. Figure 3 shows that the agents only need to move closer to the target slightly more that 50% of the time to dramatically reduce the number of steps that the message required to reach to the sink. Thus, potentially, even relatively inaccurate models of acquaintances are capable of leading to efficient, targeted information delivery.

This result shown in Figure 3 is encouraging because it shows we do not need to construct accurate and complex models for information sharing but only have reasonable models to improve agent's guessing. The key question is how to create models that allow the agent to "guess" correctly more often than not. To achieve this, we observe that each piece of domain knowledge is typically related to each other piece of domain information. For example, if agent a tells agent b about a plan to destroy an enemy base, when agent b gets the information that the base is fake, sending that information to agent a is a reasonable thing to do, since a likely either needs the information or knows who does (i.e., the WASM attaching the base). So it is reasonable to infer from an agent's formerly sent message that

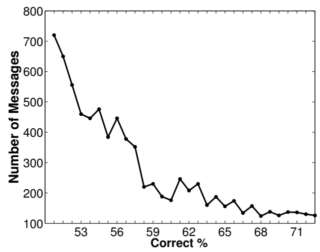

Figure 3. Agents' likelihood of correctness where to passing information can dramatically influence the efficiency of information sharing in an agent team with 800000 members

it may need the other kind of information to improve the performance as the previous example. Thus, the previously received information can be interpreted as *evidence* to infer which acquaintance to send other information to. If an agent maintains a knowledge base about what it heard from its acquaintances, it can use that knowledge to determine where to route newly received information.

Basic Approach

In this section, we formalize the algorithm described above. The key to the algorithm is the model that the agent maintains of its acquaintances. P_a is a matrix where $P_a[i,b] \rightarrow [0,1], b \in N(a), i \in I$ represents the probability that acquaintance b is the best to send information i to. To obey the rules of probability, we require $\forall i \in I, \sum_{b \in N(a)} P^t_a[i,b] = 1$. The more accurate the model of P_a, the more efficient the information sharing because the agent will send information to agents that need it more often and more quickly. P_a is inferred from incoming messages and thus the key to our algorithm is for the agents to build the best possible model of P_a. For example, in Figure 4, if $P_a[i,b] = 0.7$, then a will usually forward i to agent b as b is very likely the best of its acquaintances to send to.

Information is encapsulated into *messages* with some supporting information which is helpful for information sharing. Specifically, a message consists of two parts, $M =< i, path >$. $i \in I$ is the information being communicated. $path$ records the track where the message has been taken in the network. $last(path)$ denotes the agent where the message was sent to current agent via acquaintance network. To ensure messages do not travel indefinitely around the network, we stop the message when $|path| \geq$MAX_STEPS.

When a message arrives, the agent state, S_a, is updated by the transition function, δ, which has three

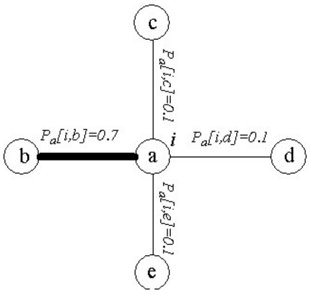

Figure 4. Relative probability example where agent b is more likely to be the best acquaintance to send information i to.

parts, δ_H, δ_K, δ_P . First, the message is appended to the history, $\delta_H(m, H_a) = H_a \cup m$. Secondly, the information contained in the message is added to the agent's local information knowledge K_a, $\delta_H(m, K_a) = K_a \cup m.i$.[1] Finally, and most critically for the purpose of the algorithm, δ_P is used to update agent's probability matrix, to help route future message. (We described δ_P in the next section.)

Each agent in the team runs the following algorithm when receiving message m:

Algorithm 1: Information Share (S_a)
(1) $While(true)$
(2) $m \leftarrow getMsg$
(3) $S_a \leftarrow \delta(m,\ S_a)$
(4) $if\ m.|path|\ < MAX\_STEPS$
(5) $\quad APPEND(self,\ m.path)$
(6) $\quad next \leftarrow CHOOSE(P[i, m.j])$
(7) $\quad SEND(next, m)$

In Algorithm 1, when an agent gets a message, it updates its state according to function δ. If an agent finds the message does not meet the stop condition (line 4), then the function CHOOSE (line 6) selects an acquaintance, according to the probabilities in matrix to pass the message to. Notice, CHOOSE can select any acquaintance, with the likelihood of choosing a particular acquaintance proportional to their probability of being the best to send to.

Updating Acquaintance Models The key to our algorithm is for the agent to often pass information to an acquaintance either needs it or know who does. These models are created based on previously received information. This requires us making use of the relationship between pieces of information and then mapping it into a mathematics description, i.e. via Bayes Rule.

1 In this paper, we ignore difficult issues related to contradictory information.

We define the relationships between pieces of information as $rel(i, j) \rightarrow [0, 1], i, j \in I$, where $rel(i, j) > 0.5$ indicates that an agent interested in i will also be interested in j, while $rel(i, j) < 0.5$ indicates that an agent interested in i is unlikely to be interested in j. If $rel(i, j) = 0.5$ then nothing can be inferred. Since rel relates two pieces of domain level information, we assume that it is given (or can be easily inferred from the domain).

Our algorithm defined action of δ_P for each piece of relative information i when received a message containing j can be described as follows: assuming information j arrives to agent a from b, then agent a will firstly decrease the probability to send this information back to b because clearly b already knows that information. Then H_a should be searched for to find any relevant previous information. For each piece of relevant information i, j should be additional evidence for a how to make decision to send to i and the probability of sending i to b should be strengthened.

The update of agent a's P_a based on an incoming message m containing j which is received from c can be achieved by leveraging Bayes Rule as following:

$$\forall i, j \in I, b \in N(a) \quad \delta_P^I(P_a[i, b], m =< j, path >)$$
$$= \begin{cases} P_a[i, b] \times rel(i, j) \times \frac{2}{|N|} & \text{if } i \neq j, b = last(m.path) \\ P_a[i, b] \times \frac{1}{|N|} & \text{if } i \neq j, b \neq last(m.path) \\ \varepsilon & \text{if } i = j, b = last(m.path) \end{cases}$$

Then P must be normalized to ensure $\forall i \in I, \sum_{b \in N(a)} P_a^t[i, b] = 1$. The first case in our equation is the most interesting. It updates the probability that the agent that just sent some information is the best to send other relative information to, based on the relationships of other pieces of information to the one just sent. The second case changes the probability of sending that information to agents other than the sender in a way that ensures the normalization works. Finally, the third case encodes the idea that you typically would not want to send a piece of information to an agent that sent it to you.

To see how δ_P works, consider the following example at some point doing execution

$$P_a = \begin{array}{c} \\ i \\ j \\ k \end{array} \begin{array}{c} \begin{array}{cccc} b & c & d & e \end{array} \\ \begin{bmatrix} 0.6 & 0.1 & 0.2 & 0.1 \\ 0.4 & 0.2 & 0.3 & 0.1 \\ 0.4 & 0.4 & 0.1 & 0.1 \end{bmatrix} \end{array}$$

The first row of the matrix shows that if a gets information i it will likely send it to agent b, since $P[i, b] = 0.6$. We assume that agents wanting information i also probably want information j but those wanting k definitely do not want j. That is, $rel(i, j) = 0.6$ and $rel(k, j) = 0.2$

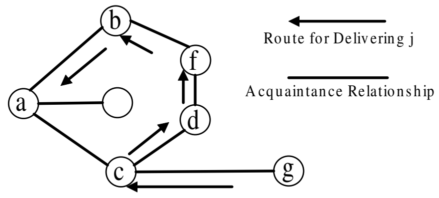

Figure 5. Example of utilizing $path$ for improve information sharing efficiency. Notice, c is a better candidate than b to send information i.

Then a message $m =< j, \{, ,d, ,b\} >$ with information j arrives from agent b. Applying δ_P^I to P_a we get the following result:

$$P_a = \begin{array}{cc} & \begin{array}{cccc} b & c & d & e \end{array} \\ \begin{array}{c} i \\ j \\ k \end{array} & \begin{bmatrix} 0.643 & 0.089 & 0.179 & 0.089 \\ \varepsilon & 0.333 & 0.5 & 0.167 \\ 0.211 & 0.526 & 0.132 & 0.132 \end{bmatrix} \end{array}$$

The effects on P are intuitive: (i) j will likely not be sent back to b, i.e., $P_a[i,b] = \varepsilon$; (ii) the probability of sending i to b is increased because agents wanting j probably also want i; (iii) the probability of sending k to b is decreased, since agents wanting j probably do not want k. Notice a knows nothing of the network topology *beyond* its acquaintances $N(a)$.

Enhance Efficiency of Information Sharing Our algorithm decides how a piece of information will be delivered, based on the path of messages with related information. Since the path of the related messages may not have been the most direct path it is possible the chosen path might not be efficient, either. This can be seen in the example in Figure 5. We assume agent a has acquaintances b and c. And a gets a message with information j which was delivered from b, but from the *path* we know before j arrives at b, j had visited c. In this case, message j has travelled via c and b, so when if we consider how to deliver another piece of information i which $rel(i,j) > 0.5$, c is a better candidate than b. Thus, instead of strengthening $P_a[i,b]$, based on $m =< j, path >$, we should strengthen $P_a[i,c]$. From this example, instead of always using the immediate sender of some information to update P_a, we use the first agent in the path that was a recipient of the information. If the path contains only one a's acquaintance, then the algorithm reduces to the basic algorithm as described above.

In summary, we can improve information sharing in the following way: If agent a get a message with j from an acquaintance, we strengthen the probability of send i to who is a's $first$ acquaintance to get the message with j which can be determined from that message's *path*. The function $first(N(a), m.path)$ is defined to find such an agent. Furthermore, by looking at that messages' path we can find all a's acquaintances who have received j. Since these agents have already received j, it should not be sent back to them. Hence, δ_P^I can be revised as following:

$$\forall i,j \in I, b \in N(a) \quad \delta_P^I(P_a[i,b], m =< j, path >, d = first(N(a), m.path))$$
$$= \begin{cases} P_a[i,b] \times rel(i,j) \times \frac{2}{|N|} & \text{if } i \neq j, b = d \\ P_a[i,b] \times \frac{1}{|N|} & \text{if } i \neq j, b \neq d \\ \varepsilon & \text{if } i = j, b \in m.path \cap N(a) \end{cases}$$

In the first case, the probability that the agent that just sent some information is the best to send other information to is not according to whom directly send that message but a's acquaintance that first got that message. In the third case, all the acquaintances who have received the message are excluded from getting the information back.

Changing the example above, so the message $m =< j, \{, ,d, ,b\} >$the result will be:

$$P_a = \begin{array}{cc} & \begin{array}{cccc} b & c & d & e \end{array} \\ \begin{array}{c} i \\ j \\ k \end{array} & \begin{bmatrix} 0.5769 & 0.096 & 0.2308 & 0.096 \\ \varepsilon & 0.67 & \varepsilon & 033 \\ 0.4255 & 0.4255 & 0.0426 & 0.1064 \end{bmatrix} \end{array}$$

4. Effects of Network Topology on Sharing Efficiency

As noted by social scientists, information sharing efficiency will be impacted by network topology. In our paper,agents to sharing information among a large scale teams adopt the same manners as the social group which composed by human beings.

The properties of social network structure have been comprehensively studied [1,11]. According to that research, there are several parameters which are important to help us understand or predict the behavior of information sharing in large scale teams. Key factors include the small-world effect, degree distributions, clustering, network correlations, random graph models, models of network growth and preferential attachment, and dynamical processes taking place on networks [11]. Most of them are interrelated. So, for the purpose of this paper, we specifically focus on only three properties: average distance, degree distribution and average acquaintance.

- Average distance: (commonly studied as "small world effect" [16]). The average distance $l = \frac{1}{\frac{1}{2}n(n+1)} \sum_{a_i,a_j \in A, i>j} distance(a_i,a_j)$, where

$n = |A|$ and $distance(a_i, a_j)$ represents the minimum number of agents a_i, a_j that a message must pass through one agent to another via acquaintance network. For example, if agent $a_1$and a_2 are not acquaintance but share an acquaintance, $distance(a_1, a_2) = 1$.

- Degree distribution: (Commonly studied as “scale free effect”) The frequency of agents having different number of acquaintances. The distribution can be represented as a histogram where the bins represent a given number of acquaintances and the size of bin is how many agents have such number of acquaintances [1].
- Average acquaintances: is the average number of acquaintances that agents have in the teams. Its value can be used to infer how many choices agents may have when delivering a message.

Well known types of social network can be described using these properties. For example, *random network* have the “flat” degree distribution. While *grid network* is distinct in that all nodes have the same degree. e.g, four is the only degree in two dimension grid network. *Small World Network* [15] and *Scale Free Network* [2] are two important types of social network topologies and research has shown that each of them possess some interesting properties. Small world networks have much shorter average distance than regular grid networks. We hypothesize that the low average distance will improve information sharing efficiently because information can potentially take less "hops" to reach a defined destination. A scale-free network, shown in Figure 6 is a specific kind of network in which the degree distribution forms a power-law, i.e, some nodes are very connected hubs and connect to other nodes much more than ordinary nodes. The hubs in scale free networks give the advantages of centralized networks, which the distribution provides the advantages of centralized approaches.

5. Experiment Setup and Result

In this section, we explore the impact of network topology on algorithm performance. In these experiments, we use a team with 400 agents and each of them has, on average, four acquaintances. One agent is randomly chosen as the source of some information and another is randomly picked as the sink for that information. The sink agent firstly sends out 20 messages containing relative information j, each with MAX_STEPS=50. Then the source agent sends out a message with information i with $rel(i, j)$ varied and we measure how many steps or messages that it takes i to be encapsulated into message and sent to get to the sink agent. In our experiments, four different types of acquaintance network topology are involved: two dimension grid networks, random networks, small world networks and scale free networks. The small world network is based on the grid network with 8% links randomly changed. The key difference between the random network and the scale free network is that the random has a “flat” degree distribution but the scale free network has a power law distribution. Each point on each graph is based on the average of 1000 runs in simple simulation environment.

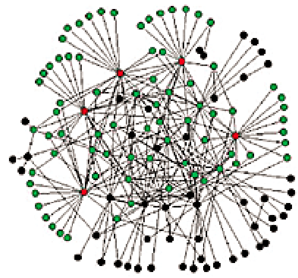

Figure 6. The topology of scale free network whose degree distribution is power law [11]

Information sharing with different information relevance We first verify our basic algorithm in different types of acquaintance network topology. In Figure 7, we show the average number of steps taken to deliver i as we varied the strength of the relationship between the information originally sent out by the sink agent and the information i sent by the source agent from 0.5 to 1. As expected, our algorithm works on the four different acquaintance networks and the stronger the relationship between originally sent information and the new information the more efficient the information delivery.

Information sharing with different number of previous messages Next, we look in detail at exactly how many messages must be sent by the source to make the delivery from the sink efficient. We use the same settings as above except the number of messages the sink sends out is varied and the relationship between these messages and i, rel (i, j) is forced at 0.9. Notice that only a few messages are required to dramatically impact member of messages required. This result also shows us that a few messages is enough for agents to make a "precise guessing" where to send messages.

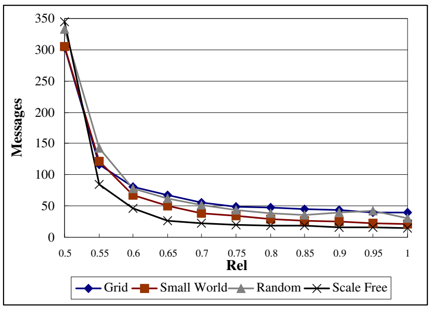

Figure 7. **The number of messages dramatically reduces as the association between information received and information to be sent increases.**

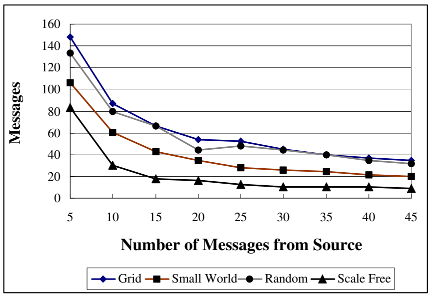

Figure 8. **The number of messages reduces as the relative messages increased from sink.**

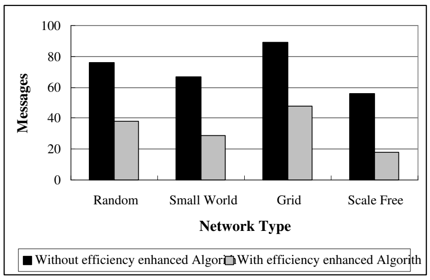

Figure 9. **By using information in path to update probability matrix, the number of messages reduces distinctly.**

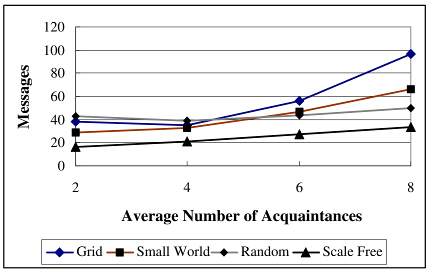

Figure 10. **The number of messages increases a little bit if each agent has averagely more acquaintances in acquaintance networks.**

Information sharing efficiency with or without utilizing *path* to update probability matrix The third experiment is designed to determine how the efficiency is impacted by using information in *path* to update probability matrix. We set $rel(i,j) = 0.8$. The result is shown in Figure 9. Notice that this small change almost halves the number of messages required to deliver information from source to sink in different acquaintance network. This result demonstrate that by utilizing *path*, information sharing efficiency has been greatly enhanced.

The influence of average acquaintances In next experiment, we looked in detail at exactly how the number of acquaintances can help to make the information sharing efficient. We run experiments with $rel(i,j) = 0.8$ and in acquaintance networks each agent has average acquaintances from 2 to 8. The result in Figure 10 shows the more number of acquaintances, there have to be more messages to delivery i which means the the information sharing cannot be enhanced by connect agent with more acquaintances. Moreover, in our experiment, we don't consider the limitation of communication breadth for agent members.

Algorithm efficiency among different size teams To investigate the influence of team scale on information sharing performance, in Figure 11, we ran experiments using different sizes of agent team from 100 to 550 with rel(i,j)=0.7. The information sharing efficiency is measure as the percentage of agents involved for information sharing use $percentage = \frac{agents\ involved\ infodelivery}{Total\ \#\ of\ agent\ team}$. Experiment result shows with different sizes of teams, the efficiency of information sharing is almost the same which indicates that the team size is not a factor which influence the information sharing efficiency.

Information sharing with different types of team organization From these experiments, we can find not only that our algorithm works on each types of acquaintance networks but also some clues that how these network topologies influence the efficiency of our in-

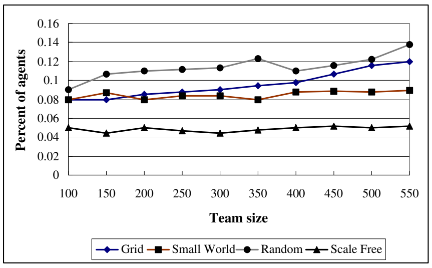

Figure 11. Information sharing algorithm works even slightly better on large scale teams according to the measure of percentage.

formation sharing algorithm. We notice that networks with small average distance (random, small world and scale free network) are always outperform the regular grid network which has not such a property. Moreover scale free network with power law distribution is clearly superior to others which do not possess this character. The difference between different acquaintance topologies is distinct when the previous messages have a strong relationship with i, for example in Figure 7, when rel(i,j)=1, the number of messages needed to delivery i in scale free network is only one third as many as it in grid network.

6. Related Work

Most related work can be classified into one of several major categories. The first strand of research is based on a centralized model or distributed model where there are agents such as match maker, information broker or message broad who can response for all information communication [4,6]. These works has been shown to be able to greatly improve multiagent system performance [9]. However, such work is inadequate for large team, since it is impossible or undesirable for all team member to share all their information all the time, i.e. because of the limit of required communication channels.

The second major strand of research is relies on agents maintaining a shared model of each other or knowing exactly other members' actual internal state as STEAM[15], COM-MTDP [14] and CAST [19]'s mental model. However, as for centralized approaches, in large team there is insufficient bandwidth to support such an approach.

The information sharing problem can also be handled by setting up decentralized model. Both [18] and [7] did a communication decision model based on Markov decision processes (MDP). Their basic idea is an explicit communication action will incur a cost and they supposed the global reward function of the agent team and the communication cost and reward are known. Moveover, [8] put forward a decentralized collaborative multiagent communication model and mechanism design based on MDP which assumed that agents are full-synchronized when they start operating, but no specific optimal algorithm was presented. Unfortunately, there are no experimental result showing that their algorithm can work on large teams.

Incomplete information theory is another way to solve the information sharing problems. [3] presents a framework for team coordination under incomplete information based on the incomplete information game theory that agents can learn and share their estimates with each other. [17] used a probability method to coordinate agent team without explicit communication by observing teammates' action and coordinating their activities via individual and group plan inference.

Research on social networks began in physics[1, 11, 16], but since it has been applied in many areas though rarely in multiagent work.

7. Summary and Future Work

In this paper, we present a novel approach and initial result to the challenges represented by sharing information to coordinate large size agent team. Especially, we presented a basic architecture for flexible, distributed information exchange among large teams. Our key information sharing algorithm was encapsulated in this architecture. This algorithm utilizes relationships between pieces of information to enable efficient information sharing. An analysis of the influences of different acquaintance network structure on the efficiency showed that social network can lead to better performance. Our experiment results show that scale free network is the best acquaintance network topology for information sharing. However, our initial experiments reveal that while our algorithms are capable of dealing with some of the challengers of the domain, many challengers remain. A major issue we leave for future research is how to calculate the relationships between pieces of information which is highly relative with domain knowledge and expertise where our algorithm should be applied. Furthermore, we do not investigate how information sharing works on negative relative messages where the relationship between piece of information is less than 0.5.

References

[1] R. Albert and A. Barabasi. Statistical mechanics of complex networks. *Review Modern Physics*, 74, 47, 2002.

[2] A. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286, 509-512, 1999.

[3] H.H. Bui, S. Venkatesh and D. Kieronska. A framework for coordination and learning among team members. *In Proceedings of the Third Australian Workshop on Distributed AI (DAI-97)*, pages 116-126, Perth, Australia.

[4] M. H. Burstein and D. E. Diller. A framework for dynamic information flow in mixed-initiative human/agent organizations. *Applied Intelligence on Agents and Process Management*, 2004. Forthcoming.

[5] P. R. Cohen, H. J. Levesque and I. Smith. On team formation. *In J. Hintikka and R. Tuomela, editors, Contemporary Action Theory*, Synthese, 1998.

[6] K. Decker, K. Sycara, A. Pannu and M. Williamson. Designing behaviors for information agents. *Procs. of the First International Conference on Autonomous Agents*, Feb., 1997.

[7] C.V. Goldman and S. Zilberstein. Optimizing information exchange in cooperative multi-agent systems. *Proceedings of the Second International Conference on Autonomous Agents and Multi-agent Systems*, 2003.

[8] C.V. Goldman and S. Zilberstein. Mechanism design for communication in cooperative systems. *Game Theoretic and Decision Theoretic Agents Workshop at AAMAS'03*, July, 2003.

[9] K. C. Jim and C.L. Giles. How communication can improve the performance of multi-agent systems. *In Proceedings of Autonomous agents'01*, 584-591, 2001.

[10] S. Milgram. The small world problem. *Psychology Today*, 22, 61-67. 1967

[11] M. E. J. Newman. The structure and function of complex networks. *SIAM Review,* Vol. 45, No. 2, pages 167-256, 2003.

[12] P. Scerri, Y. Xu, E. Liao, J. Lai, M. Lewis, K. Sycara. Coordinating very large groups of wide area search munitions, *Recent Developments in Cooperative Control and Optimization, Dordrecht*, NL: Kluwer Academic Publishers.

[13] P. Scerri, Y. Xu, E. Liao, J. Lai, K. Sycara. *Scaling Teamwork to Very Large Teams,* AAMAS 04, Forthcoming, 2004.

[14] D. Pynadath and M. Tambe. The communicative multiagent team decision problem: analyzing teamwork theories and models. *Journal of Artificial Intelligence Research*, Vol.16, pages 389-423, 2002.

[15] M. Tambe and W. Zhang. Towards flexible teamwork in persistent teams: Extended report. *Journal of Autonomous Agents and Multi-agent Systems*, special issue on "Best of ICMAS 98".

[16] D. Watts and S. Strogatz. Collectivedynamics of small world networks. *Nature*, 393:440-442, 1998.

[17] M.V. Wie. A probabilistic method for team plan formation without communication. *Proceedings of the Fourth International Conference on Autonomous Agents*, pages 112-113, Barcelona, Spain, June 3-7, 2000.

[18] P. Xuan, V. Lesser and S. Zilberstein. Communication decisions in multi-agent cooperation: Model and experiments. *In Proceedings of Autonomous Agents'01*, 2001.

[19] J. Yen, J. Yin, T. R. Ioerger, M. S. Miller, D. Xu and R. A. Volz. Cast: Collaborative agents for simulating teamwork. *In Proceedings of IJCAI'01*, pages 1135-1142, 2001.