A Scalable Agent Location Mechanism *

Onn Shehory
The Robotics Institute
Carnegie Mellon University
5000 Forbes Ave., DH 3311
Pittsburgh, PA 15213 USA
Tel: +412-268-3740
onn@cs.cmu.edu

Abstract

Large scale open multi-agent systems where agents need services of other agents but may not know their contact information require agent location mechanisms. Solutions to this problem are usually based on middle-ware such as matchmakers, brokers, yellow-pages agents and other middle agents. The disadvantage of these is that they impose infrastructure, protocol and communication overheads, and they do not easily scale up. We suggest a new approach to agent location, which does not require middle agents and protocols for using them. Our approach is simple and scales up with no infrastructure or protocol overheads, thus may be very useful for large scale MAS. In this paper, we analytically study the properties of our approach and discuss its advantages.

1 Introduction

Multi-agent systems (MAS) are taking an increasing role in the solution of highly distributed computational problems in dynamic, open domains. We assume that *large-scale* open MAS will be an inevitable part of this trend. The size of such systems poses problems which do not exist, or may be neglected in small-scale MAS. These usually stem from two major sources: (1) communication costs which are commonly (at least) polynomial in the number of agents, resulting in low performance; (2) task and resource allocation require a solution of an optimization problem of exponential complexity.

Several approaches were suggested to address these problems. For instance, the complexity of the task allocation problem in MAS is reduced via, e.g., coalitions of bounded size [7]. In other research, cooperation with reduced communication is suggested [2]. Communication reduction is also discussed in [8], where

*This material is based on work supported in part by MURI contract N00014-96-1222 and CoABS Darpa contract F30602-98-2-0138.

a mechanism for coordination in large-scale MAS with constant communication complexity is presented. These (and other) suggested solutions to the problems above refer, in many cases, to homogeneous agents. Yet, solutions that refer to agents with heterogeneous capabilities assume that agents either know all other agents they need to interact with (this is a closed MAS), or are provided with some agent location mechanism to find agents they need but do not know about in advance (e.g., middle agents [1], matchmaking [5], facilitation [3, 6]).

In agreement with previous research, we, too, perceive agent location mechanisms as necessary for open MAS. In such systems, agents with different expertise may need other agents to provide them with services. However, they may not know the contact information of the service providers. An agent location mechanism provides the agents with this missing information. In small MAS, it is sometimes possible for agents to maintain a list of all possible agents, however in large scale open MAS this is infeasible. Middle agent mechanisms for large scale open MAS are suggested in [4], where distributed matchmaking is presented. This solution, however, introduces two types of overheads: (1) each communication operation going out to another agent is preceded by communication with a matchmaker, and may also fire a series of communication operations between the distributed matchmakers; (2) there is a need for an additional computational infrastructure, in terms of matchmaker agents as well as protocols for other agents to use these matchmakers.

In this paper we suggest an agent location approach for large open MAS with no need for middle agents, thus relaxing the second type of overhead. In the following section we provide the details of our approach. In Section 2 we present the problem, then we introduce our approach to its solution (in Section 3). Section 4 describes the model that we use for analysis of our approach. In Section 5 we analyze the approach and compute and present its advantageous properties. Finally, in section 6, we conclude and present open problems and future directions.

2 The Problem

Assume an open MAS which includes heterogeneous agents, where availability of the agents varies and new agents may be added dynamically. Heterogeneity is expressed in terms of different expertise and different capacities. The agents in the MAS need to perform tasks. Tasks may be given in advance, but may also arrive dynamically. One of the characteristics of a task is the expertise necessary for its performance. We assume that agents may cache some information with regards to the attributes of other agents, their availability and their location. Though, we assume that this local information (and in particular location and availability) may be incorrect due to dynamics of the system and the environment in which it is deployed. We also assume that, at least in some cases, an agent receives tasks that it cannot perform (due to incompatible expertise or capacities), but it does want to perform them. This results in a need to cooperate with other agents, and in particular it is necessary for an agent to either know or be able to find agents that have the right expertise for the tasks it cannot perform.

Knowing other agents and being able to find them are supported in MAS in two major ways:

- Agents maintain a list of all other agents. In close systems, where all of the agents are known in advance, this is rather simple, although for very large systems may be space expensive. In open system, where agents may dynamically appear and disappear, a list of all agents cannot be maintained. If all of the possible agents are known, it is possible to hold a list of all possible agents, however if it is unknown which new agents may appear, no complete list can be constructed.

- In open MAS, middle agents [1] are a common agent location mechanism. These provide other agents with agent location services. An open MAS may have a single middle agent or multiple ones. In the first case, the location mechanism is centralized, thus may result in a need for a very large space for storage as well as a single point of failure. In the second case, there is a need to implement a mechanism for maintaining some level of coherence between the multiple middle agents. Both cases require the overhead of creating and maintaining middle agents, and some protocols for the other agents for interaction with the middle agents.

In this paper, we stress that in large-scale open agent systems there is a solution that eliminates the need for middle agents, thus prevents the need to create and maintain them. In fact, we suggest that some of the middle agents' activity can be avoided, incurring a very low cost to the rest of the agents, and that distribution of the rest of the activity among the other agents is simple to perform and yet, provides a good agent location mechanism.

3 The Approach

Our approach is rather simple: we require that each agent i hold a list L_i of other agents it knows. The list shall include information regarding names, addresses, expertise and other relevant information about other agents. The list may change dynamically, but it is not necessarily up to date or correct: it is an incomplete, inaccurate view of i's of the rest of the agent community. In this paper we assume that the frequency of change is slow enough and the reliability of messages is high enough, so that the lists agents hold, although incomplete, are mostly accurate and up to date, with a small fraction of erroneous entries.

Denote the number of agents in the system by n. In principle, L_i may include all $n-1$ other agents, but this is too costly when n is very large. In an open MAS, it may also be impossible for an agent to know all of the agents all of the time. We suggest that agents hold L_i such that $|L_i| \ll n$. When an agent needs to locate another agent for which it does not have the location information in its local list, it will consult (either some or all) agents on its list for such information. These, in turn, will perform the same procedure recursively. Motivating the agents to cooperate on this agent location is not the focus of the research presented here, but if necessary one can devise a protocol to guarantee such cooperative behavior (e.g., via some payment schemes). A unique request i.d. will prevent an agent from handling a request more than once and from request cycles (since it will allow an agent to avoid a location request that originated from itself). In the worst case, this search will cover the whole agent community, i.e., $n-1$ agents, with communication complexity $O(n)$ for the whole system (which implies an average $O(1)$ per agent, however the partition is usually not equal). The average case is much better, and by adding some heuristics for descriminatively selecting agents on the contact list, communication complexity of $O(1)$ can be achieved. However, we show that even without such heuristics, an appropriate selection of the size of L_i will result in a very low exploration depth, implying a very low communication complexity. To simplify things, we will show that a not too large contact lists L_i will allow agent location via very few communication operations, without any additional mediation services.

4 The Model

To illustrate the connections among agents as reflected by their contact lists we represent the agent society as a directed graph. Each node i in the graph represents agent i and each edge (i, j) represents the fact that i holds in L_i contact information of j, or, in simple words—i knows j. For simplicity of representation and analysis we first refer to a planar, undirected graph with a rectangular lattice pattern (see Figure 1). Such a graph represents an agent society where each agent knows exactly its 4 close neighbours. Below, we analyze the properties of such a connection structure. From this analysis we later draw conclusions with regards to more complex structures. One may assume that, if $|L_i| = O(1)$, the location of other agents, for large n, will be very costly (regarding communication) or even impossible (since there may be some disconnected cliques of agents). We shall examine this assumption through our analysis.

Denote the number of nodes by n, the number of edges by e and the degree of a node by d. The distance between two nodes is the number of edges in the shortest path between them. In the planar, rectangular graph we study, $e = 2n$ and $d = 4$. We are interested in the average distance between nodes. This distance will dictate the depth of the agent location search required by our approach and, correspondingly, the number of communication operations required. Without loss of generality, let us compute the average distance of all nodes from a specific node A. A has 4 nearest neighbours at distance 1, 8 neighbours at distance 2 and, continuing in the same fashion, $4k$ neighbours at distance k. Since we assume that n is very large, we are not interested in the particular shape of the borders of the graph. For large n, the portion of nodes which are close to the border is negligibly small (based on a ratio of perimeter to area, which converges to zero). Therefore, omission of nodes near the borders

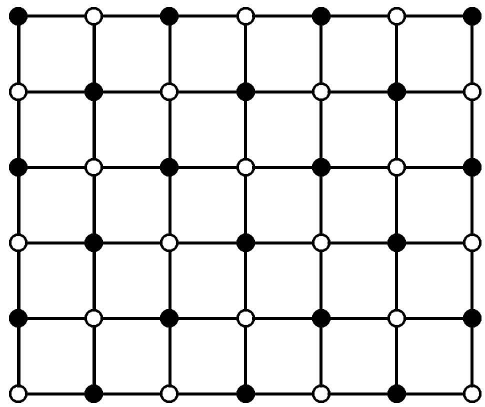


Figure 1: A segment of a planar rectangular lattice structure connectivity graph.

of the graph will have little effect on distance analysis. Hence, to simplify this analysis, we refer to a graph in which there are exactly $4k$ nodes at distance k from a center node (intuitively, this means that there are no "holes" within and no "rough" borders). Such a graph allows for a simple expression of the relation between the number of nodes n and the maximal distance from the center node (denote this distance by m), as follows:

$$n = 1 + \sum_{i=1}^{m} 4i = 2m(m+1) \tag{1}$$

which is a sum over the center node and all of its neighbours in all distances. The average distance $\bar{l}$ from a center node, for any perfect planar lattice structure, is given by

$$\bar{l} = \frac{\sum_{i=1}^{m} l_i \cdot n_i}{n-1} \tag{2}$$

where l_i is the ith distance and n_i is the number of nodes at the ith distance. In particular, for the rectangular planar lattice structure, where $n_i = 4i$, and by substitution of equation 1, we have

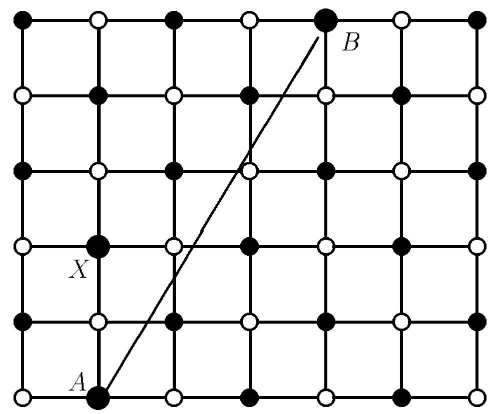
$$\bar{l} = \frac{\sum_{i=1}^{m} i \cdot 4i}{2m(m+1)} = \frac{2m(m+1)(2m+1)}{3 \cdot 2m(m+1)} = \frac{2m+1}{3} \tag{3}$$

which means, using equation 1 again, that

$$\bar{l} \sim \sqrt{n} \tag{4}$$

and this result holds for every perfect planar lattice structure as long as the degree on each node is a constant. In a three-dimensional lattice this result will change to $\bar{l} \sim \sqrt[3]{n}$, and this can be further generalized to a k-dimensional

Figure 2: A connectivity graph segment with a single shortcut.



lattice, where $\bar{l} \sim \sqrt[k]{n}$. Note, however, that high dimensionality is undesirable since it results in each node having a very large number of adjacent nodes (for dimension k, this number will be 2^k). When referring to an agent community, this requires that every agent i holds a large list L_i of known agents, which is not always feasible.

The observation that the average distance from a center node is $\sim \sqrt{n}$ is worrisome. It implies that the approach that we propose may incur a search cost of $\Theta(\sqrt{n})$, regardless of our choice of the (constant) size of adjacency lists L_i. This complexity is unacceptable for large n.

So far, we learned that our approach requires a search to a depth of the average path length $\bar{l}$ and of breadth $|L_i|$. Our analysis shows that for perfect lattice structure connectivity graphs, for large n, either the size of $\bar{l}$ is too large, or $|L_i|$ is too large. Hence, the proposed approach will fail due to the incurred complexity. Given these limitations, we need to address the following questions:

- What structural organizations, if any, can result in $\bar{l}$ and $|L_i|$ both small enough to provide an acceptable search complexity for large n?
- Is any of these organizations applicable for MAS, and can result in a good enough agent location mechanism?

5 A Simple Complexity Reduction

We show here that via a simple, almost negligibly small change in the organizational structure of the society, a very significant complexity reduction can be achieved. While $|L_i|$ is almost unchanged, $\bar{l}$ can be significantly reduced.

Suppose we connect two arbitrary nodes, A and B in the graph by an edge (A, B), as in Figure 2. How does such a connection affect the search complexity? Since so far we had $e = 2n$ and now we have $e = 2n+1$, one may assume that the improvement will be proportional to $1/n$. Let us examine this more carefully. In the worst case, A and B were neighbours, so we gain no improvement. The best case seems to be the one where the distance between A and B is $2m$, twice the maximal distance from a center node. We analyze below this case, referring to Figure 2 as an example.

Refer to the points A, B, and a third arbitrary point X which is, without loss of generality, closer to A. Denote the distances between the points by L_{AB}, L_{AX}, L_{BX}, where the subscripts refer to the points, and undirectedness is assumed. In our example (Figure 2), $L_{AB} = 8, L_{AX} = 2, L_{BX} = 6$. By adding a connecting edge between A and B, we get $L_{AB}^{new} = 1$ and, for (almost) any point X which is closer to A than to B, $L_{BX}^{new} = L_{AX}^{new} + 1 > L_{BX}$.[1] This means that the average distance of (almost) half of the nodes in the graph has been reduced via node A. For reasons of symmetry, the other (almost) halt of the node will improve via B. Altogether, virtually all of the distances in the graph have been reduced by adding a single new connection. This is an impressive result, but it raises two questions:

1. How good is the single improvement? That is, what is the relative reduction in the average distance $\bar{l}$?

2. If we add more connections, will they provide the same improvement? If not, who much will they improve?

5.1 Analysis of the Single Improvement

Assume a node Y is a neighbour of A.

Proposition 1
1. Prior to adding new edges to the graph, $L_{BY} = L_{AB} + \Delta$, *where* $\Delta \in \{-1, +1\}$.
2. The average distance of all 4 neighbours of A *from* B *is, prior to changes,* L_{AB}, *where* $0 < \epsilon \ll 1$, *and* $\lim_{n\to\infty} \epsilon = 0$.

Proof.
1. A neighbour of A is one edge away from A. This results in moving one edge further from B or one edge closer to B (thus $\Delta \in \{-1, +1\}$).
2. In most cases, there are two neighbours closer to B and two further. For these cases the average is exactly L_{AB}. For nodes A located on a path which is a straight line from B, there are three neighbours further and only one closer to B, with an average of $L_{AB} + \frac{1}{2}$. The number of such nodes in the graph is inversely proportional to the number of nodes in the graph, i.e., it is $O(\frac{1}{n})$ (or $\frac{c}{n}$, c a constant), hence the average will be

$$\frac{(n-1)L_{AB} + c(L_{AB} + \frac{1}{2})}{n} = L_{AB} + \frac{c-1}{n} L_{AB} + \frac{c}{2n} \tag{5}$$

[1] A small portion of the points will have $L_{AX}^{new} + 1 = L_{BX}$, but their relative number is negligibly small.

Denote $\frac{c-1}{n}L_{AB} + \frac{c}{2n}$ as ϵ, we have

$$\lim_{n\to\infty} \epsilon = \lim_{n\to\infty} \left(\frac{c-1}{n} L_{AB} + \frac{c}{2n} \right) = 0 \tag{6}$$

$\square$

After adding edge (A, B), all 4 neighbours of A's are at distance $L_{BY}^{new} = 2$ (one edge to A, and one from A to B).

In a similar way we analyze the 8 secondary neighbours. Denote a secondary neighbour of A's by Z. $L_{BZ} = L_{AB} + \Delta$, where $\Delta \in \{-2, -1, 0, +1, +2\}$, and the average L_{BZ} prior to changes is $L_{AB} + \epsilon$. After adding (A, B), $L_{BZ}^{new} = 3$. Generalizing this, we can show that

Proposition 2
1. The number of neighbours at distance i from A is $n(i) = 4i$.
2. The average distance of A's neighbours at distance i from it, from B, is $L_{AB} + \epsilon$ prior to changes and $i + 1$ after adding edge (A, B).

The proof is similar to the above and is not presented here. Improvement in distance applies to all $n(i)$ neighbours (only) up to a distance of $\lfloor \frac{L_{AB}}{2} \rfloor - 1$ from A, since the distance of some of the neighbours further away from A is smaller than or equal to $\lfloor \frac{L_{AB}}{2} \rfloor$ before adding (A, B).

Below, we sum up the improvement I_d^1 in distance over all of the neighbours at distance up to $\lfloor \frac{L_{AB}}{2} \rfloor - 1$ from A. Although this is only part of the improvement, it represents about half of the nodes that experience an improvement, and significantly more than half of the cumulative improvement. Hence, this sum is sufficient for analyzing the order of improvement and provides a worst case bound on it. The sum is as follows:

$$I_d^1 = \sum_{i=1}^{\lfloor \frac{L_{AB}}{2} \rfloor - 1} n(i) \cdot (L_{AB} - i) = \sum_{i=1}^{\lfloor \frac{L_{AB}}{2} \rfloor - 1} (4iL_{AB} - 4i^2) \simeq \frac{1}{3} L_{AB}^3 \tag{7}$$

So far (according to equation (5)), we know that I_d^1 (1 stands for a single node) is at least $\frac{1}{3}L_{AB}^3$ (in fact, we know it is closer to $\frac{1}{2}L_{AB}^3$, but the proof is complex and the difference does not affect our analysis). However, we do not know what the ratio between this improvement and the overall sum of distances prior to improvement is (denote this sum by D). We may compute D explicitly, however via a much simpler analysis we can provide upper and lower bounds which, although not tight bounds, are sufficient for our discussion.

Proposition 3
If the shape of our lattice-like graph were to be a circle of radius R, then the average distance $\overline{d}$ within the graph will be at most $\sqrt{2}R$ and at least $\frac{\sqrt{2}}{2}R$.

Proof.
1. $\overline{d} \geq \frac{\sqrt{2}}{2}R$: the distances of nodes measured from the center node of the graph are bound between maximal distance of $\sqrt{2}R$ and minimal distance 0.

The number of nodes increases linearly with the distance, so there are more nodes located further than closer. Therefore the average must be greater than $(\sqrt{2}R+0)/2 = \frac{\sqrt{2}}{2}R$. $\square$

2. $\overline{d} \leq \sqrt{2}R$: for a node on the edge of the graph, the maximal distance is $2\sqrt{2}R$ and minimal distance is 0. Here, the distribution of node distances is symmetric around the center value. The average is the center value, i.e., $\sqrt{2}R$. $\square$

Previously, we denoted the maximal distance by $2m$. In the circle shaped graph, $2m = 2\sqrt{2}R$. Within a circular lattice-like graph, the relation between the number of nodes and the radius is given (roughly) by $n = \pi R^2$. Since, as we have shown above, the average distance is bound, that is, $\frac{\sqrt{2}}{2}R \leq \overline{d} \leq \sqrt{2}R$ and $l_{max} = 2\sqrt{2}R$, we have $\frac{1}{4}l_{max} \leq \overline{d} \leq \frac{1}{2}l_{max}$. Explicit computation can show that $\overline{d} \approx \frac{3}{8}l_{max}$. The number of distances between nodes is $n_{dist} \approx \frac{n^2}{2}$, and therefore the sum of all distances is

$$D = n_{dist} \cdot \overline{d} = \frac{3}{16}n^2 l_{max} = \frac{3}{16}n^2 2\sqrt{2}\frac{\sqrt{n}}{\sqrt{\pi}} \tag{8}$$

In our analysis L_{AB} was selected to be m (which is $\frac{l_{max}}{2}$). The improvement is therefore (at least)

$$I_d^1 = \frac{1}{3}L_{AB}^3 = \frac{1}{3}m^3 = \frac{1}{3}(\sqrt{2}R)^3 = \frac{2\sqrt{2}}{3}\left(\frac{\sqrt{n}}{\sqrt{\pi}}\right)^3 \tag{9}$$

The relative improvement is the ratio between the improvement and the original sum of distances, which is

$$\frac{I_d^1}{D} = \frac{\frac{2\sqrt{2}}{3}\left(\frac{\sqrt{n}}{\sqrt{\pi}}\right)^3}{\frac{3}{16}n^2 2\sqrt{2}\frac{\sqrt{n}}{\sqrt{\pi}}} = \frac{16}{9\pi n} \simeq \frac{1.8}{n} \tag{10}$$

This provides a worst case bound on the improvement in the distance of a single node. Intuitively, if a single node improves by order of $\frac{1}{n}$, the improvement of the whole system, where there are n nodes, should be $O(1)$. Below we compute this improvement explicitly to examine this hypothesis.

An improvement in the distance between a node X in the neighbourhood of A and B implies an improvement in the distance between X and nodes Y in the neighbourhood of B. We formally express these improvements and then sum them up. Denote the distance between X and B by ΔB. As shown above, for X at distance i from A, $\Delta B = L_{AB} - i$. Denote the distance between X and a neighbour Y of B which is at distance k from B by $\Delta_k B$. The distance between X and Y is then $\Delta_k B = \Delta B - k = L_{AB} - i - k$. We sum over all relevant[2]

[2] Relevant nodes are nodes for which an improvements in guaranteed. Improvements are not guaranteed for nodes too far from A and B, since the original distance between such X and Y may be shorter than the distance that results from the addition of edge (A, B).

nodes to arrive at the cumulative distance improvement I_d

$$I_d = \sum_{i=1}^{\lfloor \frac{L_{AB}}{2} \rfloor - 1} \sum_{k=1}^{\lfloor \frac{L_{AB}}{2} \rfloor - i - 1} n(k)n(i)(L_{AB} - i - k) \tag{11}$$

Since $n(k) = 4k$ and $n(i) = 4i$, we have

$$I_d = 16 \sum_{i=1}^{\lfloor \frac{L_{AB}}{2} \rfloor - 1} i \sum_{k=1}^{\lfloor \frac{L_{AB}}{2} \rfloor - i - 1} k(L_{AB} - i - k) = \frac{1}{40} L_{AB}^5 \tag{12}$$

We transform this to terms of n:

$$I_d = \frac{1}{40} L_{AB}^5 = \frac{1}{40} m^5 = \frac{1}{40} (\sqrt{2}R)^5 = \frac{1}{40} 4\sqrt{2} \left(\sqrt{\frac{n}{\pi}} \right)^5 = \frac{\sqrt{2}}{10} \frac{n^{2.5}}{\pi^{2.5}} \tag{13}$$

Note that this sum is very conservative: to simplify the analysis, we excluded from the sum half of the nodes to avoid the (fewer) nodes that exhibit no improvement. The value of I_d we arrived at is roughly half of the actual cumulative improvement. The relative cumulative distance improvement is given by:

$$\frac{I_d}{D} = \frac{\frac{\sqrt{2}}{10} \frac{n^{2.5}}{\pi^{2.5}}}{\frac{3}{16} n^2 2\sqrt{2} \frac{\sqrt{n}}{\sqrt{\pi}}} \simeq \frac{1}{37} \tag{14}$$

This result confirms our hypothesis, according to which the improvement is $O(1)$. But its importance lies in the actual number: it means that (even with our very conservative analysis) *the cumulative relative improvement that results from a single added edge is of* 2.7%. The more realistic number is around 5%!

5.2 Improvement via Multiple Edges

At this point, we know that a single additional node provides a very significant improvement, however we need to now how well will additional edges contribute to distance improvement. Following the guidelines of the analysis in the previous section we have analyzed the case of a second added edge. The result was rather surprising: the improvement provided by a second added edge is numerically similar to (though slightly smaller than) the improvement provided by the first one. This implies that it is sufficient to add two edges to arrive at an improvement of 10%. The analysis for the third edge provides a less impressive result: it contribute only half of the previous contributions, i.e., $\sim 2.5\%$. Via informal deliberation one can show that the next additional forth to sixth edges will contribute at most 2.5% each but probably slightly less that this. The seventh added edge will contribute about half of the contribution of the sixth, and this behavior can be projected to further added edges.

The results above mean that in order to arrive at a significant relative distance improvement one needs to add dozens or hundreds of additional edges to the graph. Recalling that we refer to (and analyze) large n only, the number of added edges is insignificant, since it will add very little to the degree of each node. For example, if there are $n = 10,000$ nodes then we initially have an average distance

$$\bar{l} = \frac{3\sqrt{2}}{4\sqrt{\pi}}\sqrt{n} \simeq 60 \tag{15}$$

By adding $\sim$ 2000 edges to the graph the average distance between nodes is reduced by about 90% to ~ 6. The average node degree changes by only 0.2.

5.3 Applicability

What are the implication of these results to our agent location approach? In terms of communication complexity, the results above imply that a vast reduction in the average distance (and correspondingly in the depth of search and the number of communication operations) can be achieved via a very small addition to the connection lists L_i. However, the analysis was performed for a lattice-like rectangular graph. It is not difficult to show that similar analyses will hold for other lattice structures. The question is, how well will these results apply for less structured organizations, more complex graphs? In this paper we do not provide analysis of such cases. Yet, more complex graphs may be viewed as graphs where multiple additional edges were added to a simple graph. If these edges are added in a random manner, and their number is large enough, a short average distance is guaranteed. Thus the class of complex, moderately to highly connected graphs inherently exhibits short communication paths. Graphs which are excluded from our analysis are those in which unconnected (or very loosely connected) subgraphs exist.

It is important to note that, in practice, we do not suggest that edges be added to graphs. In many cases the agent connection topology is rich and complex enough in the first place. Thus, it may provide the low communication complexity of our location approach without applying changes. In simple words—our agent location mechanism and its low complexity are applicable to a large class of MAS with no need for changes in the agents' location lists.

6 Conclusion

We have shown that in a multi-agent system where agents cache a list of agents they know, it is possible for agents to locate unknown agents (i.e., agents not on their list) without using middle agents, and yet with a comparatively low communication complexity. By this, one can avoid the overhead of implementing middle agents and protocols for other agents to use the middle agents' services. With a careful design and maintenance of the local lists L_i of known agents, the communication complexity of our approach is rather low. This makes our approach feasible for implementation in large scale MAS. Our approach is very simple and introduces very little protocol and mechanism overhead.

There are several issues that still require investigation. Within the class of lattice-like graphs, we need to study analytically the asymptotic behavior of the average distance as a function of the number of added edges. We also need to explicitly analyze the properties of less structured graphs. Based on these results, it would be useful to come up with efficient (and maybe optimal) strategies for maintaining connection lists that guarantee low average distances and low communication complexity for agent location in large MAS.

References

[1] K. Decker, K. Sycara, and M. Williamson. Middle-agents for the internet. In *Proceeding of IJCAI-97*, pages 578–583, Nagoya, Japan, 1997.

[2] E. Ephrati, M. Pollack, and S. Ur. Deriving multi-agent coordination through filtering strategies. In *IJCAI95*, pages 679–685, Montreal, Quebec, Canada, 1995.

[3] Michael R. Genesereth and Steven P. Ketchpel. Software Agents. *Communications of the ACM*, 37(7):48–53, July 1994.

[4] S. Jha, P. Chalasani, O. Shehory, and K. Sycara. A formal treatment of distributed matchmaking. In *Proceeding of Agents-98*, pages 457–458, Minneapolis, Minnesota, 1998.

[5] D. Kuokka and L. Harada. Matchmaking for information agents. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 672–679, Montreal, Quebec, Canada, August 1995.

[6] M. Nodine and A. Unruh. Facilitating open communication in agent systems: the infosleuth infrastructure. In M. Singh, A. Rao, and M. Wooldridge, editors, *Intelligent Agents 4*, Lecture Notes in Artificial Intelligence No. 1365, pages 281–296. Berlin:Springer-Verlag, 1997.

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

[8] O. Shehory, S. Kraus, and O. Yadgar. Goal-satisfaction in large-scale agent-systems: a transportation example. In *Proceeding of ATAL-98*, Paris, France, 1998.