Congregation Formation in Multiagent Systems

Christopher H. Brooks and Edmund H. Durfee*
Artificial Intelligence Laboratory
University of Michigan
Ann Arbor, MI 48109
({chbrooks, durfee}@umich.edu)

Abstract.
We present *congregating* both as a metaphor for describing and modeling multiagent systems (MAS) and as a means for reducing coordination costs. We compare congregations to other models of multiagent behavior, namely teams and coalitions. We present a formal model of a congregation and then apply Vidal and Durfee's CLRI framework [21] to determine the aspects of a congregating problem that make it difficult. We discuss how agents that are able to advertise congregations using labels can reduce coordination cost, and show how such agents can exploit structure within labels to further reduce search cost.

1. Introduction

In a multiagent system, self-interested agents must often decide which other agents they want to interact with. The nature of these interactions may vary; perhaps they wish to buy and sell goods, exchange information about their environment, group together in order to exploit scaling effects, or simply benefit from the presence of other agents. These interactions are what makes a society more than just a collection of agents which happen to be in the same location; each agent's reward is dependent upon the agents that it interacts with.

In a long-lived system, agents will make this decision as to who to interact with over and over. Also, as the number of agents in a system increases, the number of potential interactions that a particular agent must consider grows exponentially, since agents must potentially consider all groups of agents which they could be a part of. Consequently, something is needed to save an agent from making this expensive computation every single time it needs to interact with another agent. This solution should allow an agent to devote some initial energy to searching for suitable partners, with the promise of reduced search costs in future iterations. It should also provide a sort of institutional memory, so that like-minded agents can benefit from previous searches.

* This work was supported in part by an IBM University Partnership Grant and by the National Science Foundation under grant IIS-9872057

One way in which human societies have dealt with this problem is through the establishment of *congregations*. Human congregations include clubs, churches, marketplaces, university departments, and Usenet newsgroups. In all of these cases, members of these congregations have devoted some upfront cost to organizing and describing themselves so that they can both reap the long-term benefits of interacting without coordinating further and also attract new agents with whom they would be likely to want to interact.

Congregations were previously described in [1] and [2]. In this paper we define the concept of a congregation in greater detail and present experiments studying the problem of agents attempting to find the correct congregation. The problem of agents finding the correct congregation, which we shall refer to as the congregating problem, can be thought of as a distributed search problem. A number of independent problem solvers are attempting to search through a space of agent interactions, each with the goal of finding a system of congregations that satisfies its needs. Viewing the congregating problem as distributed search allows us to focus on two separate aspects of the problem: first, what is the result of the search? That is, does a stable set of congregations emerge as a result of this process, and, if so, how well does this set of congregations satisfy the needs of the agents involved? The system may not find a stable set of congregations at all, but either settle into a limit cycle, or perhaps oscillate about a chaotic attractor.

The second, and equally important, aspect of the congregating problem is the nonequilibrium behavior of the search. In other words, how difficult is it for an agent to find a suitable congregation? How long does it take, and how much effort or utility must be expended in this search? If an agent must spend a large cost to find or attract a suitable congregation, it may turn out that the effort of congregating outweighs any potential benefits. If we are to analyze the benefit of congregations as a means for helping agents to find suitable partners to interact with, we must consider nonequilibrium costs and payoffs.

Much of the previous work on multiagent coordination, such as RETSINA [18], along with work on coalition formation [16] and team formation [19], as well as work on game-theoretic negotiation [13], has assumed that agents have well defined roles within an organization or specific tasks which they perform. In contrast, a congregation is a more general structure. The distinction is that congregating agents expect to have a long lifetime, during which they take on different roles, perform different tasks, and interact with different agents. By devoting some initial effort to constructing an congregation in which an agent can easily locate other agents with desirable characteristics, a community of agents can then take advantage of this structure to avoid devoting future resources to coordination.

We present congregating as a multiagent learning problem; multiple agents are learning at the same time, and the global state of the system is an important factor in determining the utility of each individual agent. The notion of coordination as multiagent learning has also been discussed in [15], where cooperative agents learned to coordinate actions to achieve a joint goal, and in [7], where self-interested agents are able to each learn a Nash equilibrium strategy. A primary difference between those works and this paper is that the congregating problem focuses on a decision as to *who* to interact with, rather than what action to choose.

We begin by defining a congregation more precisely, providing characteristics necessary for congregating to be applied to an agent domain. We compare congregating to both coalition and team formation. We then present a formal model of congregations and show how Vidal and Durfee's CLRI model [21] for analyzing multiagent learning can be used to predict the difficulty of a congregating problem. We present the affinity group domain and argue for its usefulness as a domain for studying congregating, and then present experiments and analysis showing how the difficulty of congregating increases exponentially with the number of agents involved. We present a solution to this in the form of labels. We show how labelers using a flat label space can make the congregating problem dependent upon the number of labels, rather than the number of agents. We then demonstrate how the hierarchical structure of a space of labels can be exploited so as to allow it to be searched more efficiently, thereby reducing congregating cost.

2. What is a Congregation?

In this section, we provide a concrete definition of a congregation, including features of multi-agent problem domains that must be present for a group of agents to properly be considered as a congregation. We then provide some examples of congregating, and discuss the idea of congregating as a distributed search through a space of agent interactions and discuss how this can shed some light on the difficulties of congregating in particular domains.

2.1. A Definition of Congregations and Congregating

We begin by providing a definition of congregations and the congregating problem. A congregation is a set of one or more self-interested agents. Each agent must interact with others in order to satisfy its needs. The purpose of the congregation is to allow a member to find suitable partners for interaction more easily. A congregation is a long-lived entity; members may join and leave throughout its lifetime.

The congregating problem refers to the problem that each agent has in finding other agents to interact with that satisfy its needs. Since an agent's satisfaction with a congregation is dependent upon the presence of other agents, and each of those agents' satisfaction is dependent upon the presence of other agents, inducing a particular set of congregations to form can be quite difficult, even if all agents within a congregation would be happy with it once it had formed. In talking about the congregating problem, typically we will be interested in the membership of a collection or system of congregations, and the satisfaction of each member of those congregations. We can then compare different configurations of congregations, either by using preference aggregation methods or with game theoretic concepts such as pareto optimality, to assess the relative global desirability of a system of congregations.

2.2. Characteristics of Congregations

The definition of a congregation provided above is correct, but rather terse and abstract. In this section we expand upon this definition and provide five characteristics needed for a group of agents to be considered a congregation.

- Individual rationality. Each agent is assumed to have its own utility function. Agents will act solely to maximize their long-term utility, where "long-term" indicates that an agent will take a discounted estimate of future rewards into consideration when deciding with whom to congregate.

  Note that we do not require (nor do we expect to use) any notion of "group rationality." Groups of agents which receive a single lump payment as a result of the group's performance do not fit into our definition of congregations; these are more accurately described by existing work on coalition [16] or team [19] formation.

- Agents may voluntarily join or leave congregations. An essential facet of the congregating problem is that agents are free to join (or leave, or refuse to join or leave) a congregation at any time they wish. It is this decision problem (which congregation, set of congregations, or series of congregations should an agent join?) that is at the heart of this work.

- Another principal idea behind congregations is that an agent's satisfaction with a congregation is dependent upon the other members of the congregation. Since agents congregate in order to satisfy needs which they cannot satisfy alone, it seems reasonable to assume that their satisfaction will depend upon how well these needs are met. If agents are heterogeneous in their ability to satisfy these needs then agents will prefer to congregate with those agents which better satisfy their needs over those who do not.

- Agents will have repeated interaction and long-term existence. As noted above, the whole point of developing a congregation is to allow an agent to devote fewer resources to future decisions regarding who it should interact with. If an agent is making a one-shot decision, there is no value to exploring and learning information to use in future encounters.

- We also assume that agents must expend energy or pay a cost to search for suitable partners to interact with and to advertise their presence to other agents. This cost can vary depending upon (for example) the distance between agents, the number of agents reached, or the complexity of the message. Advertising messages may also take time to propagate through the system. If an agent is able to costlessly and instantaneously search through the space of all other agents to find suitable partners, then there is no need for it to spend any effort on forming congregations, since the primary function of a congregation is to reduce the long-term search and advertising cost.

2.3. Examples of Congregations

Congregations are a pervasive feature of human existence. Recently, they have begun to appear within the Internet and the World Wide Web, albeit in a much more *ad hoc* way. In this section we provide some examples of congregations containing both human and artificial agents.

One sort of congregation of human agents is a farmer's market. Buyers and sellers of produce come together at a regular time and place. The actual participants may change from week to week, but individuals that want to sell produce know that they will have a high likelihood of finding customers. Likewise, individuals interested in buying produce know that they are likely to find a good deal. Every participant is self-interested, but each needs the presence of some subset of the other participants in order to satisfy their needs. By devoting some initial investment to creating a suitable marketplace and announcing its presence to others (possibly in a distributed fashion), participants attract a congregation of like-minded individuals.

Within the Internet, a common type of congregation is a chat group, IRC channel, or Usenet newsgroup, where individuals with particular interests join together to share information about topics of common interest. Within this domain, two features of congregations become obvious: the relationship between a congregation's description and its composition, and the relationship between the congregation's composition and the utility received by participants. For example, a broadly titled newsgroup, such as "comp.ai", might receive a wide variety of participants, posting on a variety of topics. This can be beneficial in that participants are likely to find someone else with a similar interest, but also detrimental, since a large group can make it difficult to sort out interesting posts. As the group membership grows, the entrance of new members with unrelated interests may decrease the newsgroup's utility for current members. Conversely, a specifically labeled newsgroup, such as "comp.ai.shells", might have a membership whose interests are all very similar, but the total membership might be too small to make the newsgroup useful for its participants. Describing a congregation at a level of detail that attracts only the "right" members can be a difficult problem.

A third example, which has a more computational flavor, is that of constructing the appropriate set of auctions for information services. In the University of Michigan Digital Library Project (UMDL) [4], agents would join auctions to contract for the purchase and sale of information, such as scholarly articles or web queries. One decision that had to be made was determining how many auctions to construct and placing sellers in appropriate auctions. If auctions had too few participants, or sellers whose products were too similar, it might be hard for buyers to satisfy their needs. However, if auctions were too large, then problems of scalability, as well as an unrealistic burden on the buyers to evaluate large numbers of goods, emerged. The UMDL used a centralized agent, the Auction Manager Agent [12], to control the creation of auctions. However, in larger systems, a more decentralized approach might be more appropriate. This process of selecting the right set of auctions or markets for information goods is precisely the congregating process. Each buyer would like to be in a group with agents that are selling the good it needs at an acceptable price. Similarly, each seller would like to be in a group with buyers willing to pay at least its reserve price for its good. (These needs might be mutually exclusive.) Agents would also like to have some sort of stable structure, so that they don't have to search through the space of all auctions every time they wish to buy or sell.

2.4. Congregations vs. Teams and Coalitions

A question to ask regarding congregations is how they differ from some of the other well-known models for describing multi-agent interactions, specifically teams and coalitions. We argue that congregations describe a different sort of agent interaction from either teams or coalitions, one that is more general and long-lived.

The construction of teams of agents that collaborate to pursue some shared or common goal is a well-studied field of research within MAS. Singh, Rao, and Georgeff [17] define a team as a group of agents who are constrained to have a common goal. One notable example of research on teams and team-oriented programming is that of Tambe [19]. His STEAM framework implements a joint-intentions model of problem solving [3], and demonstrates how groups of agents can be dynamically constructed to work together at problem solving. Similar approaches to multiagent coordination can be found in the work of Jennings [8], who also uses joint intentions, and Grosz and Kraus [6], who use an abstraction known as SharedPlans.

All of these approaches differ from congregating in that they explicitly require agents to share some sort of joint goal to be achieved. As a result, agents' reward or utility is dependent upon the performance of the group as a whole. In contrast, congregating makes no such assumption. All agents are presumed to be completely self-interested. Each agent's utility function is completely local, and an agent will cooperate with other agents only as long as such cooperation increases its utility. Note that this does not prevent the formation of teams so long as team formation benefits all members of the team; the point is that agents cannot be assumed *a priori* to be cooperative and only interested in team goals.

A more self-interested approach to multiagent interaction and coordination is the coalition. Coalitions have been discussed both within the context of economics [10] and multi-agent systems [16, 9, 14, 22]. While some details vary among researchers, the standard definition of a coalition is a group of agents that have all agreed to work together to achieve a larger goal. Each agent is self-interested, and by participating in the coalition, it will receive a higher utility than if it did not participate in the coalition.

This concept is closer to congregating than team formation, since it gets at the notion that agents are self-interested and want to join a larger group to improve their utility. However, there are several differences between coalition formation and congregation formation. First, coalition formation is typically structured around the accomplishment of a single task, whereas a congregation is a more long-lived entity that can assist agents in the accomplishment of a number of tasks throughout their lifetimes. Second, unlike coalition formation, an agent may not know the identities of other congregation members when it decides whether to join or not. In fact, the particular membership of a congregation may change over time. Consider the farmer's market example: when a buyer of produce decides to go to the market, it does not know what other agents will be there, although it may have some predictions. In fact, the agents who are there may change from week to week, although it is likely that there will always be agents that fill particular roles, such as seller of tomatoes or buyer of flowers. Third, there is not necessarily a notion of a joint activity or task, although one may exist. In the farmer's market example, the congregation serves to allow agents to come together and make bilateral exchanges. There is not a joint goal that the agents cooperate to pursue, except perhaps for abstract concepts such as the creation of more efficient markets.

2.5. Congregating as Distributed Search

As was discussed previously, it can be helpful to consider congregating as a form of distributed search. The state space of the search is the set of all possible congregations of agents. As each agent moves from one congregation to another, the search moves from state to state. Of course, no single agent has absolute control over the state transition. This can make a search process difficult to conduct, since particular actions by several agents may be needed in order to move to particular portions of the state space.

Congregating can potentially provide a means of coordinating this transition without central control. If agents tend to return to congregations where they have previously had successful interactions, congregation membership will tend to change more gradually, rather than in wild bursts. The greater an agent's reliance on past history as a predictor of future success, the less likely the system of congregations is to exhibit wild swings in congregation membership. However, too great a reliance on past history can prevent agents from conducting a satisfactory amount of exploration. In systems where agents move, a dependence on past history can also lead to errors in evaluating the utility of current actions. This is discussed further in section 4.

Viewing congregating as distributed search also presents us with a useful set of questions with which to evaluate a particular congregating process. For example, we can ask how long it takes the process to reach an "optimal" set of congregations, where optimal may be defined by an external source. We can also ask how efficient a particular state space trajectory to reach a set of congregations was. In other words, do the agents attempting to congregate incur large penalties while congregating as a result of a particular process? If there are several states that a process will converge to, we can compare them either as equilibrium states, asking what their value is to the participating agents, or as the goal states in a search process, asking how beneficial this search trajectory is to each agent in the process. In this paper, we will primarily concern ourselves with the question of how long a given search takes to complete.

3. A Formal Model of Congregations

In this section we provide a formal model of congregating and the congregating process. This will later be connected to the CLRI model to analyze the difficulty of the congregating problem as characteristics of the problem change.

We begin with a set $Ag = \{ag_1, ag_2, ..., ag_n\}$ of agents. This set is composed of two subsets; a set $CA = \{ca_1, ca_2, ..., ca_m\}$ of *congregators* and a set $LA = \{la_1, la_2, ..., la_k\}$ of *labelers*. Each agent must be a labeler or a congregator, but not both; that is, $CA \cup LA = Ag$ and $CA \cap LA = \varnothing$.

Also, consider a set $\alpha = \{\alpha_0, \alpha_1, ..., \alpha_n\}$ of *loci*. A locus is any place in which a congregation can form. Formally, a locus has a label (defined below) and a unique name. Labelers may assign labels to a locus in order to attract a particular set of congregators.

Next, let $\Lambda = \{\lambda_0, \lambda_1, \lambda_2, ..., \lambda_i\}$ be a set of labels. These labels are generated by labelers and placed on loci. They are then used by congregators to identify some distinguishing features of a congregation. The key is that agents place a semantic meaning on each label. This meaning may be as simple as a summary of the current members of a congregation, or as complex as a description of the features that would be common to the ideal congregation that a labeler wishes to attract. We will use λ_0 to denote the "null label", which is a label that carries no information.

Finally, we define a congregation c as a locus, a set of zero or more congregators, and a set of zero or more labelers. This is, $\forall c \in C, c = \{\alpha_i, ca, la\}$, $\alpha_i \in \alpha, ca' \in P(CA), la \in P(LA)$, where C is the set of congregations that can exist in a system.

3.1. Congregators

A congregator wishes to maximize its long-term utility. The underlying assumption here is that a congregator will have many chances to join congregations and therefore wishes to maximize its total satisfaction over all these encounters. Its predictions of future reward may be discounted so as to reflect either uncertainty of belief or the importance of immediate rewards. Our conjecture is that a congregator can benefit in the long term by devoting some initial resources to finding a good congregation. Thereafter, its decision as to who to interact with is simplified, since suitable partners are occupying the same congregation. For this work, we assume that a congregator can only join one congregation at a time.

Each congregator has a payoff function P which maps from a congregation c to a real number: $P : C \rightarrow \mathbb{R}$. Since a congregation's payoff to an agent is dependent upon the other agents that are a part of the congregation, a congregator will not be able to fully evaluate a congregation's utility before deciding whether to join it. Only after it (and all other agents) make this decision will it be able to evaluate P. While making their decision, congregators will rely on an estimate of their payoff (denoted $\hat{P}$) which maps from loci (and the labels placed on them) to the reals: $\hat{P} : \alpha \rightarrow \mathbb{R}$.

For this work, we will assume that $\hat{P}$ is reasonably accurate, so that we do not need to worry about congregators having to learn their preferences over labels. That is, when offered a choice between two labels, congregators already know which one they would prefer. This reduces the congregators' problem to that of selecting the best congregation to join from those that exist.

Let Λ_i be the set of labels which are offered to congregators (by labelers) in iteration i. A congregator c_j's decision function δ_{c_j} is then to choose a locus which maximizes $\hat{P_{c_j}}$.

3.2. Labelers

A labeler's problem is potentially more complicated. As with the congregators, a labeler wishes to maximize its long-term utility. It has one decision to make on every iteration: which label to offer. There are two confounding factors which make this problem difficult: a labeler does not necessarily know the congregators' preferences over congregations (or labels), and so must learn them; and the labeler's payoff for selecting a particular label may also be contingent on the labels offered by other labelers.

Formally, a labeler la has a payoff function P which is a function of all labelers' decision functions. (Recall that labelers each decide on a label simultaneously.) $P : \lambda_{la} \times \lambda_2 \times ... \times \lambda_n \rightarrow \mathbb{R}$. As with the congregators, labelers most likely do not have access to this function, and may find it easier to instead work with a function Con which predicts the congregation that will result from a given labeling decision. $Con_{\lambda_i} : \lambda_1 \times \lambda_2 \times ... \times \lambda_n \rightarrow P(CA)$ This function tells, for a labeler's label choice, along with all other choices, which agents will join its congregation.

This function may also need to be learned over time, meaning that a labeler would have functions $\hat{P}$ or $\hat{Con}$ or both. In order to do so, it may wish to model either the strategies and knowledge of other labelers or the preferences of congregators, or both. In this paper, we shall assume that labelers are 0-level learners in the RMM sense [5], meaning that they do not explicitly model other agents and their learning processes. Future work will examine the gains that labelers can make by attempting to infer something about the learning processes of both congregators and other labelers.

4. Applying CLRI to Congregating

In order to evaluate the difficulty of a particular congregating problem, we need a model that describes different aspects of the problem, how they relate to each other, and how varying an aspect changes the difficulty of the congregating problem. In this work, we have used CLRI as a model for describing congregating.

CLRI is a framework for analyzing multiagent learning that was developed by Vidal and Durfee [21, 20]. CLRI describes how simultaneously learning agents can affect the difficulty of each other's learning problem. Since the congregating problem is one in which each agent is learning which other agents it does and does not want to interact with, this framework is quite suitable. [1]

The CLRI framework consists of five parameters that are used to model the problem of agents learning a moving target function, which is a function that is a best response to other agents' actions, where their actions change over time. In every iteration, agents choose an action a from a set A of possible actions. Agents have a decision function $\delta^t(w)$ which returns an action for a given world state w at time t. They would like for this function to match $\Delta^t(w)$, the target function, which maps to the true optimal action at time t. In the above framework, the set of actions are congregations to join, and so this δ is identical to the δ described in the Congregators section above.

[1] As a note to readers who are concerned about a conflict between our characterization of congregation as multiagent learning and our earlier characterization of congregating as distributed search, understand that we are of the opinion that learning is a search through a space of possible hypotheses [11]

One point to note here is that the CLRI framework is based upon traditional PAC-learning assumptions [11]. In particular, it assumes that, for a given world state, there is one correct action and all the others are incorrect. There are also other assumptions which will be discussed as they become evident. We will maintain these assumptions throughout this paper. In the conclusion, we will discuss the limitations that CLRI's assumptions place on our model of congregations and present some ideas for extending the CLRI model.

The first two parameters of CLRI are change rate (c) and learning rate (l). Change rate is the probability that an agent will adjust an incorrect mapping of $\delta(w)$ from time t to time $t+1$, and learning rate is the probability that the agent will change the mapping of $\delta^{t+1}(w)$ to be equal to $\Delta^t(w)$. (That is, the agent will know what it should have done at time t.) Obviously, l must be less than or equal to c. In fact, l is equal to c times the probability that an adjustment will be the correct one for world state w. In the congregation framework, change rate is the probability that an agent will choose a different congregation than it had previously when faced with the same choices, and learning rate is the probability that an agent will be able to say after the fact where it should have gone (and thus make the right choice the next time it is faced with this set of choices).

The third CLRI parameter is retention rate (r), which is the probability that an agent retains a correct mapping ($\delta^t(w) = \Delta^t(w)$). In terms of congregating, this is the probability that an agent which chose the correct congregation will make this same choice again when faced with the same set of alternatives, even if it has changed its mapping of δ for other alternatives.

The fourth CLRI parameter is volatility (v), which is the probability that the function Δ which the agent is trying to learn changes. In other words, the probability that $\Delta^{t+1}(w) \neq \Delta^t(w)$. Note that nothing is said about *how much* Δ changes or what the correct action at time $t+1$ is, only that the correct action does change. In congregating terms, this is the probability that a locus α which was chosen at time t from a given set of loci is no longer the correct choice at time $t+1$ when given the same set of loci to choose from.

Since volatility can be difficult to determine through observation, CLRI provides a way to derive it using a fifth parameter: impact (I_{ij}). Impact is the effect that agent i's learning has on agent j's target function. In particular, it is the probability that agent j's Δ changes between t and $t+1$ as a result of agent i's δ changing. In terms of congregating, this is the probability that when congregator c_i decides to choose locus α' rather than α, congregator c_j's best choice of congregations changes.

In order to apply CLRI to congregating in any specific way, we need a domain in which these probabilities can be determined. In this paper, we examine congregating using CLRI within the context of the affinity group domain.

5. The Affinity Group Domain

We chose the affinity group domain as an intimal domain for studying the congregating problem. An affinity group is a set of agents that all share some characteristic, such as hair color or agent research. Agents of a particular affinity group want to join congregations that contain other members of their group and avoid congregations containing members of other affinity groups.

This is a very simple domain, but it is one that has several properties that make it useful for studying congregating. The first is the simplicity of agent preferences. They are straightforward to enumerate, which makes it easy to generate utility functions for each agent. Second, all agents agree on the states of the world that are optimal. Any state in which every agent from a particular affinity group is in the same congregation, and each affinity group is in a separate congregation, is equally preferred. This means that we, as evaluators of the system can identify these states ahead of time and recognize when the system of congregations is in one of these states. More importantly, it means that once such an optimal state is reached, all agents will remain where they are. Since each agent is in a state it believes optimal, it has no reason to move. In fact, this is the only time when all agents will remain still. This means that we can detect convergence by noticing when no agents wish to switch congregations. It also means that we will not have scenarios where agent A wishes to congregate with agent B, but agent B does not wish to congregate with agent A. In that instance, the two agents would chase each other indefinitely.

6. Congregating Without Labels

As a first attack at understanding the congregating problem, we examine the case in which there are no labelers. This means that, for all loci, the associated label is λ_0, the null label. Congregators therefore can only guess randomly as to which congregation is best for them.

We begin with the simplest case, in which there is only one affinity group and a number of loci. Once we have established some analytic results for this problem, we extend this to a model with multiple affinity groups and examine how this affects the problem complexity.

6.1. One Affinity Group

If there is only one affinity group, then all congregators simply want to congregate in the same location. The problem, of course, is that they don't know ahead of time which locus to choose. This is a very simple problem, but it lets us establish some notation and lead into scenarios with multiple affinity groups.

Assume that there are ϕ agents and m loci. Each agent a receives the following payoff:

$$Payoff(a) = \begin{cases} 1 & \text{if } |C(a)| = \phi \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $|C(a)|$ indicates the size of the congregation agent a is in. Congregators will search randomly until they find themselves in a congregation with every other agent.

The first question we can ask is how long it will take for a set of ϕ congregators to find each other. The probability that all agents will choose the same locus on any given time step is:

$\epsilon = \left(\frac{1}{m}\right)^{\phi-1}$.

Therefore, the agents will succeed in finding each other with probability ϵ on the first iteration, $\epsilon(1-\epsilon)$ on the second, and so on. The probability that all congregators have found each other within t iterations is:

$$\epsilon \sum_{i=0}^{t-1} (i+1)(1-\epsilon)^i = 1 - (1-\epsilon)^t. \tag{2}$$

We can then solve for t to ask how long it takes for a set of congregators to find each other with probability p.

$$t \approx m^{\phi-1} |ln(1-p)|. \tag{3}$$

Since this is an affinity group, once all the agents have converged to the same congregation, they will remain there, since they are all receiving a positive payoff. Also, if all agents have not converged to the same congregation, all of them will be unsatisfied, since they will be receiving a payoff of 0. Therefore, there are m stable states in the global search process (one for each locus). The salient point in this example is that the time needed for agents to find each other increases exponentially with the number of agents in the system. We will use the time to converge as an indicator of the problem's hardness or complexity throughout the remainder of this paper.

We can use the above formulae to determine the corresponding CLRI parameters. We begin with c, the change rate, which is the probability that an agent will change an incorrect mapping. This will happen any time a congregator is not grouped with every other agent of its affinity group, so $c = 1 - \epsilon$. Similarly, l, the learning rate, is the probability that an incorrect mapping will be changed to a correct one. This is simply c times the probability of picking the right locus, which is $(\frac{1}{m})$. r, the retention rate, is 1. When congregators find the right location, they quit moving and so there is no chance of "forgetting." Finally, we have volatility (v), which is the probability that the target function will change. This is the probability that the congregators are not all in the same location multiplied by the probability that the "right" locus changes on the next iteration: $v = (1 - \epsilon)\frac{m-1}{m}$.

The above parameters indicate that as more agents are added, the problem becomes exponentially more complicated. In particular, v contains the terms $(1-\epsilon)$ and $\frac{m-1}{m}$. Since ϵ becomes exponentially small as the number of agents increases, volatility approaches $\frac{m-1}{m}$ in the limit. If m is at all large, this will be close to 1, meaning that every agent's actions have a very strong effect on the target functions of other agents. This is what we would intuitively expect, since the problem depends on all agents making the same decision.

6.2. Multiple Affinity Groups

We now generalize the problem to a case where there are g affinity groups of congregators, where each group is of size s, for a total of gs congregators. Again, there are a total of m loci (in the experiments, we will make the simplifying assumption that $m = g$). This means that there are m^{sg} different world states. Once again, each congregator wants to join a congregation which maximizes its payoff. In this experiment, we modify the payoff function slightly.

$$Payoff(a) = \begin{cases} \Sigma_{c \in C(a)} \frac{Distance(c)}{|C(a)|} - tax & \text{if } sum > tax \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where $C(a)$ is the congregation an agent is in (and $|C(a)|$ is the number of agents in this congregation) and tax is an "existence tax" that agents must pay every iteration (to pay for computational resources, etc.). $Distance$ is a function that determines how different (or how far apart) two affinity groups are. To ease analysis and remain consistent with the assumption discussed above that there be only one correct action in a given world state, we will let $Distance$ equal 1 if agents are in the same affinity group and 0 otherwise. In addition, to make non-optimal actions undesirable, we set *tax* equal to $\frac{s}{s+1}$. In short, this function assures that agents will only receive a positive payoff and therefore stay still in the case where they are in a congregation consisting solely of their own affinity group.

We are now able to determine the CLRI parameters. This problem is quite complicated, since there are such a large number of combinations of congregators. The first parameter is c, the change rate. A congregator will change an incorrect mapping any time it is not in the correct congregation, which, if all agents move randomly, will happen with probability $c = 1 - ((\frac{m-1}{m})^{(g-1)s})$. This is the probability that a congregator is not in a congregation consisting solely of members of its affinity group.

The second parameter is the learning rate (l). This is simply the change rate c times the probability of randomly moving to the best available congregation, which is $(\frac{1}{m})$.

The third parameter is the retention rate (r). A congregator will retain a correct mapping if it is in the correct congregation and no congregators of another type arrive. In other words, $r = (\frac{m-1}{m})^{(g-1)s}$.

The fourth parameter is impact (i), which is the effect that agent b's decision has on agent a's target function. Agent b can affect agent a in two ways: if it is different and moves into agent a's congregation, or if it is the same and it moves out. The first case will happen with probability $\frac{1}{m}$. In the second case, a like agent will move out if there is a different agent in the congregation, which happens with probability $1 - \frac{m-1}{m}^{s(g-1)}$.

To make things concrete, assume that there are 3 affinity groups, each containing 3 agents. There are also 3 loci. This is the smallest example that demonstrates the complexity of the congregating problem. $c \approx 0.998$, $l \approx 0.332$, $r \approx 0.087$, the impact that like agents have on each other is $\frac{1}{3}$, and the impact that different agents have on each other is approximately 0.913.

Determining closed-form solutions for the time to convergence and the probability of convergence is more difficult in this scenario, since it is possible for a subset of congregators to settle early on and wait for others to find them. In other words, the probability of convergence is dependent on the entire past history. If we examine the learning rate and impact of agents from different affinity groups, we can see that the problem's complexity lies in the fact that both terms contain exponents for the number of agents: $(s(g-1))$. As in the previous problem, as the number of agents increases, the number of possible solutions explodes and l declines. In addition, each agent is moving around through loci trying to find the right combination , so v approaches 1 as every agent impacts the learning of other agents.

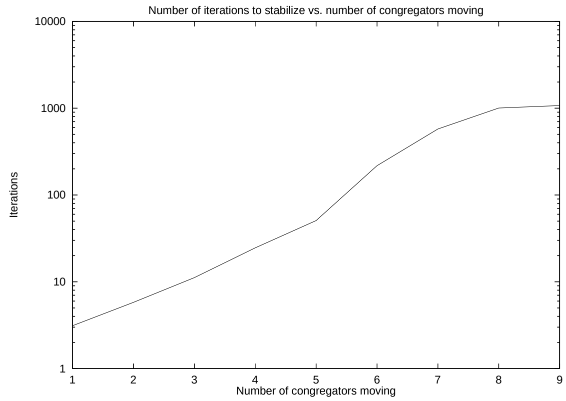

Figure 1. Iterations to stabilize as a function of the number of congregators moving.

6.3. Experimentation

Since it is difficult to analytically determine the time needed for a system with multiple affinity groups to converge, we performed some simple experiments to develop an estimate. Using the same scenario as described above (3 loci, 3 affinity groups of three congregators each), we begin by correctly fixing 8 congregators and having 1 move. We then fix 7 and allow 2 to move, and so on. Agents move randomly between congregations until they settle on a locus where there are only members of their own affinity group (i.e. their payoff was 1). They then stay there unless “pushed out” by the arrival of a congregator from a different affinity group. Results (averaged over 50 trials) are shown in Figure 1.

As predicted in the analysis, the time needed for a system of congregations to stabilize increases exponentially as the number of congregators increases. Even with this relatively small system, with a very simple payoff function, it quickly becomes very difficult for congregators to find each other.

7. Introducing a Flat Label Space

As we noted earlier, if agents are able to label the loci they are at and attract like members in that way, the congregating problem can potentially become much easier.

In this situation, the congregator’s decision function becomes trivial; simply move to the congregation with the label that most closely corresponds to its affinity group. If no such label exists, move randomly.

In this case, it is the labeler's problem that is the interesting one. Each labeler must choose a label which will attract an affinity group. The value of its decision will be influenced by the choices of the other labelers. Therefore, each labeler wishes to attract a distinct affinity group. We begin with a flat label space. That is, each label potentially refers to at most one affinity group. In section 8, we extend these results to consider the case of hierarchical labels.

7.1. Analysis

In order to fit this problem within the CLRI framework, we make some simplifying assumptions. First, each labeler is neutral as to which affinity group it wants to attract. A labeler is happy whenever all congregators in its group are of the same type. Second, labelers have no memory regarding the payoffs from labels other than the current one. A labeler simply makes the decision as to whether to keep or discard the current label. We will also assume that each congregator agrees on the meaning of each label. If any congregator misinterprets the meaning of a label, they all will (in the same way).

We retain the notation used in the previous section. In addition, since there are m loci, there will be m labelers. Since g (the number of groups) is equal to m, there are m labels.

Within this framework, c (the change rate) is 1. If a labeler has an incorrect label, it will always change it. l, the learning rate, is the probability that the labeler will select an unused label, which is $(\frac{g-1}{g})^{m-1}$. r is also 1, by definition; if a labeler has a correct label, it will keep it. i is the impact that labeler i has on labeler j, which is the probability that i selects j's label, given that i was wrong and j was right. This is $((\frac{g-1}{g})^{m-1})\frac{1}{m}$.

Using the same parameters as above (3 labelers and loci, 3 affinity groups of 3 agents each), $l = \frac{4}{9}$ and the impact of i on j is $\frac{4}{27}$.

Once again, determining the time needed to converge is non-trivial, since one labeler's decision may aid another. In fact, what typically happens is that one labeler will attract an affinity group, who will then hold still and make the problem easier for other labelers. Therefore, to determine the time needed to converge with probability p, one must determine this probability for each possible history and then aggregate these probabilities appropriately.

The point to notice is that both learning rate and impact here are dominated by $(m-1)$ terms, meaning that the difficulty of the problem grows with the number of labelers and labels (or loci), rather than the number of congregations. In systems with a small number of labelers

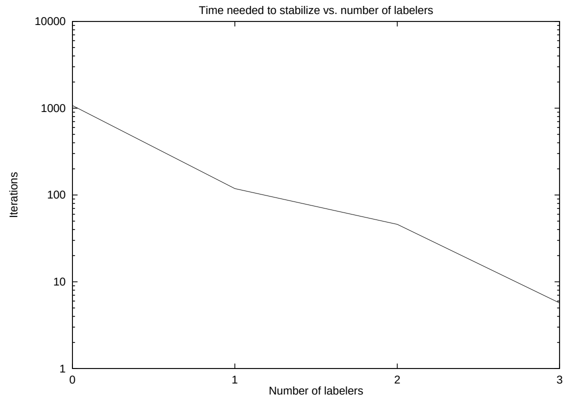

Figure 2. Iterations to stabilize as a function of the number of labelers.

relative to congregators, labeling would seem to provide a substantial advantage.

7.2. Experimentation

Once again, since it was difficult to determine convergence time analytically, we decided to develop an experimental estimate. In this experiment, we began introducing labelers for each locus. A labeler could choose which affinity group it wanted to attract; its decision was independent of the other labelers. Congregators would automatically go to a group with the corresponding label (or choose randomly between offerings if more than one labeler offered the same label) or move randomly if the label for their group was not offered. Each congregator received a payoff according to the function described in section 6.2. The labeler's decision was analogous to the congregators' in the previous experiment: if the label chosen was successful (that is, it attracted a complete affinity group with no extra agents) it would continue to use that label. Otherwise, it would select a new label at random.

We began with a baseline experiment with no labelers. This is the same as allowing all congregators to move in the first experiment. We then added one labeler, then two, and then three. Results are shown in Figure 2.

There are a few things to notice here. The first is that the problem becomes trivial with three labelers. They settle on a solution almost immediately. Even with only two labelers, the problem becomes very easy; the congregators which don't have a label to attract them are able to settle on a common locus relatively quickly.

The slightly more subtle point is that adding one labeler produces the same effect as holding two congregators steady, and adding two labelers produces the same effect as holding five congregators steady. The label produces a shared decision amongst all members of the corresponding affinity group; they will all move to the same congregation, and so in terms of convergence and impact they can be considered as one agent. This is a big savings; as was noted previously, a major difficulty with the congregating problem is the impact that one agent's decision has on another. If all agents in an affinity group make the same decision, then they will produce the same amount of impact on an agent of another affinity group as a single agent in a system without labelers. As more affinity groups are introduced, the potential benefits from introducing labelers will be even greater, since a system with labels will increase impact linearly (with each group added) rather than exponentially (with each group added *times* the number of agents in each group) in systems without labelers.

It is also worth noting that a large benefit of labeling in this example comes from the fact that the set of labels is the same size as the number of affinity groups. If the set of labels is much larger, then using labels to attract a congregation becomes more difficult, since labelers will often choose labels that do not correspond to any affinity group. One way around this is the introduction of hierarchical labels, where some are more general than others.

8. Adding a Hierarchy of Labels

One common form of hierarchical labeling, which makes sense within the affinity group domain, is the introduction of abstract labels that can potentially attract more than one affinity group. This is in contrast to a base label, which corresponds to at most one affinity group. When there are many more labels than affinity groups, using abstract labels can make sense; they allow labelers to have a better chance at attracting congregators. In this paper, we restrict ourselves to abstract labels that are the logical OR of simpler labels. The corresponding abstract and base labels form a lattice, as seen in figure 8.

A labeler that can select an abstract label has a tradeoff to make: a more abstract label is more likely to attract congregators, but if several affinity groups are attracted, their members will not be happy with the congregation. Also, we assume that a congregator will choose a more specific label over a more general one, since the more specific label will be more likely to contain a higher percentage of congregators in its affinity group, making too-general labels unuseful. The larger the

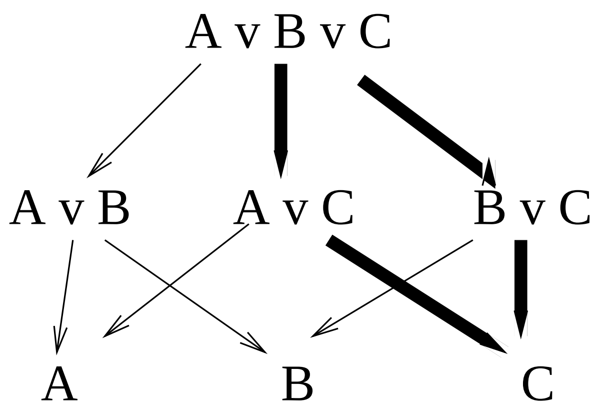

Figure 3. All possible abstract and base labels of a three-label system.

number of labels is relative to the number of affinity groups, the higher in the hierarchy a labeler should select labels. We will now outline this more formally in CLRI terms.

8.1. Analysis

We begin with some terminology that will aid in applying CLRI. We refer to the *level* of a label in a hierarchy as the label's height in the hierarchy. In figure 8, the label a is at level 1 and the label $a \vee b$ is at level 2. More general labels have a higher level.

As was mentioned above, one danger of selecting an abstract label is the selection of a similar, but more specific label by another labeler. The labeler that offers the more specific label is *covering* the other labeler. For example, assume there is one affinity group, labeled a. If labeler 1 chooses the label $a \vee b \vee c$ and labeler 2 chooses the label $a \vee b$, labeler 2 covers labeler 1, since all the congregators in group a will choose labeler 2, as its label meets their requirements more specifically.

We will also refer to a label *matching* an affinity group. A label matches an affinity group if congregators in that group would pick this label, given that no other label covers it. For example, $a \vee c \vee d$ and $a \vee b$ both match group a.

The primary calculation that needs to be made in order to calculate the various CLRI parameters is the probability of a labeler choosing a successful label. A label is successful if it matches at least one affinity group, and it is not covered by any other label. We begin by determining the total number of possible labels. Let g be the number of affinity groups, m be the number of labelers, and l be the number of base labels, where $g < l$. The total number of labels is:

$$\sum_{i=1}^{l} lCi = 2^{l-1}$$

If a labeler chooses a label at level l', it will have a probability of $g(1 - \frac{(l-l')}{l} = g(\frac{l'}{l}$ of matching at least one affinity group. As the label becomes more abstract, the chances of matching an affinity group increase. However, the chances of being covered by another labeler also increase. For example, in figure 8, if labeler 1 chooses label $a \vee b \vee c$ and the affinity group is truly of type c, any labeler who chooses a label in the sub-lattice from c to $a \vee b \vee c$ (indicated with the dark lines) would cover labeler 1.

Given that a labeler has chosen a successful label at level l', the probability that another labeler will cover it is:

$$m(1 - \frac{\sum_{i=1}^{l} lCi - \sum i = 1l' - 1(l'-1)Ci}{\sum_{i=1}^{l} lCi} = \tag{5}$$

$$m(1 - \frac{2^{l-1} - 2^{l'-2}}{2^{l} - 1}) = \tag{6}$$

$$m(1 - (1 - \frac{2^{l'-2}}{2^{l-1}} = \tag{7}$$

$$m(2^{l'-l+1}) \tag{8}$$

As a labeler chooses a label with a higher label, its chances of being covered increase exponentially. We can now determine the probability of success as the probability of matching and not being covered. This is:

$$Pr(success \neg covered) = m^2(2^{l'-l+1})(\frac{l'}{l}).$$

Note that the probability of matching grows with the ratio of $'l$ and l, whereas the probability of being covered grows with their difference.

Once we know this probability, the CLRI parameters are straightforward to determine. As above, change rate and retention rate are 1. Learning rate is simply the probability of success, which was determined above. Impact is the probability that a particular labeler will cover a successful label is $2^{l'-l+1}$ (from equation **??**), and volatility is the probability that any other labeler will cover a successful label, or m (the number of groups) times this, or $m2^{l'-l+1}$.

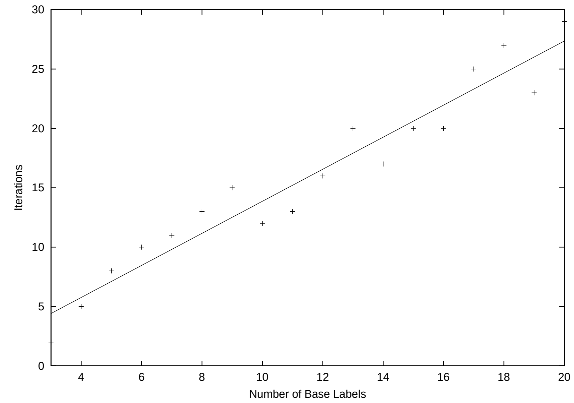

Figure 4. Iterations to stabilize as a function of the number of base labels.

9. Experimentation

Once again, convergence time was determined experimentally. The addition of hierarchical labels also adds an additional decision for the labeler: whether to offer more or less abstract labels. We chose a very simple strategy. A labeler that receives no payoff will attempt to abstract, assuming that it has not matched with anyone. A labeler that receives a partial payoff will choose a less abstract label, assuming that its generality has attracted too many affinity groups. Finally, a labeler that receives full payoff will stay put.

We began with a scenario in which there were three labelers, three affinity groups of three congregators each, and three base labels. We then held the number of labelers and congregators fixed and introduced extra base labels, up to a maximum of 20. The median number of iterations needed for the labelers to each attract an affinity group (over 100 trials) is shown in figure **??**.

We used the median here rather than the mean since the experiments had a large standard deviation. Notice that the number of iterations needed for the system to stabilize increases linearly as more labels are added. Examining the individual runs indicates that, in a typical run, each labeler tended to use about $\frac{1}{3}$ of the available labels. Once each labeler added enough base labels to have this many in their abstract label (each started with one), they were then reduced to a problem very similar to the flat label space presented in section 7; as we saw previously, this problem also grew linearly.

In general, it appears that allowing labelers to offer hierarchical labels and, in particular, to exploit the structure of such a hierarchy when searching for appropriate labels, makes finding optimal sets of congregations much easier. This result offers hope that congregating can indeed be applied to large-scale multiagent systems of the kind that we expect will soon appear within the Internet.

10. Conclusions

This paper has presented congregating as both a a metaphor for understanding how groups of agents can discover each other and as a formal model for describing the difficulty of particular multiagent coordinating problems. We have used the affinity group domain to apply CLRI to the congregating process. Doing so, we have shown that when agents have no means of describing themselves to each other, the problem complexity grows exponentially in the number of agents. By adding specialized agents known as labelers, the problem complexity reduces to linear in the number of labelers. If the number of labels is small, this is sufficient. However if the number of labels is large relative to the number of affinity groups, extra structure is needed to avoid exponential search. By allowing labelers to offer abstract labels arranged in a hierarchical structure, the congregating problem remains linear in the number of labels, which offers hope for the scalability of large multiagent systems.

This work is merely the beginning of a long series of question regarding congregating. The affinity group is a useful domain for studying congregating, but it is considerably more simple than real-world problems. We would like to apply congregating principles to more realistic problems, such as information economies. The will necessitate the extension of CLRI to utility-based domains. We would also like to consider more satisficing sorts of congregations. In this work, agents typically had one congregation they were happy with; in other domains, agent might want to trade off the quality of a congregation against the expected cost of finding a better congregation. We believe that the congregating framework is robust enough to support these extensions.

References

1. Brooks, C. H. and E. H. Durfee: 1999, 'Congregation Formation in Information Economies'. In: *Proceedings of the AAAI-99 Workshop on AI in Electronic Commerce*.
2. Brooks, C. H., E. H. Durfee, and A. Armstrong: 2000, 'An Introduction to Congregating in Multiagent Systems'. In: *Proceedings of the Fourth International Conference on Multiagent Systems*. pp. 79–86.
3. Cohen, P. R. and H. J. Levesque: 1991, 'Teamwork'. *Nous* **25**, 487–512.
4. Durfee, E. H., D. L. Kiskis, and W. P. Birmingham: 1997, 'The Agent Architecture of the University of Michigan Digital Library'. *IEEE/British Computer*

Society Proceedings on Software Engineering **144(1)**, 61–71. Special Issue on Intelligent Agents.

5. Gmytrasiewicz, P. J. and E. H. Durfee: 1993, 'Toward a Theory of Honesty and Trust Among Communicating Autonomous Agents'. *Group Decision and Negotiation (Special Issue on Distributed Artificial Intelligence)* **2**, 237–258.
6. Grosz, B. J. and S. Kraus: 1996, 'Collaborative plans for complex group action'. *Artificial Intelligence* **86**, 269–357.
7. Hu, J. and M. Wellman: 1998, 'Multiagent Reinforcement Learning: Theoretical Framework and an Algorithm'. In: *Proceedings of the Fifteenth International Conference on Machine Learning (ICML-98)*.
8. Jennings, N. R.: 1995, 'Controlling cooperative problem solving in industrial multi-agent systems using joint intentions'. *Artificial Intelligence* **75**, 195–240.
9. Ketchpel, S.: 1994, 'Forming Coalitions in the Face of Uncertain Rewards'. In: *Proceedings of the National Conference on Artificial Intelligence*. Seattle, WA, pp. 414–419.
10. Mas-Colell, A., M. D. Whinston, and J. R. Green: 1995, *Microeconomic Theory*. New York: Oxford University Press.
11. Mitchell, T. M.: 1997, *Machine Learning*. Boston, Massachusetts: WCB/McGraw-Hill.
12. Mullen, T. and M. P. Wellman: 1998, 'The Auction Manager: Market middleware for large-scale electronic commerce'. In: *Proceedings of Third USENIX Workshop on Electronic Commerce*.
13. Rosenschein, J. S. and G. Zlotkin: 1994, *Rules of Encounter: designing conventions for automated negotiation among compters*. Cambridge, MA: MIT Press.
14. Sandholm, T. W. and V. R. Lesser: 1997, 'Coalitions among Computationally Bounded Agents'. *Artificial Intelligence* **94**(1), 99–137.
15. Sen, S., M. Sekaran, and J. Hale: 1994, 'Learning to Coordinate without Sharing Information'. In: *Proceedings of the National Conference on Artificial Intelligence*. pp. 426–431.
16. Shehory, O. and S. Kraus: 1998, 'Methods for Task Allocation via Agent Coalition Formation'. *Artificial Intelligence* **101**, 165–200.
17. Singh, M. P., A. S. Rao, and M. P. Georgeff: 1999, 'Formal Methods in DAI: Logic-based Representation and Reasoning'. In: G. Weiß (ed.): *Multiagent Systems: A Modern Aapproach to Distributed Artificial Intelligence*. Cambridge, MA: MIT Press, pp. 331–376.
18. Sycara, K., K. Decker, A. Pannu, M. Williamson, and D. Zeng: 1996, 'Distributed Intelligent Agents'. *IEEE Expert* **11**(6), 36–46.
19. Tambe, M.: 1997, 'Toward Flexible Teamwork'. *Journal of Artificial Intelligence Research* **7**, 83–124.
20. Vidal, J. M.: 1998, 'Computational Agents that Learn About Other Agents: Algorithms for Their Design and a Predictive Theory of Their Behavior'. Ph.D. thesis, University of Michigan.
21. Vidal, J. M. and E. H. Durfee: 1998, 'The Moving Target Function Problem in Multi-Agent Learning'. In: *Proceedings of the Third International Conference on Multi-Agent Systems*. Paris, France.
22. Zlotkin, G. and J. S. Rosenschein: 1994, 'Coalition, Cryptography, and Stability: Mechanisms for Coalition Formation in Task Oriented Domains'. In: *Proceedings of the National Conference on Artificial Intelligence*. Seattle, WA, pp. 432–437.