

A Unified Framework for Multi-Agent Agreement *

Kiran Lakkaraju and Les Gasser

Technical report number: UIUCDCS-R-2006-2803

December 15, 2006

Abstract

Multi-Agent Agreement problems (MAP) - the ability of a population of agents to search out and converge upon a common state - are central issues in many multi-agent settings, from distributed sensor networks, to meeting scheduling, to development of norms, conventions, and language. While much work has been done on particular agreement problems, no unifying framework exists for comparing MAPs that vary in, e.g., strategy space complexity, inter-agent accessibility, and solution type, and understanding their relative complexities. We present such a unification, the Distributed Optimal Agreement Framework, and show how it captures a wide variety of agreement problems. To demonstrate DOA and its power, we apply it to two well-known MAPs: convention evolution and language convergence. We demonstrate the insights DOA provides toward improving known approaches to these problems. Using a careful comparative analysis of a range of MAPs and solution approaches via the DOA framework, we identify a single critical differentiating factor: how accurately an agent can discern other agent's states. To demonstrate how variance in this factor influences solution tractability and complexity we show its effect on the convergence time and quality of Particle Swarm Optimization approach to a generalized MAP.

1 Introduction

In a *Multi-Agent Agreement Problem* (MAP) multiple agents must navigate a space of possible states (a *potential agreement space*) and eventually converge on the same state. By illustration, a simple MAP from distributed transaction processing is the *distributed commit* problem: all agents participating in a single transaction must eventually converge on one state in a two-valued potential agreement space, namely whether to *commit* or *abort* the transaction. The well-known “two-phase commit” and “three-phase commit” protocols are different specific *solutions* to this MAP, with differing degrees of robustness, complexity, and centralization (cf. [9]).

*A version of this paper was submitted to the 2007 International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007)

Different types of MAPs occur in many other multi-agent domains as well. In distributed active sensor networks (such as an array of active temperature sensors embedded in a road), the agreement state often of interest is some function f of the sensor values across the whole sensor collection (commonly, f is *max*, *min*, or *average*), which should become known to all active sensors. The agents must locate and agree upon this state using a distributed algorithm. Here the potential-agreement space is the set of potential values of f , and the final needed agreement is the actual value of f . [13]. Similar problems arise in stabilizing the physical formation of a set of agents (agreement on roles and positions), flocking/swarming (agreement on overall direction), and synchronization of coupled oscillators (agreement on state of the oscillation). Negotiation can be viewed as a MAP, in which agents aim to converge on a common state in an *offer space* (or fail). Meeting scheduling can be viewed as a MAP governed by a set of equality constraints over time and availability [11]. Multi-agent agreement is at the heart of the notions of “norms” and “conventions” for multi-agent systems [15, 18]. Here the potential agreement space is the collection of agents’ possible action strategies in a particular class of situations, and the resulting convention is a strategy agreed to and followed by all agents.

Our particular area of interest is the autonomous creation of language by an agent collection [2, 7, 16, 17]¹. Language necessarily a collective phenomenon. One quality measure over a population of language users is the population’s average *communicability*². Communicability implies agreement on language. A set of agents can achieve high communicability by settling on any one of a wide variety of possible languages. But individual languages might also differ on “objective” qualities, such as *expressivity* (ability to express important concepts), *efficiency* (of production and interpretation), and so on. Thus, for the language MAP, a solution has both a *frequency-dependent* component (communicability, which depends on the *frequency* of agents sharing a language) and an *intrinsic* or objective quality component. Agents need *agreement on a common and high-quality state (language)*.

This type of solution condition that combines frequency-dependent and intrinsic components is characteristic of many MAPs, but not all. Clearly, agents pursuing *distributed commit* must collectively make the right choice, not just any choice. In contrast, however, for some social conventions agreement itself is more important than what is agreed upon—it’s frequency-dependent only—so agents can settle on any state in the possible agreement space.

At first glance it would seem that since these various problems can all be cast as agreement problems, both the MAPs and their solutions should follow a general and by now well understood pattern. Indeed, our first assumption was that well-known distributed algorithms such as those collected in [9] provide the solution concept for most MAPs, so we could reuse these and little more

¹Language is also important because it is a model problem for dynamic semantics in general in distributed information systems: e.g., evolving shared schemas/ontologies, shared denotations, common information structures, etc.

²Defined as the average probability that an “utterance” by one agent will be understood by another; see [12].

remained to be done. Trying this, however, we found two issues:

First, each MAP domain has its own idiosyncrasies of problem description that are sometimes hard to apply to other domains. Some kind of MAP *lingua franca* is needed to validate any unity among MAPs. Thus the first contribution of this paper is to present a framework, which we call *Distributed Optimal Agreement (DOA)*, that can unify the different agreement problems from different problem domains under one common descriptive model, allowing researchers to compare, contrast, and understand MAP differences clearly. We detail the DOA framework in Section 2 and show examples of MAPs captured with Distributed Optimal Agreement in Section 3.

Second, and more importantly, by using the DOA framework to compare MAPs across domains we discovered how MAP problems differ on dimensions including *complexity of the potential-agreement space*, *state-to-state accessibility*, *agent interaction topologies*, *state evaluation measures*, and *type of solution needed* (to name a few). Further, we analyzed how these dimensions cause complexity to emerge in MAP solutions, what factors make a MAP difficult to solve, and how differences impact potential MAP solution strategies.

From our comparison of different MAPs via the DOA framework we identified the discernibility level - that is how well an agent can discern the states of other agents in the population - to be a critical factor in determining how difficult it is to solve a MAP. Our final contribution is to elaborate on this role of discernibility and demonstrate how variance in this factor influences convergence times for Particle Swarm Optimization approaches to MAPs.

2 DOA Framework

The Distributed Optimal Agreement framework comprises a specification of a problem and a specification of the dynamics of the agents in solving that problem. Informally, we view the process of solving a MAP as search. Each agent moves about in a *possible agreement space* that comprises a number of *possible agreement states (PAS)*. Any of the PASes might be the substance of an agreement, depending on its own qualities and the number of agents that have settled on it. A *complete agreement* is the condition that all searching agents have arrived at the same PAS. If there is a distance metric on the space of states, a MAP may enjoy the concept of *complete ϵ -agreement*, i.e. all agents being within ϵ distance of each other. For example, an *accessibility relation* over the possible activity states allows us to define distance as path length between states, and ϵ as the largest diameter of the accessibility graph for states agents are in. Analogously, a *k-agreement* is the condition that at least k agents have settled on a single state.

Defining a MAP in the DOA framework involves defining the characteristics of the possible agreement space, accessibility relation, solution criteria, and so on. We present the more formal DOA model below.

2.1 Formal Problem Model

An agreement problem in the DOA framework is defined by the 7-tuple:

$$\{A, \Sigma, \Delta, \Theta, \rho, S, \Omega\}$$

where:

1. **Agents:** A is a set of n agents, $\alpha \in A$. Agents are the active processes in the DOA model, whose actions take place in an interval in a time line T . At each time $t \in T$, an agent is said to be “in” some Possible Agreement State (see below).
2. **Possible Agreement Space:** The substance of an agreement in the DOA model is the *possible agreement state (PAS)*, denoted by σ : a state of the world on which agents could agree. For instance, a PAS could be a language an agent chooses to speak, an offer in a negotiation, a candidate strategy for a convention, or a decision to commit/abort a transaction. Some previous work in this area uses the term “strategy” where we use “Possible Agreement State”, but we prefer PAS because we aim to capture many more kinds of agreement than just shared strategy choices. Σ is the set of all PASes, thus $\sigma \in \Sigma$. We use $\sigma_{\alpha_i, t}$ to denote the PAS that agent α_i is “in” at time t .

Configurations

Let Σ^n be the set of all possible associations of PASs with all the agents in A . Σ^n is thus an n -dimensional space. At time t the *configuration* of the entire system is $s_t \in \Sigma^n$ —that is, one specific association of all agents with states.

3. **Accessibility Relation:** $\Delta : A \times \Sigma \times \Sigma \rightarrow \{\mathbb{R} \cup \infty\}$ is the *accessibility relation* for PASes σ . $\Delta(\alpha_i, \sigma_j, \sigma_k)$ describes the (possibly infinite) cost for some agent α_i to move from σ_i to σ_j . Δ models the structure of the possible agreement space Σ from the perspective of each agent. An agent with more limited capabilities might have a higher cost for changing from one PAS to another, or one PAS might be inherently more difficult (or impossible) to reach directly. For example, representing languages as binary strings and assuming only single point mutations as transition operators[10] results in a hypercube-structured Δ for the language space.
4. **Interaction Relation:** $\Theta : A \times A \times T \rightarrow \{\mathbb{R} \cup \infty\}$ is the *interaction relation*. $\Theta(\alpha_i, \alpha_j, t_i)$ describes the cost for an agent α_i to interact with (e.g. sense, observe, communicate with) some other agent α_j at time $t_i \in T$. Cost is a very general basis for an interaction relation. For example, a close interaction neighborhood for some agent can be defined as the set of other agents with which communication is cheap relative to other agents. If cost is inversely related to probability of interaction over time, then Θ describes agent-to-agent interaction frequencies, and can be

used to model a type of *frequency-weighted social network*. In many MAPs the interaction relation is already specified as a graph, where the nodes are agents and the weighted edges reflect the probability with which the agents interact. This is easily represented in the DOA framework. The interaction relation is conditioned on time to capture changing topologies of interaction (cf. [13]).

5. **Intrinsic Value:** $\rho : A \times \Sigma \rightarrow \mathfrak{R}$ defines the *intrinsic value* of an agent being in a particular PAS. $\rho(\alpha_i, \sigma_j)$ defines the reward agent α_i receives from being in PAS σ_j . Σ can be seen as a landscape with hills and valleys corresponding to ρ . Since ρ is defined based only on the agent and what strategy it is using, and not on what strategies other agents have, we consider ρ to be the *intrinsic* value of the state with respect to an agent. In many cases ρ is independent of the agent as well. We define $\max(\rho)$ as the set $\{(\alpha_i, \sigma_j)\}$ with the highest $\rho(\alpha_i, \sigma_j)$
6. **Starting Configurations** The set of possible starting configurations, $S \subseteq \Sigma^n$. $s_0 \in S$ is the initial state of the population.
7. **Termination Configurations** The set of possible termination configurations, $\Omega \subseteq \Sigma^n$. There are many types of termination configurations. Here are several interesting ones:

Simple Consensus Configurations in Ω are *agreements*. A *complete agreement* is formed by a set of agents all being “in” the same PAS, for example all choosing to subscribe to a *particular* language, negotiation offer, convention strategy, etc. This is denoted as a configuration with the following property:

$$s \ni \forall i, j, \sigma_{\alpha_i, t} = \sigma_{\alpha_j, t} \quad (1)$$

Other consensus-oriented configurations types include those for the ϵ - and k - agreements as described informally above.

Consensus+Optimization At some t $\sigma_{\alpha_i, t} = \sigma_{\alpha_j, t}, \forall i, j$ and $(\alpha_0, \sigma_{\alpha_0, t}) \in \max(\rho)$. This is the set of configurations in which every agent is using the same strategy, and that strategy has the highest intrinsic value.

Consensus+Computation Given a function $\chi : \Sigma^n \rightarrow \Sigma$, at some t , $\sigma_{\alpha_i, t} = \chi(s_0), \forall i$. The set of configurations where every agent is in the same PAS, and that specific PAS is a function of the initial PASes of the entire population.

2.2 System Dynamics

Solving an instance of a DOA problem involves specifying the behavior of the agents such that the system moves from a configuration $s_0 \in S$ to a configuration $s_\omega \in \Omega$ in some finite amount of time. We assume a turn-based system, where

at each time step t the three-step process of active agent selection, information gathering, and information use occurs, as follows:

1. Active Agent Selection: A subset $C_t \subset A$ (called the *active agent set*) becomes active at this time step.

2. Information Gathering: Information (call it ψ) is necessary for efficient search. Complete information about a system configuration is costly, being influenced by Θ , n , and $|\Sigma|$ and the history of activity represented by T . Thus strategic selection of information sources at each time step is necessary in a MAP. Once this choice is made, the actual interactions occur.

2a. Interaction Choice Each active agent $\alpha_i \in C_t$ chooses some other subset of agents $I_{i,t} \subset A$ (called the *interaction set* of α_i), from which to gather information about the current configuration. This substep is purely the choice a set of other agents to observe or communicate with.

2b. Interaction α_i interacts with the agent(s) in I from the previous step. This interaction produces some *information* for α_i about the current configuration.

3. Information Use Finally, all agents active in this time step (C_t) apply a decision rule $\sigma_{\alpha_i,t+1} = f(\sigma_{\alpha_i,t}, \psi)$ possibly moving to another PAS.

2.2.1 Agent Activation

The agent activation stage which agents are active at each time step. This choice is implemented at a system wide level. In a discrete-time model this amounts a rule to decide which agent or agents are allowed to gather information and change their state. There are generally three options for this choice: Random, State-based, or Complete.

In Random activation an agent or set of agents is chosen at random from the population, and proceed to gather information and change their state. Work in the evolution of conventions area has used this type of dynamics ([15]).

In Complete agent activation all agents are chosen to be active at each time step. An example of this kind of activation occurs in Particle Swarm Optimization (PSO) systems ([6]).

Finally, in State-based activation, an agent or set of agents are chosen for activation based on some attribute of the state they are in. For instance, the probability of choosing an agent might be proportional to the intrinsic value of the agent's state. The system described in Lieberman et. al., ([8]) exhibited this property - agents were chosen according to their current fitness.

2.2.2 Information Gathering

Once the set of agents C_t has been chosen, each agent must gather information from other agents in the population. There are two issues: which other agents are accessible for information, and what type of information can be gathered.

We define an information gathering event as an *interaction*, and it is governed by the interaction relation Θ . The decision of what agents to interact with is influenced by the interaction cost. In much of the literature, interaction

cost is implemented as a social network in which vertices are agents and edges denote the probability of interactions between the agents at their ends. In some MAP work ([15], [4], [14]) agents choose interaction sets with a neighbor in their interaction relation, where neighbor-ness is defined by weight of the edge between them.

On the other hand, in PSO system and Olfati-Saber & Murry ([13]) agents interact with all of their neighbors. In this case, the edge weights do not indicate the probability of interaction, but rather the degree of influence of one agent on another.

Note that if the social network is not defined we assume that it is complete. Thus a situation where an agent picks some other agent at random from the population can still be modeled as a random choice from its neighborhood - which just happens to be every other agent in the population.

Once an agent decides on an interaction set, an interaction will take place. The purpose of an interaction between two or more agents is for the agents to gain information about each others' states. The information could be direct knowledge of the agents state, as in PSO systems or [13], or it could be based on a task that the agents must do. This latter case was modeled in [15]. Two agents played a game defined by a payoff matrix. Two games in particular were examined: the coordination game and the prisoners' dilemma game. In the coordination game the agents are given rewards based on whether their strategies are equivalent or not. If two agents have the same strategy, they are given a positive reward. If the two agents do not have the same strategy, they are given a negative reward. The reward, combined with an agent's knowledge of its own state and a memory of past rewards, provides the only information available about the strategy of the other agent.

In PSO systems and the systems studied by Olfati-Saber et. al., Σ is usually continuous - oftentimes it is the space of reals, \mathbb{R} . In this case direct knowledge of the strategy of the other agents can allow agents to find the "average" or "center" strategy. The use of such information is studied in the next section.

In these two stages the goal of an agent is to form an estimate of the state of the entire system. Since this is impossible, the agent relies on an approximation. This approximation is influenced by two factors, first the choice of agents affects the diversity of the approximation. The more potential agents to interact with - the better estimates of the state of the entire system. Secondly, the quality of information. The more precis information and agent gets from its interactions, the better its approximation of the systems state.

These two stages are the key factors in determining the complexity of an agreement problem.

2.2.3 Information Usage

In the final step, all agents that were active in this time step (including both C_t and I_t have the opportunity to change their strategy based on the information they have gathered in the previous step. Laying out the different types of uses of the information gathered is difficult, as there are a wide variety of methods

that could be used.

One division could be between *memoryfull* and *memoryless* systems. In a memoryfull system the agent has a memory that can store the information it has gathered over many time steps. In [15] each agent had a memory that could keep track of its strategy and the reward it received in the last k time steps. An agent decided whether to change its strategy or not by applying the Highest Cumulative Reward(HCR) rule. According to the HCR rule, an agent changes its strategy when the total reward in the past k steps for that strategy is greater than the total reward in the last k steps for the current strategy the agent is in.

Agents in a particle swarm optimization system also have a memory. Each agent can remember the best (according to the intrinsic value) state that it has ever been at.

On the other hand, in memoryless systems agents only remember the results of their just completed interaction. [13] is an example of a memoryless system - agents do not remember the result of interactions from before.

3 Convention Emergence in the DOA Framework

The DOA framework is general enough to represent many different situations described in the MAS community. As an example, we show how to map convention emergence situations (as described in [15]) into our DOA framework.

Shoham & Tennenholtz were interested in the emergence of *social conventions*. A social convention, as defined in [15], is:

A social law that restricts the agents' behavior to one particular strategy is called a (*social*) *convention*.

(emphasis in the original).

A social convention is agreed upon by everyone in the society. Thus it is an instance of an multi-agent agreement problem.

Shoham & Tennenholtz use the framework of *stochastic games* to explore the emergence of social conventions. At each time step two agents are chosen. The two agents play a 2-person-2-choice symmetric game. The agents can choose between two strategies, 0 and 1. The payoff matrix for this coordination game is:

$$M = \begin{pmatrix} 1, 1 & -1, -1 \\ -1, -1 & 1, 1 \end{pmatrix} \quad (2)$$

This payoff matrix means that when the two agents are using the same strategy they both get a positive reward. When they are using opposite strategies both agents will get a negative reward.

Each agent used the *Highest Cumulative Reward* (HCR) rule to determine whether to change its strategy. Each agent has a memory that allowed it to keep track of its last k strategies and the payoff each strategy received.

According to the HCR rule, an agent changes its strategy when the total reward in the past m steps for that strategy is greater than the total reward in the last m steps for the current strategy the agent is in. Shoham & Tennenholtz show, for a particular set of payoff matrices, that a population of agents using the HCR rule will reach a social convention, and stay in the social convention.

Mapping the Shoham & Tennenholtz model into the DOA framework is straightforward. First the set of agents, A is just the population of agents in S&T.

The possible agreement space, Σ is just the space of the two strategies that the agents can play. We will label them 0 and 1: $\Sigma = \{0, 1\}$.

There are no restrictions on the accessibility between states. Thus the accessibility relation will map every pair of states to 0. $\Delta(\sigma_i, \sigma_j) = 0 \forall \sigma_i, \sigma_j$

There are no restrictions on who an agent can interact with. Thus the interaction relation will specify the same, 0, cost for every pair of agents. $\Theta(\alpha_i, \alpha_j) = 0 \forall \alpha_i, \alpha_j$. This corresponds to an interaction graph that is complete - every agent can interact with every other agent.

In this model the intrinsic value of a state is not considered important. All that matters is that the population converges on some state. Thus the intrinsic value function will specify the same value, 0, for every possible agreement state. $\rho(\alpha_i, \sigma) = 0 \forall \alpha_i, \sigma$.

There are no restrictions on the start state.

The termination configurations are the states where all the agents agree upon the same strategy - the consensus problem.

We can see that mapping the problem into the DOA framework is relatively straightforward. Modeling their solution is also straightforward. At each time step two agents are picked to be active. This is done uniformly randomly and there is no choice in the matter for the agents chosen. Both C_t and I_t are chosen in one step.

In the information gathering stage the two agents chosen in the previous step play the stochastic game defined by the payoffs listed above. The payoffs do not indicate the state of the other agent, so we can consider this a case of gathering *indirect* knowledge of the other agents state (as we will see in the next section the discernibility of the system is low - agents do not get direct knowledge of another agents state ³)

Finally, the agents employ the HCR rule to update their state. This is mapped into the information use stage of the system dynamics.

Mapping the situation from [15] into the DOA framework has provided insight into the inner workings of the model. In the next section we summarize our results from mapping several different MAPs into the DOA framework and discuss the key factor in understanding the complexity of MAPs, the discernibility level of the system.

³in a 2-strategy case, though, it is actually direct knowledge. A payoff of -1 would indicate that the other agent has the opposite state. All results and simulations done in [15] assume a 2-strategy 2-agent game

	State Space Size	Accessibility	Interaction Graph	Intrinsic Value	Start Configuration	Termination Configuration	Agent Activation	Local Dynamics	Info. Gathering
<i>Distributed Commit Problem</i>	2	complete	complex	n/a	random	Consensus	Complete	Complete	Direct
<i>Consensus on Graphs</i>	∞	complete	complex	n/a	random	Consensus+Computation	Complete	Complete	Direct
<i>Particle Swarm Optimization</i>	∞	complete	complex	non-uniform	random	consensus+optimization	Complete	Complete	Direct
<i>Emergence of Conventions</i>	2	complete	complete	n/a	random	Consensus	Random	Single	Indirect
<i>Social Conventions in Complex Networks</i>	2	complete	complex	n/a	random	Consensus	Random	Single	Indirect
<i>Emergence of Socially Efficient Conventions</i>	2	complete	complex	non-uniform	random	Consensus+Optimization	Random	Single	Indirect

Figure 1: Examples of consensus problems. **Distributed Commit Problem** are from [9]; **Consensus on Graphs** are from [13] (in their work they study dynamic graphs - graphs where edges appear and disappear. This is modeled in the DOA framework by making the interaction relation change with time.); **Particle Swarm Optimization** systems are from [6]; **Emergence of Conventions** is from [15]; **Social Conventions in Complex Networks** is from [4]; **Emergence of Socially Efficient Conventions** is from [14]

4 The role of discernibility in MAPs

To gain an understanding of the similarities and differences between MAPs across domains we mapped several MAPs into the the DOA framework. Figure 1 summarized our results.

In a MAP the goal of a set of agents is to agree on a particular state. To achieve this each agent *must* make an approximation of the current configuration of the system - that is the states of all the agents. By making an accurate estimate of the configuration of the system an agent has the information to be able to change its state to minimize the time required to reach consensus.

To illustrate the necessity of an agent making this approximation let us consider the simplest possible scenario for a MAP. In this scenario there are only 2 agents in some possible agreement space. The fitness function specifies only a single high fitness state. In this situation each agent can independently arrive at the global optimum by performing a gradient ascent type search. There is no need for any agent to know the states of the other agents in the population - since both agents are doing gradient ascent it is guaranteed that they will converge at some point in time on the optimum.

Now let us expand the situation by adding just one other peak to the state space. So now there are two peaks with equal fitness. Under this situation it is no longer guaranteed that two agents independently following a gradient ascent type algorithm will converge on the same state. Because there are two optima, one agent may choose the optima A, and the other may choose optima B. Thus, in order to converge, it is necessary for the agents to estimate what peak its neighbors are going towards. Thus even in this extremely simple case, the agents must endeavor to estimate the configuration of the population.

The way an agent arrives at an approximation of the system configuration is specified by the global dynamics section of the DOA framework. Agents arrive at this approximation by interacting with other agents in the population and using

that information to change their own state. The quality of the approximation is determined by three factors, who they interact with (modeled in the *interaction choice*), what information they gather from these interactions (modeled by the *information gathering stage*) and finally how they use this information (modeled in the *information usage stage*).

Of these three, let us consider the information gathering stage. Issues in interaction choice are being explored in numerous places already (for instance [4]). The information usage stage is beyond the scope of this paper.

In all the models presented in Figure 1 the agents collected information about the state of neighboring agents. The only difference was in how accurate an estimate of the agents state it is. We call this the *discernibility level* of the system, and it measures how accurately an agent can discern the state of another agent.

In many of the scenarios in Figure 1 the discernibility level is very high, in fact agents could directly observe the states of other agents. In other instances discernibility is low, for instance in [15]. From this admittedly limited set of examples we start to see a pattern emerging. In the scenarios where there is low discernibility (any of the situations where the info. gathering column indicates “indirect”), the size of the possible agreement space is very small. On the other hand, when there is high discernibility (as in the first three cases), most of the time the state space is quite large⁴.

This emerging pattern that links the discernibility level of a system and the size of the potential agreement space led us to conjecture that the discernibility level plays a large role in determining the complexity of solving a MAP.

We demonstrate the role of discernibility by showing how differing levels of discernibility affect the convergence time in a Particle Swarm Optimization system.

5 PSO Model example

In a *Particle Swarm Optimization*(PSO) system multiple agents (called particles) navigate a high dimensional state space in search of high fitness states. The fitness of a state is defined by a globally known fitness function, f . Agents change their states by integrating information from their neighbors and from the history of states they have visited. The goal is for all the agents to converge on the optimal state (the one that has the highest fitness).

PSOs have been used to solve complex optimization problem in many areas and domains, see [5] for more information.

PSO systems map naturally into the DOA framework, as we show below. The goal of a PSO system is for all the particles to converge on the single best state in the state space, which is equivalent to the Consensus+Search termination condition from the DOA framework and the objective of a MAP. Because

⁴With the exception of the Distributed Commit Problem. This could be due to the fact that in the distributed commit problem messages between nodes could be lost, thus making it much more difficult to solve.

of their natural fit into the DOA framework and their relative simplicity, PSO systems make an illustrative domain in which to explore the effect of partial discernibility on convergence.

In the following we describe how PSO systems are related to DOA systems, describe how we implemented partial discernibility, and finally show results from our trials.

5.1 PSO systems in the DOA Framework

We do the mapping by going through each element of a PSO system and show its equivalent in the DOA framework. First we consider a DOA problem.

A swarm is composed of a set of particles, where each particle is identified by its position in the solution space (x_i) and its velocity (v_i). The set of particles is equivalent to the set of agents, A in the DOA framework. The solution space maps to the potential agreement space Σ . The velocity of an agent(particle) is part of its internal memory and will be discussed later.

PSOs do not define any cost to move from state to state - thus the accessibility relation Δ is uniform over all pairs of states.

Each agent(particle) can interact with only its neighbors in a social network. This network is equivalent to the interaction relation Θ in the DOA framework.

Each position in the solution space has a fitness defined by the fitness function f . The fitness of a potential solution is independent of the agent(particle). Thus the fitness function maps into the intrinsic value function ρ - where $\rho : \Sigma \rightarrow \mathbb{R}$.

The agents(particles) in a PSO can start from anywhere in the solution space. The start configuration of the system is randomly drawn from $\mathbf{S} = \Sigma^n$.

The goal of a PSO system is to have the agents(particles) converge upon the best state. This is equivalent to the Consensus+Search termination condition.

Now let us look at the system dynamics.

At each time step every agent in the population is chosen to be active, thus there is a complete agent activation: $C_t = A$. This is the Agent Activation stage of the system dynamics.

Each agents queries the agents in its neighborhood to see who has the best state. Thus I_t for an agent is equivalent to its neighbors in the social graph. An interaction involves one agent querying the other agent about its state and fitness. An agent gather information about who has the best state, and remembers the position of its neighbor with the best state. This is the interaction choice and information gathering stages of the system dynamics.

In the information usage stage an agent updates its state by applying this update rule:

$$\begin{aligned} v_i &\leftarrow \chi \left(v_i + U \left(0, \frac{\phi}{2} \right) \cdot (p_i - x_i) + U \left(0, \frac{\phi}{2} \right) \cdot (p_g - x_i) \right) \\ x_i &\leftarrow x_i + v_i \end{aligned}$$

where p_i is the best state that agent i has been at, and p_g is the best state of agent i 's neighbors. χ is a constriction coefficient and ϕ is an acceleration constant. χ is usually set to 0.729 and ϕ to 4.1 (for more information on these parameter settings see [1]). v_i is part of an agents internal memory.

We can see that PSO systems can be easily mapped into the DOA framework. Next we show how we implemented partial discernibility in the PSO system.

5.2 Partial Discernibility

Discernibility refers to how well an agent can make an estimate of another agents state via an interaction. Discernibility plays an important role in the information gathering phase of the global dynamics, as this is the only time in which agents interact.

In a PSO system the information gathering stage is where an agent finds its neighbor with the best state. In regular PSO systems agents have full discernibility - they get accurate information of their neighbors state.

To investigate the effects of partial discernibility we added random error to the information an agent gathers from its neighbors.

In the information gathering stage an agent evaluates the state of its neighbors and finds the best position based on fitness. This evaluation relies on particles being able to discern the state of another particle. We implemented partial discernibility by adding noise to this evaluation process.

When a particle evaluates its neighbors, it receives an estimate of each neighbors fitness based on a noisy version of its position. For instance, Particle 1 will get the impression that Particle 2 is at a position that is different from its actual position. The error is bounded by the "max error level" d which is a system wide parameter.

The error was implemented as random additive noise to each dimension of an agents state. Suppose agent i queries its neighbor agent j for its state. Given a max error level of d , agent j informs agent i that its state is (in the 2D case):

$$\begin{bmatrix} x_{j,0} + U(-d, d) \\ x_{j,1} + U(-d, d) \end{bmatrix}$$

Figure 2 visually depicts this.

Clearly, as the max error level increases the less likely the state returned by agent j is its true state. Thus, as the max error level *increases*, the discernibility of agent j 's state *decreases*.

5.3 Methodology

We studied the effects of varying max error levels (and thus discernibility) on the convergence of the particle swarm. We measured two quantities:

1. Whether the swarm had converged within 1000 iterations.

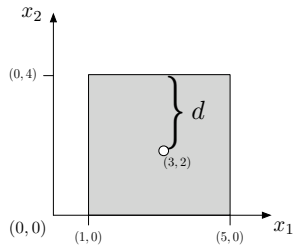


Figure 2: When queried about its location a particle will respond with a location within the grey cube. The parameter d determines the size of the hypercube.

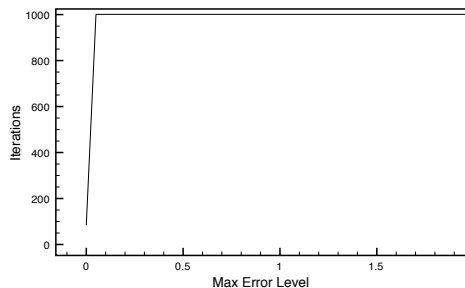


Figure 3: Convergence time. Results were same for max level of error greater than 2.0

2. At the end of the run, how well did the swarm converge (measured by the average of the the euclidean distance between every pair of particles in the swarm).

We explicitly looked at the effects on a very simple fitness function, the 2- d sphere:

$$f(x, y) = x^2 + y^2$$

This is the first function in DeJongs widely used five function testbed ([3]). There is one global optima at $(0, 0)$ with a fitness value of 0. By using a very simple fitness function we can study the effects of partial discernibility on the swarm and discount the influence of a complex fitness function.

The specific parameters we used were: $\chi = .4179$, $\phi = 4.1$, 10 particles, randomly initialized between $[-10, 10]$ for each dimension. The neighborhood of each particle was the entire population. We did 10 trials for each run and averaged the results.

5.4 Results

Figure 3 and Figure 4 show the results of our experiments.

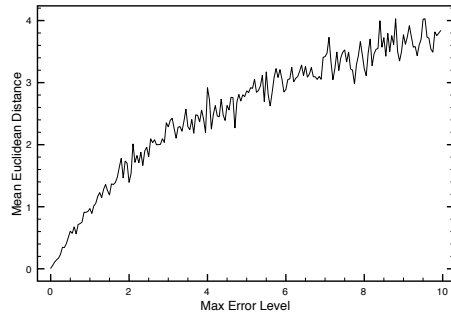


Figure 4: max error level vs. Mean Euclidean distance of population.

Figure 3 shows the number of iterations till a population successfully converged on the correct state. We considered a run to be a success if the agents had an average euclidean distance of less than .01 and the average fitness of the population was greater than $-.01$.

If the population did not successfully converge within 1000 iterations we used the value 1000. It is clear that the maximum error level had a devastating effect on the convergence time - with even very small levels of error the system would not converge within 1000 iterations. (we also tried to see if the population would converge within 3000 iterations and the results were similar).

Figure 4 graphs the mean euclidean distance between agents in the population (calculated by summing the distances between every pair of particles in the population then dividing by the number of particles) versus the max level of error. We can see a clear decrease in the clustering of the population as the max level of error increases. This corresponds to the particles not clustering.

We can clearly see that the max level of error - and thus the discernibility - make a huge impact on the convergence of the population of particles. The convergence of the particles relies on accurate estimates of the states of the other particles in the population. As more error was added the particles could not accurately estimate the state of their neighbors and thus could not converge.

From these simple experiments on PSO systems we can see that discernibility plays a huge role in the convergence of such a system. While the PSO system was very simple its results are quite interesting. Especially in light of the fact that these results were reached in situations where a very easy fitness function was used. More complex problems, with possible agreement spaces that cannot be dimensionalized, would prove to be much more difficult to solve as well!

6 Future Work

Language evolution is an area of substantial interest to us, and was in fact the inspiration to studying MAPs. Our future work is to show how the language convergence problem - the problem of having a society of agents come to an

agreement about what language to use - is in fact a type of a MAP. By considering language convergence as a *search* problem, we hope to leverage the results from other MAPs. This will be much easier to do with the use of the DOA framework which can serve as an interlingua between the different models.

Studying language convergence will also shed light on the effect of learning under varying discernibility levels. Humans coordinate meanings and lexicons in situations with very low levels of discernibility. Studying how discernibility affects the complexity of solving a MAP would bring us closer to understanding the language convergence issue.

7 Conclusions

While there has been much work in MAP problems for individual domains no unifying framework has emerged that can provide a MAP *lingua franca* that will allow us to leverage results between domains. In this paper we propose the framework to do just this - provide a unifying framework that can model many different agreement problems under one common descriptive domain.

The creation of the Distributed Optimal Agreement framework allowed us to compare and contrast MAP problems from a wide variety of disciplines. By mapping MAPs from different domains into the Distributed Optimal Agreement framework we discovered how MAP problems differ on dimensions including: *complexity of the potential-agreement space*, *state-to-state accessibility*, *agent interaction topologies*, *state evaluation measures*, and *type of solution needed* (to name a few).

A significant outcome of this comparison was the the discovery of a single critical differentiating factor that determines the difficulty of a MAP problem. The discernibility level - that is how accurately an agent can discern the state of another agent - has a large impact on the difficulty of solving a MAP problem. Our final contribution in this paper is to show how discernibility has an affect on the convergence time and quality of convergence in particle swarm optimization systems.

References

- [1] M. Clerc and J. Kennedy. The particle swarm - explosion, stability, and convergence in amultidimensional complex space. *Evolutionary Computation, IEEE Transactions on*, 6(1):58–73, 2002.
- [2] Felipe Cucker, Steve Smale, and Ding-Xuan Zhou. Modeling language evolution. *Foundations of Computational Mathematics*, 4(3):315–343, July 2004.
- [3] K. A. DeJong. *An analysis of the behavior of a class of genetic adaptive systems*. PhD thesis, University of Michigan, Ann Arbor, 1975.

- [4] Jordi Delgado. Emergence of social conventions in complex networks. *Artificial Intelligence*, 141(1-2):171–185, October 2002.
- [5] X. Hu, R. Eberhart, and Y. Shi. Recent advances in particle swarm. In *IEEE Congress on Evolutionary Computation*, 2004.
- [6] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, 1995.
- [7] Kiran Lakkaraju and Les Gasser. The complexity of finding an optimal policy for language convergence. In *Proceedings of the the Ninth International Conference on the SIMULATION OF ADAPTIVE BEHAVIOR (SAB'06)*, 2006.
- [8] Erez Lieberman, Christoph Hauert, and Martin A. Nowak. Evolutionary dynamics on graphs. *Nature*, 433(7023):312–316, January 2005. Supplementary notes in: lieberman05evolutionary-supp.pdf.
- [9] Nancy A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1997.
- [10] Frederick A. Matsen and Martin A. Nowak. Win-stay, lose-shift in language learning from peers. *PNAS*, 101(52):18053–18057, December 2004.
- [11] Pragnesh Jay Modi and Manuela Veloso. Bumping strategies for the multi-agent agreement problem. In *Proceedings of Autonomous Agents and Multi-Agent Systems, (AAMAS)*, 2005.
- [12] Martin A. Nowak, Natalia L. Komarova, and Partha Niyogi. Computational and evolutionary aspects of language. *Nature*, 417(6889):611–617, June 2002.
- [13] R. Olfati-Saber and R.M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *Automatic Control, IEEE Transactions on*, 49(9):1520–1533, 2004.
- [14] J.M. Pujol, J. Delgado, R. Sangüesa, and A. Flache. The role of clustering on the emergence of efficient social conventions. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence IJCAI-05*.
- [15] Yoav Shoham and Moshe Tennenholtz. On the emergence of social conventions: modeling, analysis, and simulations. *Artificial Intelligence*, 94(1-2):139–166, July 1997.
- [16] L. Steels. The origins of ontologies and communication conventions in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 1(2):169–194, October 1998.
- [17] Kyle Wagner, J.A. Reggia, J. Uriagereka, and G.S. Wilkinson. Progress in the simulation of emergent communication and language. *Adaptive Behavior*, 11:37–69, 2003.

- [18] A. Walker and M. J. Wooldridge. Understanding the emergence of conventions in multi-agent systems. In *ICMAS95*, pages 384–389, San Francisco, CA, 1995.