

The Statistical Mechanics of Belief Sharing in Multi Agent Systems

Robin Grinton, Katia Sycara, David Scerri, Paul Scerri
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA, U.S.A.
(rglinton, katia, dscerri, pscerri)@cs.cmu.edu

Abstract - *Many exciting, emerging applications require that a group of agents share a coherent view of the world given spatial distribution, incomplete and uncertain sensors, and communication constraints. This article describes an analysis and design methodology for distributed algorithms that coordinate the exchange of information for extremely large groups of agents maintaining a coherent belief in some environment property. The design methodology uses the tools of statistical mechanics to create a probability distribution which relates groupings of agents, called Sharing Groups, to pair-wise agent divergences and social temperature. Social temperature is a decision parameter, the same for all agents, that agents use probabilistically to decide when to join a Sharing Group. We show empirically as well as via Monte-Carlo simulations that for a critical value of social temperature the sharing groups formed result in bandwidth efficiency and divergence from ground-truth that is simultaneously optimal independent of the method of information exchange.*

1 Introduction

The increasing availability of cheap, low power sensors and processors has made large-scale, networked, cooperative autonomous sensor systems appropriate for a range of domains including distributed tracking [13], force protection [3], and locating lost hikers [5]. In such systems agents must act cooperatively in an incompletely observable environment. To do so effectively it is critical for agents to leverage their joint sensing ability to create an accurate view of the world. Creating joint beliefs about the world is highly communication intensive, since many individual sensor readings, over time, go toward creating a view of the world. However, in all practical systems the available communication bandwidth is strictly limited.

Our intuition is that efficient belief sharing will occur if agents are distributed into clusters where the average pair-wise belief divergence between agents is high. All sharing of beliefs between agents should occur in these clusters which we call Sharing Groups. In this work we concentrate on distributed algorithms which form Sharing Groups and exchange information between agents within them randomly. Forming Sharing Groups ensures economy of bandwidth usage by making sure that any messages exchanged have the maximum effect on lowering divergence between pairs of agents.

One of the key challenges in forming effective Sharing Groups is that agents must do so with limited knowledge of the rest of the team. We look to physical systems that act analogously to find mechanisms for forming effective Sharing Groups in a distributed way. There are many systems both natural and artificial that operate analogously e.g. ferromagnetism in a metal [11]. Ferromagnetism, or spontaneous magnetization, occurs when at a critical environmental temperature all atoms in a metal form groups of atoms which exist in the same state or *spin*. However, the spin of a given atom has the strongest effect on other nearby atoms, therefore the clustering is a result of the diffusion of *spin* information between neighboring atoms. Another example can be

found in sociology where segregation, a form of clustering, occurs in a population as a result of pair-wise cultural differences between individuals [18]. This work also showed that at critical social temperature, a parameter controlling the vigour of interaction, segregation is most evident in a society. Our hypothesis is that similar system dynamics will occur in belief sharing networks and can be exploited for the design of belief sharing policies. Specifically, it is typical for such systems to exist in distinct states or *phases* dependent upon a critical value of a system property. This system property, traditionally called *temperature* because the phenomenon was first discovered in physics, is a scale factor for the probability of interaction between individuals. Below the critical temperature T_c , clustering and order is observed and information diffuses between individuals. Conversely, above T_c most individuals exist in distinct states and information flow stagnates. Furthermore, at T_c such systems become ordered into clusters with minimal interaction between individuals. Striking similarities in the dynamics of systems where information diffuses as a result of nearest neighbor interactions have been shown through studies in a variety of fields [10, 18, 11]. The similarities in dynamics between these systems and a large scale system of agents maintaining a distributed belief by forming Sharing Groups suggests that order/disorder phase transitions likely occur when agents form Sharing Groups and use randomized information sharing policies.

Our design methodology uses the tools of Statistical Mechanics to develop a probability distribution which assigns probabilities to global configurations of assignments of agents to Sharing Groups. This probability distribution is a function of pair-wise divergences between neighboring agents as well as social temperature. Social temperature is a local decision parameter, with constant value common to all agents, that scales the probability with which agents interact. Specifically, at higher social temperatures agents are more likely to join Sharing Groups regardless of the resulting average pair-wise belief divergence within the new group formed. Conversely at lower temperatures agents are more likely to join groups when the average pair-wise divergence increases significantly. Statistical mechanics also provides us with the tools to find the critical value of social temperature at which bandwidth usage and divergence to the ground state over time, are simultaneously optimized.

In this article we present a formal model and methodology for analyzing and designing two types of multi-agent belief sharing algorithms distinguished by the mechanism used for information exchange. In the first type of algorithm, information is shared via nearest neighbor interactions only. In the second type, information is shared via token propagation. For both information exchange mechanisms we explore the existence of a critical temperature and a corresponding phase transition in system energy which is a measure of the *goodness* of Sharing Groups formed. Our results show that a critical temperature occurs independent of the mechanism employed by agents to share information. Based on the intuition gained through this analysis we present two specific algorithms corresponding to the two types of interaction dynamics. We show via Monte-Carlo simulation and empirically that by using either of these algorithms to share beliefs between agents optimal Sharing Groups form in a way as to minimize divergence and message bandwidth at the critical social temperature. Furthermore, We show that token propagation dynamics are inherently more bandwidth efficient while producing the same solution convergence as nearest neighbor algorithms.

The two models of statistical mechanics utilized in this work are the Potts Model [14], and the Random Cluster model [7], a generalization of the Potts model, both of which have been used to study clustering in ferromagnetism and segregation. These two models correspond to the two types of algorithms for belief sharing we are interested in studying. The first type, the local decision algorithm, is characterized by the fact that an agent can make the decision to exchange information with a neighboring agent at any time. We study this type of algorithm from a statistical mechanics perspective by modeling it as a Potts model. The Potts model allows us to probabilistically relate the Sharing Groups formed by agents to the social temperature and the pairwise divergence between agents. Our results show that a system of agents using this type of algorithm to share beliefs has a critical value of social temperature which results in the formation of Sharing Groups such that net divergence of beliefs across the agent system is minimal. Furthermore, we show that at the critical social temperature the agents exchange the fewest number of messages to reach convergence. However, our results also show that this type of algorithm is extremely inefficient with a large number of messages exchanged per agent even at the critical social temperature. We found that in terms of bandwidth efficiency, even at the critical social temperature, this type of algorithm is outperformed by both a naive token algorithm as well as by a simple broadcast algorithm.

The second type of algorithm, which we call the Advanced Percolation Token Propagation (APTP) algorithm,

uses token propagation [24, 23, 17, 16] to share information between agents. In this type of algorithm an agent can only make the decisions to incorporate a belief while holding a token. We found that the Potts model was not expressive enough to study a system utilizing this type of algorithm. Instead we used a Random Cluster model, a generalization of the Potts model. Like the Potts model, the Random Cluster model allows us to relate the social temperature and pair-wise divergence of agents to the distribution over global configurations of Sharing Groups. However, the Random Cluster model extends the Potts model with link variables that can be used to consider the relationship between the Sharing Groups formed and the links along which tokens propagate.

Our results show that a system of agents exchanging information via token propagation also has a critical social temperature where convergence first occurs. Furthermore, by comparison to a naive token algorithm, we show that using the critical social temperature as a decision parameter for token routing, reduces the number of token hops needed for convergence by an average of 68%. Our results also show that the token algorithm using social temperature as a decision parameter used only 2% of the messages used by the local decision algorithm to achieve convergence. Finally, we found that for the (A)PTP algorithm, the percentage improvement in bandwidth efficiency gained over all other approaches actually increased when starting with increasingly noisy and divergent agent beliefs.

1.1 Article Organization

The remainder of this article is organized as follows. Section 2 provides background on the key models used in the article. Section 3 formally describes the problem being addressed. Section 4 describes the types of communication networks used by agents in this work. Section 5 introduces Potts model, describing its key functions and properties. Section 5.1 maps the belief sharing problem to Potts model, then Section 5.2 describes a simple nearest neighbor algorithm followed by the agents, using the critical temperature as a decision parameter, which can be modeled solely as a Potts model. Section 5.3 shows empirical results for the nearest neighbor algorithm, illustrating that a critical temperature does indeed exist for the belief sharing system and comparing performance against other approaches. Results demonstrate that using the critical point produces the desired convergence of the algorithm but results in poor message passing efficiency because of the limited dynamics. Section 6 introduces the Random Cluster model, and Section 6.1 gives its formal relationship to the Potts model. Section 6.2 describes a token algorithm using the critical social temperature as a decision parameter. Section 6.3 shows empirical results which show that the token passing algorithm has the same desired convergence as the nearest neighbor algorithm but with greatly improved message passing efficiency. Sections 7 and 8. give related work, conclusions and future work.

2 Background

This section provides a brief introduction to token based algorithms and provides background information on the primary models of Statistical Mechanics used in this article, the Potts model and the Random Cluster model.

2.1 Token Based Algorithms

The key distinguishing feature of any token based algorithm for information sharing and coordination [17, 24, 23] is that all information, sharable resources, assignable tasks, and coordination interactions are encapsulated in *tokens* which are passed between agents.

There are two primary benefits of token based algorithms. The first benefit of token algorithms is that tokens provide access control to information and resources. The current holder of a token has exclusive rights to the information, resources, or tasks held by a token. This obviates the messages necessary to resolve conflicts in other types of distributed algorithms and hence reduces bandwidth.

The second benefit is that decisions are distributed between agents. A chain of agents propagating the same token effectively make a joint decision about the task or information the token represents. In a token based algorithm an agent receives a token and makes a decision about what to do with it based on strong knowledge

of its own current status, weaker knowledge about other agents based on previously received tokens, as well as the knowledge of how many times the token has been passed. This is in contrast with other algorithms with a central decision node, like auction algorithms, where a single agent has the responsibility for decision making. In such algorithms it is necessary for each agent to keep the decision maker abreast of its status. This creates both a central node of failure and a bandwidth bottleneck at the decision maker.

Token based algorithms for distributed multi-agent information sharing and coordination have been shown empirically to produce extremely efficient algorithms in terms of bandwidth usage and optimization of team utility [24, 23]. However, previous work has relied primarily on empirical evidence to investigate global system changes as a result of the spreading of information via token passing. This work is a first step in providing a formal link between the two.

2.2 Potts Model

For complex systems consisting of large numbers of interacting individuals or particles, the Potts model has been extensively used to statistically model the relationship between local interactions of individuals to global system properties[14, 4, 11, 18].

A Potts model can be defined for any system with the following characteristics:

- A group of individuals or particles, related through a neighborhood system.
- A finite set of states common to all individuals in which each individual can exist.
- Some type of interaction between neighboring individuals or particles which encourages homogeneity in their states.

For example in the case of ferromagnetism, each atom has a certain spin. When the spins of neighboring atoms are different a magnetic force is exerted between them which encourages the atoms to switch to the same spin. In the case of belief sharing neighboring agents can share information. A strong impetus to share information, like a magnetic force between them, encourages neighboring agents to have the same belief.

A Potts model of a belief sharing system allows us to analytically determine randomized policies of information exchange for individual agents which are optimal in terms of minimizing both the number of messages exchanged and degree of global belief divergence across the team. The Potts model allows us to relate the distribution over global assignments of agents to Sharing Groups to pair-wise belief divergence and social temperatures. Furthermore, via Monte-Carlo simulation of the belief share Potts model we can determine the critical value of social temperature, which governs information exchange, at which a clustering/no-clustering phase transition occurs. At this critical value the number of messages exchanged and global divergence are minimized. Thus, by determining this value we can use it in the design of an algorithm for agent interaction that both minimizes information divergence and minimizes communication.

Characterizing a system of belief sharing agents using a Potts model allows us to relate the degree of allowable global divergence in beliefs across the team to policies individuals use to control interaction with their neighbors. This is important because there are applications where complete homogeneity in the beliefs of individuals is undesirable and in fact sub-optimal. For example, in a tracking application, spatially distant sensors, might only need to know that a target is present in a distant sector but not its precise location. Thus, the model provides a powerful mechanism not available in previous approaches, specifically allowing a system to be tuned to minimize communication while achieving different divergence levels.

2.3 Random Cluster model

The Random Cluster model was introduced by Fortuin and Kasteleyn [7]. The Potts model is deficient in that it only allows for the representation of agent interaction dynamics where the influence of an agent extends solely to its nearest neighbors in a fixed communication network. In contrast, token passing algorithms allow the influence of a single agent to have direct influence over a chain of agents, those who receive and propagate a token. Such information sharing dynamics produce a degree of information distribution close to that which would be achieved through broadcast methods without incurring the exponential growth in messages with team size incurred by broadcast. However with token propagation Sharing Groups formed are also related to which links tokens were propagated through. The Random Cluster model can be thought of as an extension

of the Potts model with an added binary variable which represents a communication link as being passable or impassable to a token. With the addition of this variable the Random Cluster model can be used to relate global configurations of the assignment of agents to Sharing Groups to both the social temperature and to the clusters of links over which tokens are passed.

3 Problem Statement

This section formally describes the problem addressed by this article. Agents $A = \{a_1, \dots, a_m\}$ are a team with a joint objective in a partially observable domain. Decisions about actions by the agents are based on state variables $X(t) = \{x_1(t), \dots, x_n(t)\}$. There exists a logical network defined by a graph $G = (A, E)$ where agents correspond to vertices and $e \in E$ correspond to communication links. No assumptions are made about the structure of the logical network.

The agents have uncertain sensors, thus via some filter they must determine the probability of each of the state variables. Agent a_i 's probability distribution over X at time t is $P^i(X(t), t)$. For convenience, define $p^i(x_j(t), t)$.

The performance of the team will be adversely affected whenever their estimate of the state of environment varies from the actual state of the environment. The information difference (KL-divergence or similar) is $\Delta^i(X, P^i(X(t), t))$. The bigger this number, the higher the divergence. However, depending on their current activities, individual agents will not be equally effected by divergence in all variables. In general, they will only need to know precisely some values, while others can be coarsely understood or not known at all. Specifically, the cost of $\delta^i(\bullet)$ divergence to an agent a_i at a particular time is: $c(a_i, \delta^i(\bullet)) \rightarrow \mathcal{R}$.

Using their sensors, agents take sensor readings $r \in R$. A sensor reading influences $P^i(X(t), t)$ via some filter f , $P^{i'}(X(t), t) = f(P^i(X(t), t), r)$. The only assumption made about the filter is that estimates of variables improve monotonically with more sensor readings. Using the cost of information divergence and filter equations, the value of that sensor reading to a_i is $\hat{v}(s, a_i) = C(a_i, \Delta^i(X, P^i(X(t), t)), t) - C(a_i, \Delta^{i'}(X, P^{i'}(X(t), t)), t)$, i.e., the change in cost. We assume $\hat{v}(s, a) \geq 0$. The value of s to the whole team is: $\hat{V}(s) = \sum_{a \in A} \hat{v}(s, a)$. Thus, the overall optimization function to minimize is:

$$\sum_{a_i \in A} \int_{t=0}^T C(a_i, \Delta^i(\bullet)).dt + CommCost$$

where *CommCost* is the cost of communication. Intuitively, the team should receive reward by sharing the sensor reading unless the added value exceeds the communication cost.

4 Networks

In our model of belief sharing, agents communicate over a logical network. The topology of this network decides who an agent's neighbors are and consequently which agents it can communicate with directly. The four network topologies used were lattice, small world, scale free, and random. An example of each of these networks is shown in Figure 1. The following is a description of how each of these networks was created for the investigations described in this work.

For the lattice each agent is arranged on a regular grid and connected to the agents in the 8 neighboring grid cells. For the random network agents are paired with two other agent selected at random using a uniform distribution over the other agents. A small world network was created by placing agents on a regular grid and creating links to three other agents within a fixed Euclidean distance. One tenth of these connections were randomly removed and rerouted to a random node elsewhere in the environment [21]. The scale free network consists of several hubs, with connections to many agents[22].

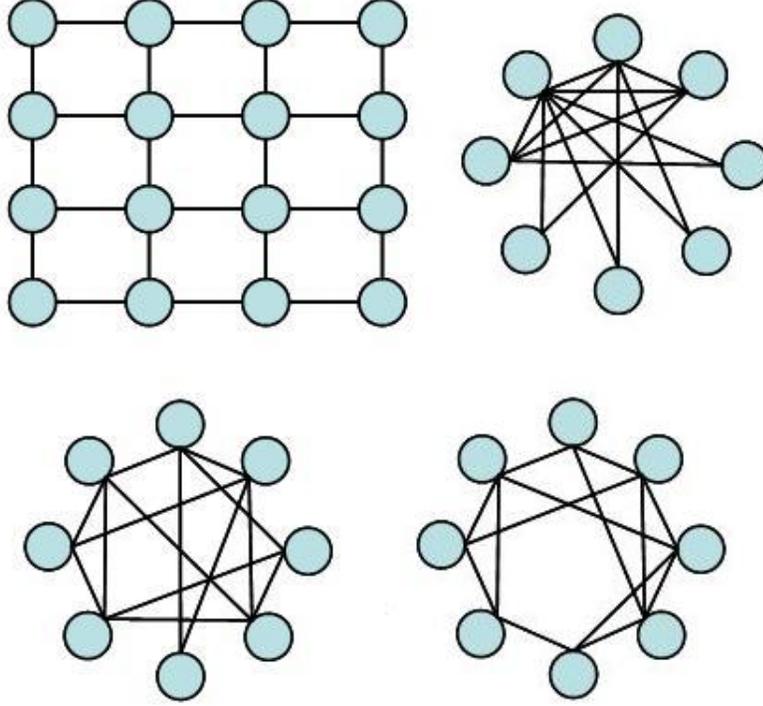


Figure 1: Examples of the four various network types. Clockwise from top left: Lattice, Scale Free, Small World, and Random.

5 Modelling Belief Sharing with Potts Model

A Potts model can be defined for systems of interacting particles that exist within a neighborhood system. The Potts model represents the system as a graph $G = (V, E)$, where $V = \{v_1, \dots, v_n\}$, the set of vertices, represent the individuals in the system under study. The set of edges $E = \{e_1, \dots, e_m\}$, determine which individuals are neighbors. Each v_i can take on a label from one of the l states taken from the finite set $Q = \{q_1, \dots, q_l\}$. The q_i correspond to the states that individuals can exist in. Alternatively these can represent an id for the groups that the individuals are a part of. The latter interpretation is most appropriate for this work.

The main quantity that must be defined in order to model a particular system as a Potts model is the *bond* energy function $d(e_{v_i, v_j})\delta_{q_i, q_j}$ where $e_{v_i, v_j} \in E$ and

$$\delta_{q_i, q_j} = \begin{cases} 1 & q_i = q_j, \\ 0 & q_i \neq q_j \end{cases}$$

is the Kronecker delta function which makes the energy of a bond connecting individuals in different groups zero. The bond energy function $d(e_{v_i, v_j})\delta_{q_i, q_j}$ gives a measure of how *comfortable* v_i and v_j are existing in the same group. For example, in the labelling problem from computer vision $v_i \in V$ would correspond to pixels in an image and $q_i \in Q$ would correspond to objects in the image (labels). The bond energy function $d(e_{v_i, v_j})$ would then likely be defined as a color difference between pixels so that the bonds connecting neighboring pixels that are closer in color have lower energy.

Consider the state space $S = \{q_1, \dots, q_l\}^V$ of possible assignments of states or group memberships $q_i \in Q$ to individuals $v_i \in V$. The total energy for the system is given by the Hamiltonian function $H(s)$, where $s \in S$ is a particular configuration of the system. The Hamiltonian is given by:

$$H(s) = \sum_{e_{v_i, v_j} \in E} d(e_{v_i, v_j}) \delta_{q_i, q_j} \quad (1)$$

The Potts formalism gives us Equation 2, the probability $P(s)$ over mappings of individuals to groups $s \in S$ that is a function of the Hamiltonian $H(s)$ evaluated at state s and system temperature T and k is Boltzmann's constant:

$$P(s) = \frac{e^{-\frac{H(s)}{kT}}}{\sum_{s \in S} e^{-\frac{H(s)}{kT}}} \quad (2)$$

Let us substitute a few variables to make this equation more manageable. First let us make the substitution $\beta = 1/kT$. Next let us substitute the normalizing constant of Equation 2 explicitly. Let us call the normalizing constant Z_p , where Z_p is defined as:

$$Z_p = \sum_{s \in S} e^{-\beta H(s)}$$

Then equation 2 reduces to:

$$P(s) = \frac{1}{Z_p} e^{-\beta H(s)} \quad (3)$$

Re-substituting $H(s)$ from Equation 1 into Equation 3 we get:

$$P(s) = \frac{1}{Z_p} e^{-\beta \sum_{e_{v_i, v_j} \in E} d(e_{v_i, v_j}) \delta_{q_i, q_j}} \quad (4)$$

Intuitively T dictates how disordered the system is. At higher values of T states with higher energy bonds are possible and just about any pair of individuals can exist in the same group regardless of the bond energy. For low values of T states s with very low energy bonds are probable. These are states where individuals assigned to the same groups are very comfortable. In the image example this would correspond to a state where almost all pixels were assigned the correct object labels.

The critical temperature T_c is the temperature below which the system is most likely to have minimal energy. That is when most individuals in the system are most likely to be assigned to the correct groups.

An example of a Potts model where each individual can exist in one of three states and a neighborhood system defined by a two dimensional lattice is show in Figure 2. This example is intended to give an intuition for how the model works. The bond energy in this case is simply a constant $-J$ for bonds that link individuals occupying the same states and zero when the two states differ. States q_i are represented by color in the diagram and the divergence metric is given by, $d_{q_i, q_j} = -J \delta_{q_i, q_j}$ where:

$$\delta_{q_i, q_j} = \begin{cases} 1 & q_i = q_j, \\ 0 & q_i \neq q_j \end{cases}$$

5.1 Belief Sharing Mapped to Potts Model

The intuition behind mapping a belief sharing system as a Potts model is as follows. We wish to achieve a joint belief with as few messages as possible. The best way to do this is to make sure that every message exchanged between two agents has the maximum effect in terms of equalizing beliefs across the system. We define a bond energy function such that agents with great divergence in beliefs are *happier* when in the same group. We can then exchange beliefs within such groups and attain a joint belief with good bandwidth efficiency. This type of grouping might seem counterintuitive because one is used to thinking of grouping *like* things. However, for equalizing beliefs it makes more sense to group agents with large belief divergence and have message exchanges to equalize beliefs within those groups.

$$H(S) = \sum_{edges} -J\delta_{s_i, s_j} = -7J$$

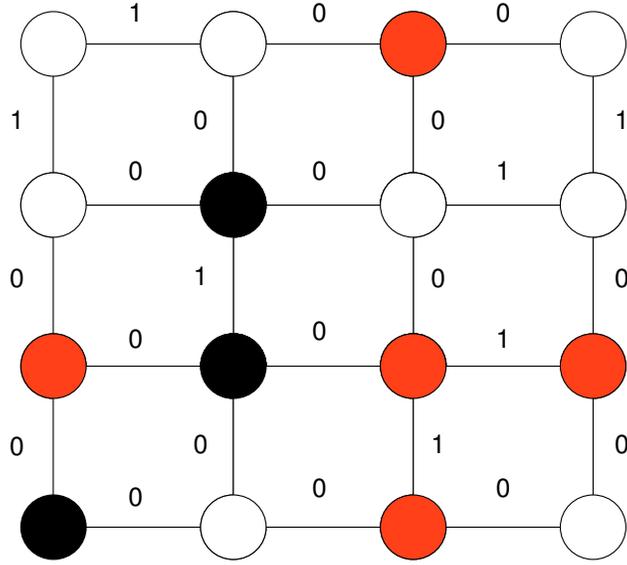


Figure 2: Hamiltonian for a three state Potts model with a neighborhood system defined by a two dimensional lattice. States (or group memberships) of individuals are indicated by the color of the corresponding site in the lattice. J is a constant which gives the energy in a bond between individuals occupying the same state.

Next we formally define the mapping of the quantities of interest in the belief sharing problem to the important functions of the Potts model. N agents maintain a distributed belief. There is a communication network defined by the set E where $\exists e_{i,j} \in E$ if a communication link exists between agents i and j . The belief of agent i is represented by a vector $X^i(t) = \{x_1^i(t), \dots, x_n^i(t)\}$ where $x_j^i(t) \in \mathcal{R}$. Ground truth is given by $X^g = \{x_1^g, \dots, x_n^g\}$. At time t agent i takes readings $R^i(t) = \{r_1^i(t), r_2^i(t), \dots, r_n^i(t)\}$ readings are generated by adding zero mean gaussian noise to the ground truth according to $r_j(t) = x_j^g + N(0, \sigma_k)$. Agents share information by exchanging these readings. Each agent i maintains a filter f that is simply a maximum likelihood estimate of $x_j^i(t)$. That is $x_j^i(t) = \frac{1}{t-t_0} \sum_{t_0}^t r_j^i(t)$. That is, an agent maintains an estimate of its belief by averaging over all of the readings available to the agent, both its own and those received from its peers.

As a measure of information divergence we use the mean square error MSE as a metric. Formally the divergence between the belief of agent i and ground truth is given by $\sum_{j=1}^n (x_j^i(t) - x_j^g(t))^2$ while the divergence between the beliefs of agents a_i and a_j is given by $d(i, j) = \sum_{j=1}^n (x_j^i(t) - x_j^k(t))^2$. The value $d(i, j)$ maps to $d(q_i, q_j)$ the energy in the link between neighbors as defined in Section 5. Using this measure of divergence the Hamiltonian of the belief share system is given by:

$$H = \sum_{i,j} d(i, j)\delta_{i,j}$$

where

$$\delta_{i,j} = \begin{cases} 1 & \exists e_{i,j} \in E, \\ 0 & \nexists e_{i,j} \in E \end{cases}$$

and E is the set of communication links between agents.

Substituting into Equation 4 we get the probability $P(H)$ of the distributed belief having total divergence or energy as defined by H above:

$$P(H) = \frac{1}{Z_p} \exp\{-\beta \sum_{e_{i,j} \in E} d(i,j)\delta_{i,j}\} \quad (5)$$

Equation 5 tells us that the system is more likely to exist in states where the belief divergence of the system is smaller. If we could evaluate Equation 5 exactly we could answer questions about the system by calculating expectations over system variables. Most important to this work, we could find the critical system temperature T_c . Recall that T_c is the temperature below which the system energy is minimal with high probability. Using Equation 5 we could calculate the expected value of H , the total energy of the system as $E(H) = \sum_s P(s)H(s)$ for a range of values of T . T_c would then be the value of T for which the value of H first becomes minimal on the resulting graph of H versus T . However, calculating Z_p from Equation 5, which normalizes the distribution, is generally undecidable for even the smallest systems. This is because the computational complexity of evaluating the denominator, commonly known as the partition function, is exponential in the number of possible ways to assign states to individuals. For this reason we turn to the Metropolis algorithm [12] which allows us to calculate expectations over H using samples of states s drawn according to the distribution $P(H)$.

5.2 Local Decision Algorithm

In this section we introduce a distributed algorithm for exchanging beliefs between agents that results in a system where the probability of a certain amount H of belief divergence in the system is given by Equation 5. We achieve this by developing a distributed version of the Metropolis algorithm [12]. The Metropolis algorithm is commonly used to draw samples from an arbitrary probability distribution. Our distributed version exchanges beliefs between agents such that the resulting belief divergence of the system occurs with a probability distribution given by Equation 5 and hence the belief divergence in the system is small with high probability. Furthermore by drawing samples from Equation 5 for $T = T_c$ the result is low belief divergence for a minimal number of interactions between agents.

The distributed algorithm operates by having agents randomly decide to exchange readings weighted by the change in system belief divergence (energy) that would result if the two agents shared the same belief. An agent can detect the change in system energy resulting from its change in state locally since the total system energy is given by the sum of the bond energy between nearest neighbor agents.

In the distributed version of the Metropolis algorithm each agent a_i runs Algorithm 1. In Lines 2 and 3 of the algorithm an agent first selects an integer at random from a uniform distribution on the range $[0, N]$, where N is the total number of agents in the system. The agent proceeds to exchange beliefs only when the selected integer is 0. The purpose of this step is to ensure that on average only a single agent in the system is exchanging beliefs at every time step. In lines 4-6, an agent selects a neighbor at random and requests the current belief of that agent. Upon receiving this belief the agent a_i calculates the divergence $d(i, j)$ between its belief and the belief received from the selected neighbor a_j . Using lines 9-11 the agent then sends readings to neighbor a_j with a probability that is exponential in the divergence $d(i, j)$. In lines 12-14 an agent will check to see if it has received any readings from its neighbors and incorporate them into its own belief. Section 5.1 gives details on how agents calculate divergence and incorporate readings into their beliefs.

5.3 Experiments

This section presents experimental results for a simulation of a system of agents, where, on each iteration of the simulation, each agent, in a fixed order, runs Algorithm 1 once. For the experiments we have used a simulator

Algorithm 1: Algorithm used by agent to decide when and with whom to share information

BELIEFSHARE($\beta, nlist, N, readings$)

```

(1)   while true
(2)      $r \leftarrow \text{UNIFORMRANDOM}([0, N])$ 
(3)     if  $r = 0$ 
(4)        $i \leftarrow \text{RANDOMINTEGER}(\text{SIZE}(nlist))$ 
(5)        $n \leftarrow nlist.get(i)$ 
(6)        $n.belief = \text{REQUESTBELIEF}(n)$ 
(7)        $d \leftarrow \text{DIVERGENCE}(n.belief, belief)$ 
(8)        $r \leftarrow \text{UNIFORMRANDOM}([0, 1])$ 
(9)        $p \leftarrow \text{EXP}(-\beta * d)$ 
(10)      if  $r < p$ 
(11)         $\text{SEND}(readings, n)$ 
(12)       $readings \leftarrow \text{GETRECEIVEDREADINGS}()$ 
(13)      if  $readings \neq \emptyset$ 
(14)         $\text{INCORPORATEREADINGS}(belief, readings)$ 

```

which allows us to vary different variables in order to examine their effect. Communication between agents is simulated, and assumed to be perfect, to allow many experiments to be performed.

In each of the graphs the model was run for 1000 iterations for each value of kT and the resultant energy averaged over 10 runs. Figures 3 - 4 show the energy (belief divergence) on the Y-axis and $\log(kT)$ on the X-axis. Any exchange of information between agents during the experiments described in this work is referred to as a message. This includes the exchange of readings and beliefs. Many of the results presented in this section appear in [8].

Figure 3 shows the results for four different network implementations. These networks are described in Section 4. The key feature of this graph is that in each case there was a clear and large phase shift, occurring at the same critical point for each network. This supports the key hypothesis of this article that the system dynamics of a Potts model apply to a belief sharing problem. The networks had close to the same performance, but both the lattice and small-worlds networks had lower divergence after the critical point, probably due to the clustering in these network types.

Figure 4 shows the results of varying the number of agents in the team. Team sizes of 100, 400, 900 and 1600 were used. The critical point, where the phase transition occurs, is at the same location for each of the team sizes. This may be somewhat surprising, since one might expect bigger teams to require more interaction to reduce divergence, but the local nature of the interactions reduces the impact of team size.

Figure 5 shows the number of messages between agents next to the resultant energy. The left Y axis shows the energy and the right Y axis shows the number of messages after 1000 iterations. The x axis shows the change in kT . It is clear that the phase transition occurs at a value where a substantial number of messages are sent. This is important because predicting this number of messages, about 100,000 messages in the figure, has not been possible to estimate in the past. Even the ability to control the number of messages (and thus get this optimal behavior) has not been previously possible.

Figure 6 shows how the Potts derived model compares to a token based model[24] and a broadcast model. The token model was implemented by creating five tokens which contained a belief and a visit counter. The agent averaged his belief with the belief contained in the token, incremented the visit counter and passed the token on to a random neighbor. The broadcast model consisted of random agents 'broadcasting' their belief to 20% percent of the population. Agents who received a broadcast belief averaged it with their own. Notice that the Potts derived model uses more messages and is thus less efficient than the other two models. In the remainder of this article we introduce a new algorithm, the APTP algorithm, based on the Random Cluster model which has the same convergence properties as the Local Decision algorithm but uses an order of magnitude fewer messages to achieve convergence.

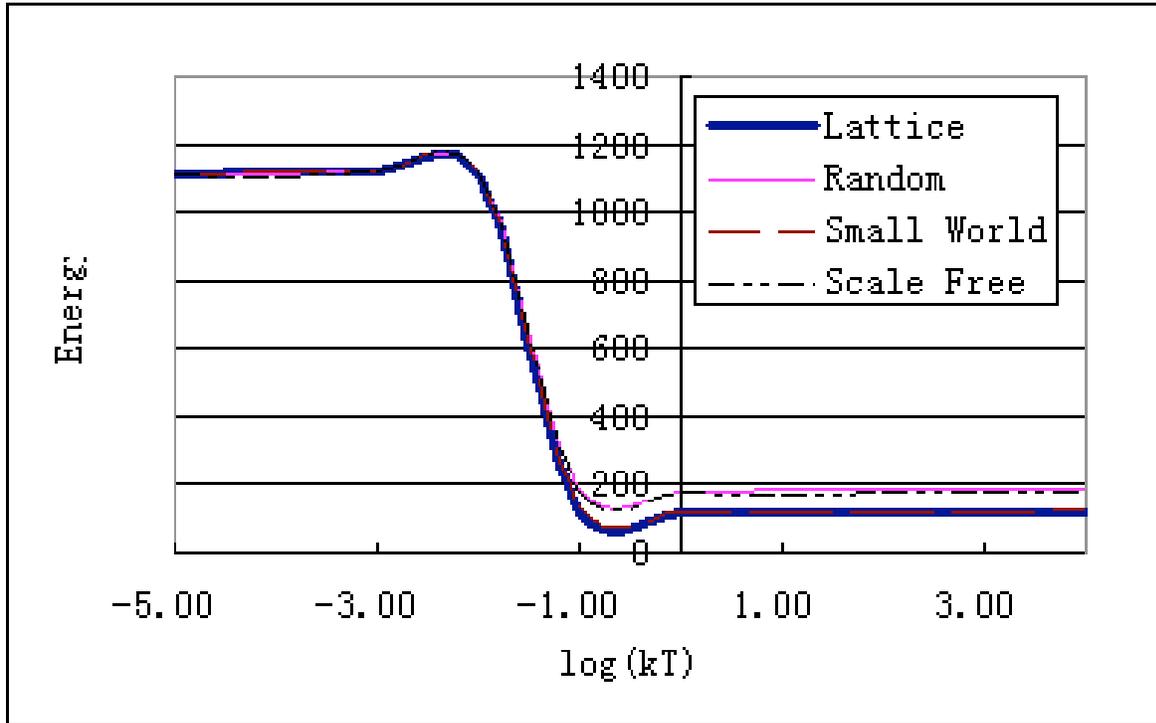


Figure 3: Comparing performance across network types.

6 The Random Cluster Model

It is our goal to develop an algorithm with the efficiency of a token algorithm and the ease of analysis and the tight guarantees on system utility of a Potts model. In Section 5 we showed that a belief sharing system could be modeled by a Potts model. However, the Potts model allowed only the modeling of the influence of agents on nearest neighbors in a communication network. By passing tokens, an agent can directly influence an entire set or cluster of agents that are visited by the token. It is desirable to incorporate token passing into the model of belief sharing because sharing information by token passing greatly increases the bandwidth efficiency of a multi agent system. In contrast, the random cluster model provides a natural representation of the agents influenced by the passing of a token. The mathematics of the random cluster model is based on a graph with edges that can be open with probability p closed with probability $1 - p$. Where $d(e_{v_i, v_j})$ is the pair-wise divergence measure between the beliefs of agents v_i and v_j that was introduced in Section 5. As shown in Figure 7 sub-graphs or clusters of vertices are formed by vertices connected by open edges. The random cluster model provides a probability distribution ϕ over configurations of the states of graph edges. Figure 7 shows one such configuration of open and closed edges and the sub-graphs formed.

Consider tokens being passed between agents in a multi agent system with a fixed communication network. If an agent decides to pass a token to a peer with probability p we can think of the corresponding communication link connecting the agents as open. By propagating tokens in this way, the agents visited by a token would form sub-graphs of the communications network and ϕ would give the probability that the token would visit a particular sub-graph.

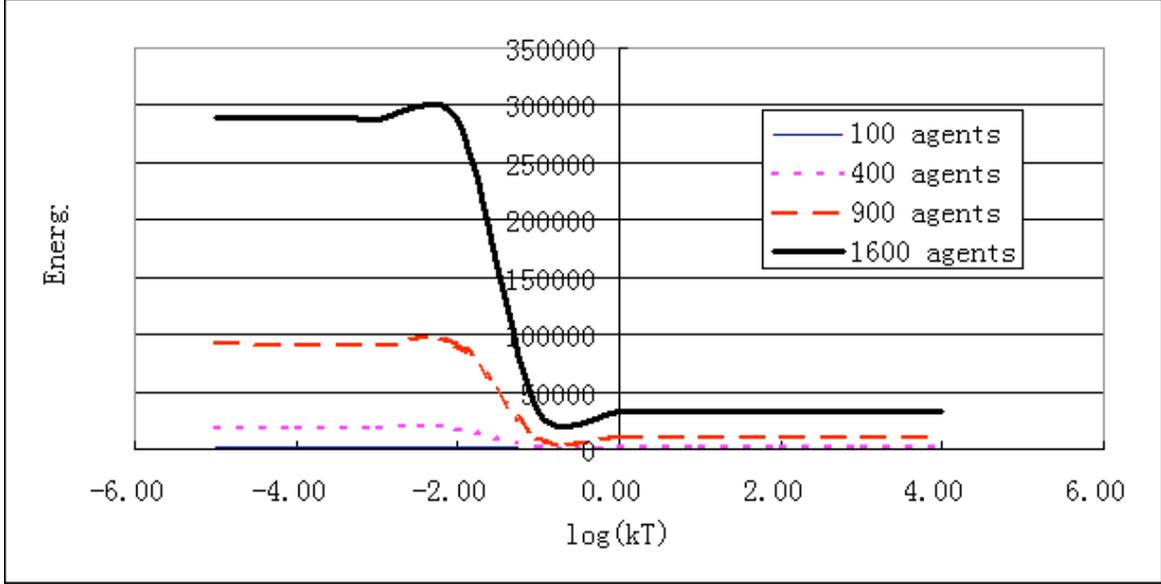


Figure 4: Performance as team size is varied.

The following is a formal description of the random cluster model [7] and ϕ the probability distribution over configurations of open and closed graph edges. Consider a graph $G = (V, E)$ with vertex set V and edge set E . The state space of interest in the random cluster model is the space of all possible assignments of 0 (closed edge) and 1 (open edge) to every $e \in E$. Herein we refer to the space of configurations of graph edges as $\Omega = \{0, 1\}^E$. Finally for a particular $\omega \in \Omega$ let each graph edge $e \in E$ be open with probability p , $\omega(e) = 1$, or closed with probability $1 - p$, $\omega(e) = 0$. With the variables of interest thus defined the probability mass function of a particular configuration of open and closed edges, $\omega \in \Omega$, is assigned by the probability mass function ϕ_p .

$$\phi_p(\omega) = \frac{1}{Z} \prod_{e \in E} (p^{\omega(e)}(1-p)^{1-\omega(e)})q^{k(\omega)}$$

Where Z the normalizing constant for the distribution is given by the partition function below:

$$Z = \sum_{\omega \in \Omega} \prod_{e \in E} (p^{\omega(e)}(1-p)^{1-\omega(e)})q^{k(\omega)}$$

and $k(\omega)$ gives the number of connected sub-graphs formed by the open and closed edges.

6.1 Relationship to the Potts model

In section 6 we showed that the Random Cluster model provides a probability distribution ϕ over the agents in the sub-graph of the communication network that would be visited by a particular token. This equation is

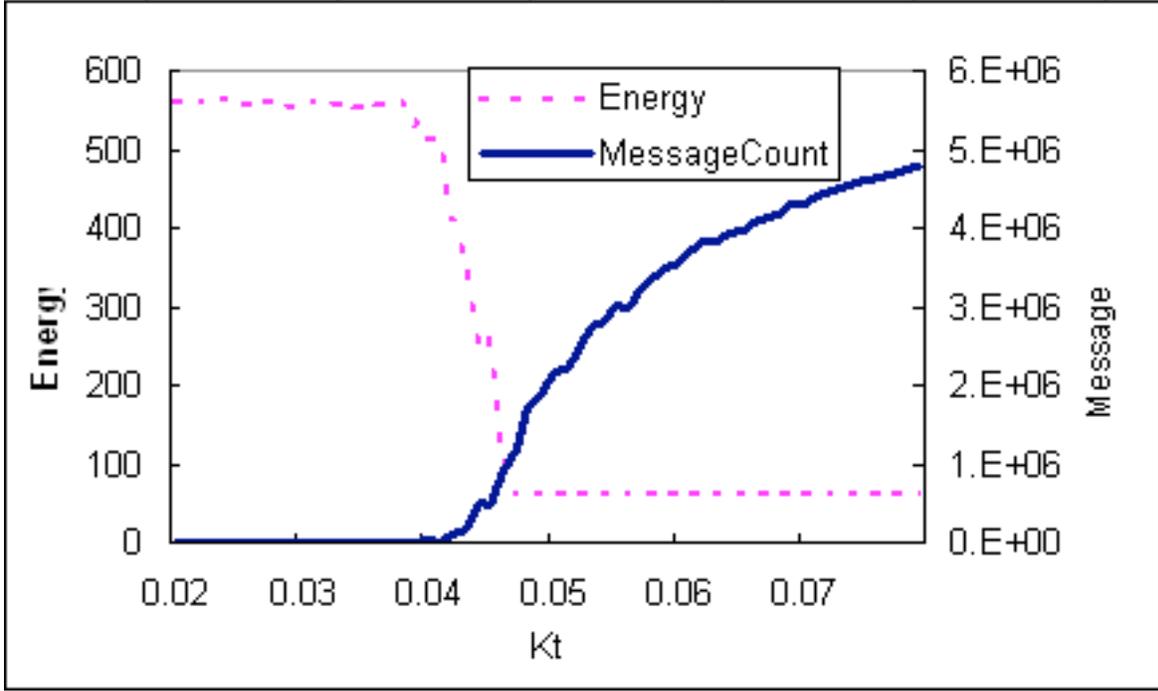


Figure 5: System energy and messages as the temperature is varied.

valid for agents that make a decision to pass the token with probability p . When a token visits an agent and the agent incorporates the belief held by the token into its own, this means that the agent has joined the Sharing Group that is made up of the other agents visited by the same token.

By formulating a joint distribution over the states of both graph vertices and edges, Edwards and Sokal [6] provided the link that we can use to show how the movement of tokens is related to the probability of a given assignment of agents to Sharing Groups given by Equation 5.

Recall that $q = \{q_i, \dots, q_l\}$ represent ids for Sharing Groups that agents can join and p the probabilities that communication links are open or closed to the passage of tokens. Furthermore $\Omega = \{0, 1\}^E$ gives the space of possible configurations of open and closed communication links. The joint sample space over both sets of variables is: $S \times \Omega$ where $S = \{q_i, \dots, q_l\}^V$. Further consider the vector $\omega \in \Omega$. The entries of ω correspond with graph edges such that $\omega(e) = 1$ indicates that edge e is open and conversely $\omega(e) = 0$ indicates that edge e is closed.

The Random Cluster model gives a probability distribution function μ over the joint space $S \times \Omega$ given by:

$$\mu(\sigma, \omega) = 1/Z_{RC} \prod_{e_{i,j} \in E} \{(1-p)\delta_{\omega(e),0} + p\delta_{\omega(e),1}d(e_{v_i,v_j})\delta_{q_i,q_j}\} \quad (6)$$

Equation 6 gives the joint probability distribution over $S = \{q_1, \dots, q_l\}^V \times \Omega = \{0, 1\}^E$ where $s \in S$ give the possible configurations of the assignment of agents $v_i \in V$ to Sharing Groups $q_i \in Q$ and $\omega \in \Omega$ give the state of communication network edges, (1) open or (0) closed.

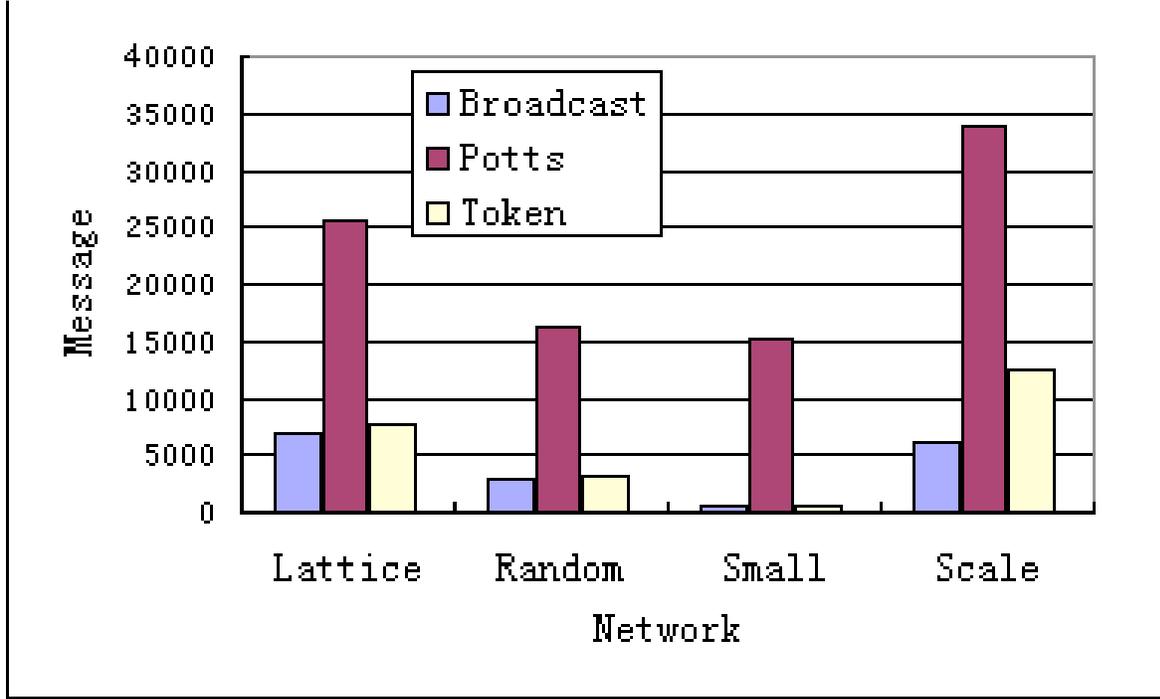


Figure 6: Comparison with other approaches.

Edwards and Sokal [6] based on the work of Swendsen and Wang [19] gave two important results that follow from the choice of $p = e^{-\beta d(e_{v_i, v_j}) \delta_{q_i, q_j}}$ where p is the probability that an edge is open to the movement of a token and $d(e_{v_i, v_j}) \delta_{q_i, q_j}$ gives the pair-wise divergence between the beliefs of agents v_i and v_j . The first result is that $Z_p = Z_{RC}$. That is that the normalizing constant for Equation 6 the Random cluster distribution and Equation 5 the Potts distribution are the same. The second result is that by marginalizing the joint distribution given by Equation 6 over S , the result is the distribution:

$$\mu_1(\sigma) = \frac{1}{Z_p} \exp\left\{ \beta \sum_{e_{v_i, v_j} \in E} d(e_{v_i, v_j}) \delta_{q_i, q_j} \right\} \quad (7)$$

Recall that this is the Potts distribution given by Equation 4, as described in Section 5 which gives the distribution over assignments of Sharing Group ids q_i to agents v_j given the divergence metric between agent beliefs $d(e_{v_i, v_j})$.

$$\mu_2(w) = \frac{1}{Z_{RC}} \left\{ \prod_{e \in E} p^{w(e)} (1-p)^{1-w(e)} \right\} q^{k(w)}$$

These results lead to Theorem 1 due to Edwards and Sokal [6] which we can use to construct conditional probabilities of assignments of agents to Sharing Groups given a particular configuration of edges opened and closed to the movement of tokens.

Theorem 1

”For $w \in \Omega$, the conditional measure $\mu(S|w)$ is obtained by putting a random Sharing Group id q_i on entire clusters of w (these are sub-groups of agents of agents that are formed by open edges and are visited by the same token. There are $k(w)$ such clusters). These ids are constant on given clusters, are independent between clusters, and each is uniformly distributed on the set $\{q_i, \dots, q_l\}$.”

This theorem tells us that tokens being passed between agents with probability p result in a particular configuration s of the assignment of agents to Sharing groups that is a sample of Equation 7. This distribution is more likely to return low energy configurations. Since we defined our bond energy as the negative of the belief divergence between agents, this means that Sharing Groups with higher net belief divergence are more likely when tokens are propagated in this way.

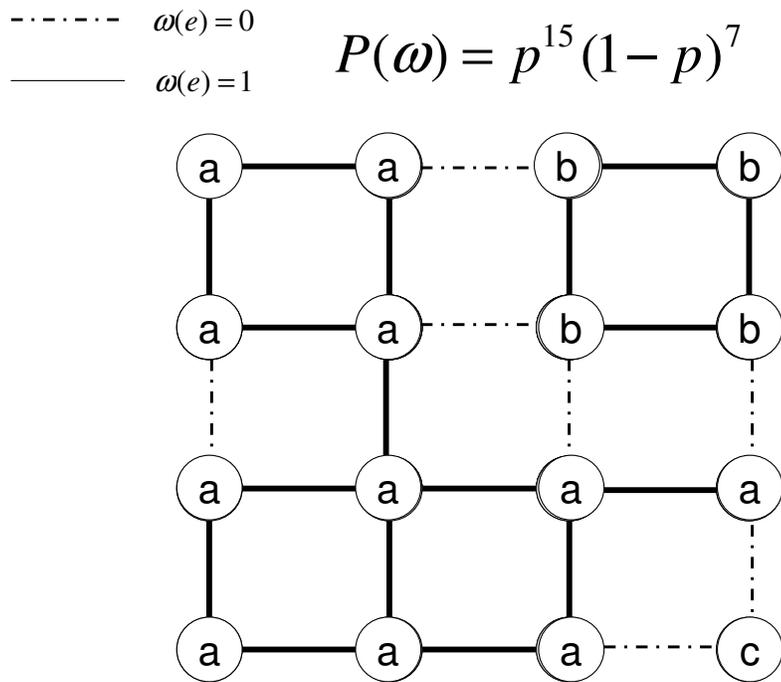


Figure 7: Key elements of the percolation model, which consists of a graph with particular static configuration ω of edges, open and closed independently with probability p . $P(\omega)$ gives the probability of a given configuration ω of closed and open edges. On the diagram one can see that the open and closed edges induce sub-graphs or *percolation-clusters*. Percolation clusters in the diagram are identified by having the same letter id. The diagram shows three percolation clusters a,b, and c.

Figure 7 shows a particular configuration $\omega \in \Omega$ of a percolation graph. Notice that the open and closed edges result in distinct sub-graphs or percolation-clusters of vertices. Figure 7 shows three distinct percolation-clusters

labeled a,b, and c. In this work beliefs would be exchanged within percolation clusters.

6.2 Advanced Percolation Token Algorithm

Using the results of Section 6.1 we construct the APTP algorithm, Algorithm 2, which draws samples from the joint Potts distribution over assignments of agents to Sharing groups given by Equation 7.

The algorithm has two conceptual stages. In the first stage tokens are propagated to mark communication links to neighboring agents as opened or closed with probability:

$$p = e^{-\beta d(e_{v_i, v_j}) \delta_{q_i, q_j}} \quad (8)$$

This is equivalent to opening and closing bonds in the Random Cluster model described in Section 6. Several such tokens are created at random within the system. The second conceptual step is that agents use tokens to exchange beliefs along open edges. In practice we use the same token which marks links to propagate beliefs between agents. This effect is achieved by having agents pass a token to a neighbor, the neighbor who receives the token decides with probability given by Equation 8 whether the link was open or closed. If the agent decides that the link the token was received on was in fact closed it destroys the token. Otherwise it propagates it at random to a neighbor who repeats the process. Either way the agent incorporates the belief contained on the token.

Algorithm 2: Algorithm used by agent for token creation and propagation

BELIEFSHARE($\beta, nlist, tokenQueue, numTokens, N, maxNetworkDiameter$)

```

(1)  while true
(2)    token ← GETNEXTTOKEN(tokenQueue)
(3)    if token ≠ NULL
(4)      d ← DIVERGENCE(token.belief, this.belief)
(5)      r ← UNIFORMRANDOM([0, 1])
(6)      p ← 1 - EXP(-β * d)
(7)      if r < p
(8)        INCORPORATEBELIEF(token.belief, this.belief)
(9)        ADDBELIEFTOTOKEN(token, this.belief)
(10)       token.TTL --
(11)    else
(12)      r ← UNIFORMRANDOM([0, N])
(13)      if r < numTokens
(14)        token ← CREATETOKEN()
(15)        ADDBELIEFTOTOKEN(token, this.belief)
(16)        token.TTL = maxNetworkDiameter
(17)      if token ≠ NULL
(18)        if token.TTL ≠ 0
(19)          n ← UNIFORMRANDOM([0, numNeighbors])
(20)          SEND(token, n)

```

Algorithm 2 induces percolation clusters on the agent system associates network like the ones shown in Figure 7. These clusters are formed among agents with very divergent beliefs. The token also moves beliefs between these agents. Agents randomly introduce tokens into the system to replace those that die. This process is illustrated in Figure 8.

There are two important things to note about the operation of the APTP algorithm. The first is that intuitively it is very efficient in terms of message passing. In an iteration of the algorithm all sharing of information occurs between agents whose beliefs are particularly divergent. Furthermore, the fact that the information is propagated via token passing greatly reduces bandwidth usage since only a few tokens are ever being passed around simultaneously.

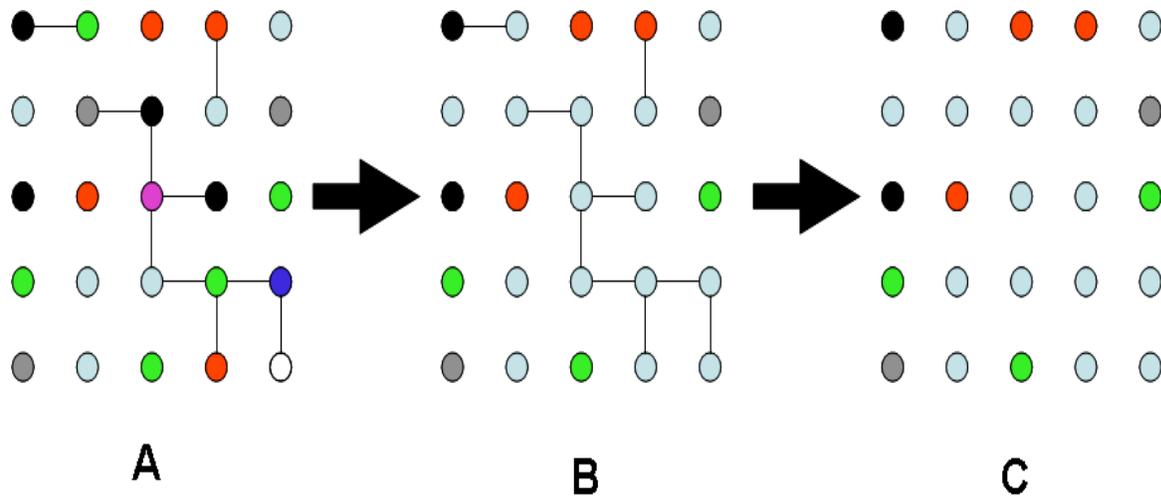


Figure 8: Illustration of the token passing algorithm using Sharing Groups. As shown in the diagram, groups of open edges form clusters of agents with very divergent beliefs. Divergent beliefs are illustrated by different colors. Tokens also propagate beliefs throughout these Sharing Groups (clusters) and the resulting divergence in the cluster is low after an iteration of the algorithm.

6.3 Experiments

We conducted experiments to evaluate the convergence properties, bandwidth efficiency, and resiliency to noise and error of the APTP algorithm. Unless specified, all parameters are the same as those for the experiments described in Section 5.3.

To investigate the resiliency of the APTP algorithm to situations where individuals might have no knowledge of the environment or situations where an individual's belief contains no information (a faulty sensor perhaps) we created certain areas of the network that had no readings for some beliefs. This 'spatial blackout' was created by randomly choosing a node in the graph and for each neighbor within a certain range, setting all of the readings for a given belief to 0. The range of this blackout was proportional to the size of the network, from 6 nodes in a 100 node network to 20 nodes in a 1000 node network.

The number of nodes in the experiment varied from 100 to 1000. In the experiments to compare the network type and compare the critical value, team sizes of 100, 400 and 1000 nodes were used. For the experiment comparing the Advanced Percolation Model to the Simple Token model, team sizes of 100, 200, 300, 400, 500, 600, 700, 800, 900 and 1000 were used.

In two of the experiments the critical temperature was fixed at 0.7 and in the third experiment the temperature was varied logarithmically to show its effect on the algorithm. Values from 0.00001 through to 10000 were used.

A simple token algorithm (different from the one evaluated in Section 5.3) was devised as a benchmark against which to compare the APTP algorithm. Initially a number of tokens were created and placed randomly around the network. As with the APTP algorithm the number of tokens for each team size increased from 5 tokens for 100 nodes through to 25 tokens for 1000 nodes. At each step of the algorithm, each token was randomly passed on to a neighbor of the node holding the token. The node which received the token then averaged each of its beliefs with the beliefs contained in the token so that both the token and the node held the same belief.

Figure 9 is a plot of the net divergence to ground truth of agent beliefs vs. the log of the product of boltzmann's constant and the social temperature. This plot shows that as predicted the agents sharing beliefs by token propagation using Algorithm 2 have a critical social temperature where a phase transition from high divergence to low divergence occurs.

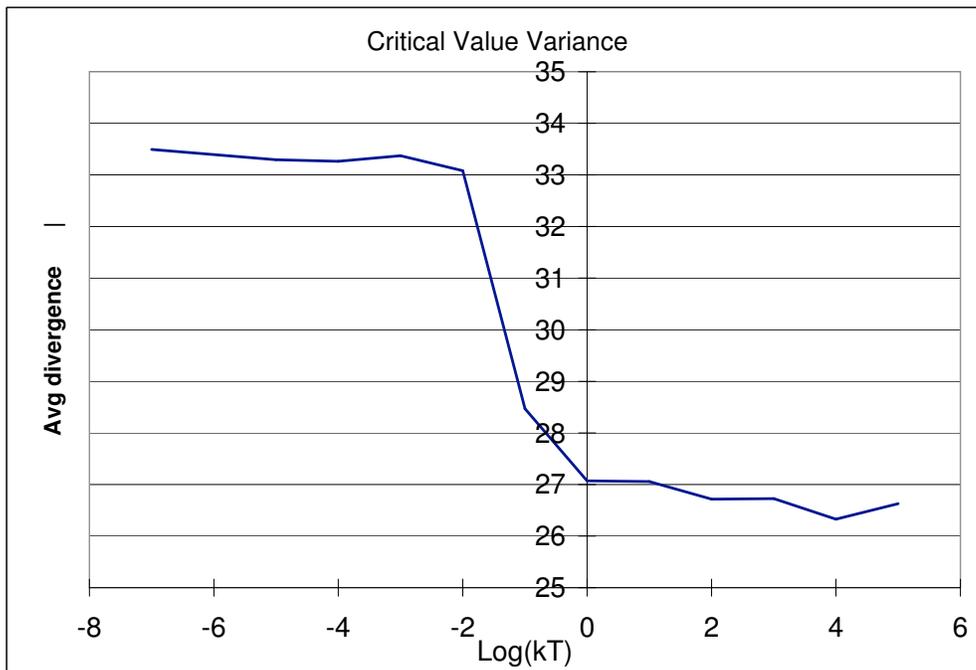


Figure 9: Plot of average agent belief divergence with respect to ground truth vs. the product of boltzman's constant and social temperature. The figure shows that a phase transition in divergence occurs for a critical value of social temperature.

Figure 10 gives the result of the experiment comparing the simple token algorithm to the APTP algorithm, Algorithm 2. The graph shows that APTP algorithm required an average of 32% of the messages required by the simple token algorithm to arrive at the same level of net belief divergence.

Figure 11 shows the number of messages required for the average belief divergence to ground truth of all agents to fall below 1%. As expected the number of messages required increases with increasing numbers of agents. However the relationship between number of agents and number of messages is linear suggesting that the algorithm is scalable. Not surprisingly the bandwidth efficiency is worst for the lattice network topology

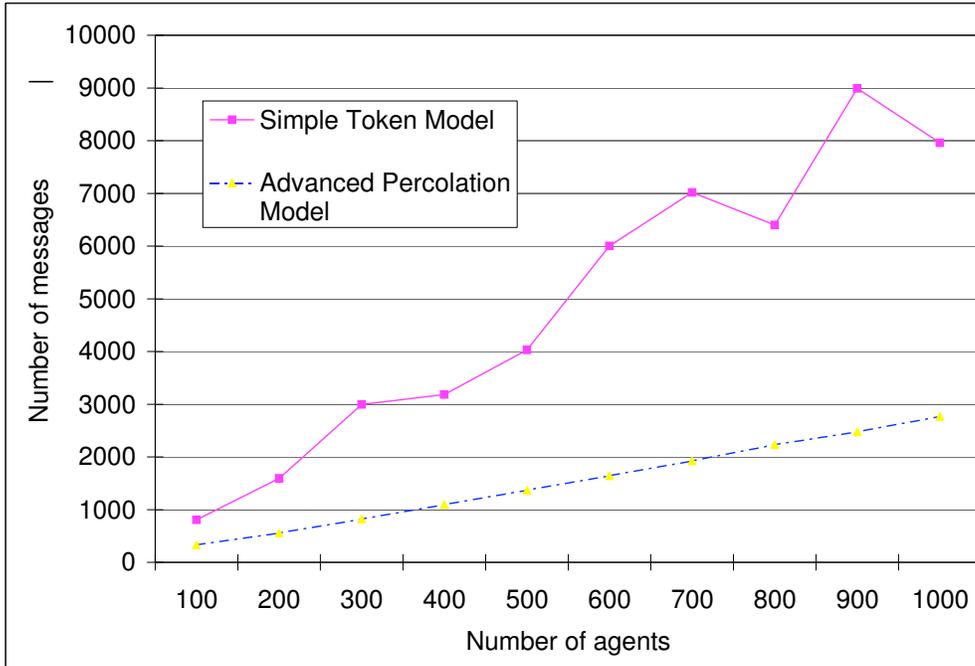


Figure 10: Plot of the number of messages required to arrive within 1% of ground truth vs the number of agents

because it has the greatest average number of hops connecting an arbitrary pair of agents. It is unusual that the random network outperforms the other topologies in terms of bandwidth efficiency. We suspect that this is because the long hop nodes of the small worlds and scale free networks are encountered by tokens more often than other nodes and as a result have more of an influence on the joint belief formed than other nodes. This means if the belief at such a node was particularly noisy then this would result in slower convergence of beliefs. By comparison to Figure 6 we can see that the APTP algorithm uses an order of magnitude fewer messages than the local decision algorithm to achieve the same convergence.

7 Related Work

There has been recent interest in the use of decentralized Bayesian filters such as the ones proposed in [9, 2] to manage beliefs over a large team. Communicating these beliefs, however, is expensive, prompting several selective communications approaches. Divergence metrics such as Hellinger affinity and KL-divergence are commonly used to measure the information gain of individual communications. However, existing methods of integration such as channel managers [1] or query-based particle filters [15] face scaling issues, in these cases, dealing with redundant data and polynomial-time scaling with team size, respectively. In contrast our approach provides excellent convergence with bandwidth usage that scales linearly with team size.

Decision theoretic approaches have also been successfully used for small teams [20]. Consensus protocols over communication networks are another method to achieve consistent belief among agents. Multi-hop relay protocols have been demonstrated to allow for fast consensus seeking [25], but previous work has focused on

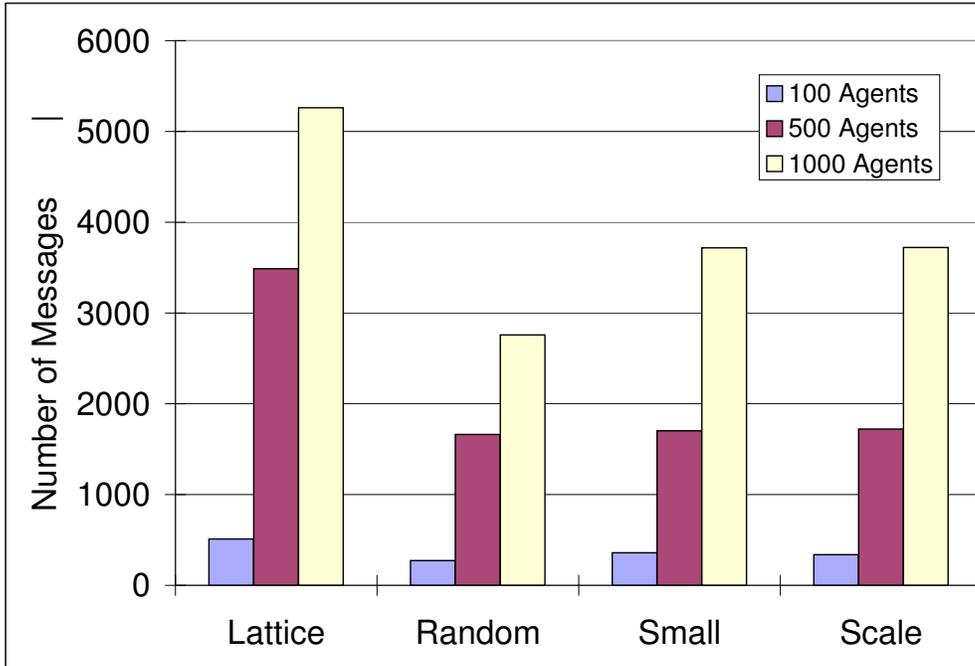


Figure 11: Bar chart showing the number of messages required for the average belief divergence of agents to arrive within 1% of ground truth for varying numbers of agents and four different network topologies

trading robustness and convergence, rather than information gain and overall network traffic, which is the focus of our investigation. Consequently unlike these other approaches, our approach allows system designers to trade off the desired level of consensus with the bandwidth usage of the system. This is important because for many applications complete convergence is not required and hence using excess bandwidth to achieve this would be wasteful. For example in a sensor network monitoring temperature over a large area, it would make sense that only sensors that are spatially close maintain a coherent reading of temperature.

Uses of the Potts model in the literature include statistical models of ferromagnetism in metals[11], the explanation of the phenomenon of segregation in socio-economic models [18], and the study of political districting [4].

8 Conclusions and Future Work

This article presented a novel approach to designing and analyzing token-based belief sharing algorithms using selected mathematical models of Statistical Mechanics. These algorithms utilized a single control parameter for all agents called social temperature which is a measure of how eager agents are to interact. We demonstrated that forming agents into groups of agents with high net pair-wise divergence called Sharing Groups was an effective strategy for producing low divergence and high bandwidth efficiency. Furthermore, we showed that there exists a critical value of social temperature, independent of the information exchange mechanism employed. Finally we showed that at the critical temperature a phase transition between low and high belief divergence

occurred.

These findings can be used in the development of fusion systems with very large numbers of nodes (1000s), when it is desirable that all nodes maintain a common view of a state space by fusing their limited and or noisy individual views, in a distributed fashion. For example, in very large scale sensor networks monitoring the environment or in tracking applications using multiple sensors. Using the techniques developed in this work, designers of such systems can achieve reliable convergence of nodes to a common picture of the environment with very efficient bandwidth usage.

There are several extensions to this work that we wish to pursue. The first direction we wish to proceed is to analyze the effect of altering the policies for sharing beliefs within Sharing Groups. Next we plan to consider extending the method to do distributed probabilistic inference via token propagation. This would allow for fast inference with low bandwidth.

References

- [1] F. Bourgault and H Durrant-Whyte. Communication in general decentralized filter and the coordinated search strategy. In *Proc. of FUSION'04*, 2004.
- [2] F. Bourgault, T. Furukawa, and H Durrant-Whyte. Decentralized bayesian negotiation for cooperative search. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.
- [3] L. Chaimowicz and V. Kumar. Aerial shepherds: Coordination among uavs and swarms of robots. In *in 7th International Symposium on Distributed Autonomous Robotic Systems*, 2004.
- [4] C. Chou and S. Li. Taming the gerrymander, statistical physics approach to political districting problem. *Physica A Statistical Mechanics and its Applications*, 369:799–808, September 2006.
- [5] J. L. Drury, J. Richer, N. Rackliffe, and M. A. Goodrich. Comparing situation awareness for two unmanned aerial vehicle human interface approaches. In *in Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics*, 2006.
- [6] R.G. Edwards and A.D. Sokal. Generalization of the fortuin kasteleyn swendsen wang representation and monte carlo algorithm. *Phys. Rev. D*, 38:2009–2012, 1988.
- [7] C.M. Fortuin and P.W. Kasteleyn. On the random cluster model. i. introduction and relation to other models. *Physica*, 57:536–564, 1972.
- [8] R. Ginton, P. Scerri, D. Scerri, and K. Sycara. An analysis and design methodology for belief sharing in large groups. In *Proceedings of the Tenth International Conference on Information Fusion FUSION'07*, 2007.
- [9] B. Grocholsky. *Information-Theoretic Control of Multiple Sensor Platforms*. PhD thesis, The University of Sydney, 2002. Available from <http://www.acfr.usyd.edu.au>.
- [10] Katsutoshi Hirayama, Makoto Yokoo, and Katia P. Sycara. The phase transition in distributed constraint satisfaction problems: First results. In *Proceedings of the 6th International Conference on Principles and Practice of Constraint Programming*, pages 515–519, 2000.
- [11] Epele L.N., Fanchiotti H., and C.A. García Canal. Ferromagnetism and antiferromagnetism in the Potts model in $(1 + 1)$ and $(2 + 1)$ dimensions. *Physical Review B*, 25:1997–1999, February 1982.
- [12] N.A. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.
- [13] C. L. Ortiz, R. Vincent, and B. Morisset. Task inference and distributed task management in centibots robotic systems. In *AAMAS*, 2005.

- [14] R. Potts. Some generalized order-disorder transformations. *Proceedings of the Cambridge Philosophical Society*, 1952.
- [15] M. Rosencrantz, G. Gordon, and S. Thrun. Decentralized sensor fusion with distributed particle filters, 2003.
- [16] P. Scerri, E. Liao, Yang. Xu, M. Lewis, G. Lai, and K. Sycara. *Theory and Algorithms for Cooperative Systems*, chapter Coordinating very large groups of wide area search munitions. World Scientific Publishing, 2004.
- [17] P. Scerri, Yang. Xu, E. Liao, J. Lai, and K. Sycara. Scaling teamwork to very large teams. In *Proceedings of AAMAS'04*, 2004.
- [18] T.S. Schelling. Dynamic models of segregation. *Journal of Mathematical Sociology*, 1:143–186, 1971.
- [19] R.H. Swendsen and J.S. Wang. Nonuniversal critical dynamics in monte carlo simulations. *Phys. Rev. Lett.*, 58:86–88, 1987.
- [20] M. Tambe. Agent architectures for flexible, practical teamwork. In *National Conference on AI (AAAI97)*, pages 22–28, 1997.
- [21] Duncan Watts and Steven Strogatz. Collective dynamics of small world networks. *Nature*, 393:440–442, 1998.
- [22] Yuri I. Wolf and Georgy Karevand Eugene V. Koonin. Scale-free networks in biology: new insights into the fundamentals of evolution? *BioEssays*, 2002.
- [23] Y. Xu, M. Lewis, K. Sycara, and P. Scerri. Information sharing in very large teams. In *In AAMAS'04 Workshop on Challenges in Coordination of Large Scale MultiAgent Systems*, 2004.
- [24] Y. Xu, P. Scerri, B. Yu, S. Okamoto, M. Lewis, and K. Sycara. An integrated token-based algorithm for scalable coordination. In *AAMAS'05*, 2005.
- [25] Jin Zhipu. *Coordinated control for networked multi-agent systems*. PhD thesis, California Insitute of Technology, 2006.