Stochastic Self-organization

Timothy D. Barfoot[*]
Controls and Analysis, MDA Space Missions,
9445 Airport Road, Brampton, Ontario L6S 4J3, Canada

Gabriele M. T. D'Eleuterio
Institute for Aerospace Studies, University of Toronto,
4925 Dufferin Street, Toronto, Ontario M3H 5T6, Canada

Starting from a random initial predisposition, the creation of a common piece of information in a network of sparsely communicating agents is the first step towards showing how order can be built from the bottom up rather than imposed from the top down. The model used for this study has been termed stochastic cellular automata (SCA). We identify three important elements of self-organization using SCA: instability, averaging, and fluctuations, all of which have been identified in other nonlinear phenomena. From a practical point of view, a simple coordination mechanism for a group of autonomous agents or robots is provided.

1. Introduction

A common thread in all multiagent systems is the issue of coordination. How are a large number of sparsely coupled agents able to produce a coherent global behavior using simple rules? Answering this question will not only permit the construction of interesting and useful artificial systems but may allow us to understand more about the natural world. Ants and the other social insects are perfect examples of local interaction producing a coherent global behavior. It is possible for millions of ants to act as a "superorganism" through local pheromone communication [1].

With ants as our inspiration, we look to create self-organizing coordination mechanisms for large-scale artificial systems. In particular, we are motivated to carry out truly decentralized decision making in a group of sparsely communicating autonomous robots such as those shown in Figure 1. Our interest in groups of robots stems from planetary exploration applications such as distributed sensing [2]. To maximize robustness, it is extremely important that every robot is treated exactly equally. This removes the possibility of having a leader robot. Moreover, we cannot even afford to have a centralized "polling station" wherein votes are

[*] Work carried out while at the University of Toronto Institute for Aerospace Studies.

Figure 1. Examples of small groups of autonomous mobile robots. The coordination mechanisms studied would benefit small groups such as these but are also intended to scale to much larger groups.

tabulated to elect a leader. Agreeing upon a leader, or any other piece of information, must be accomplished through self-organization.

The word *self-organization* is used in many contexts when discussing multiagent systems, which can lead to confusion. Here we use it to mean: multiagent coordination in the face of more than one alternative. We will be describing a stochastic version of cellular automata (CA). The goal will be to have all cells agree on the same symbol from a number of possibilities using only sparse communication. An example can be found in Figure 2. We maintain that rules able to succeed at this task are self-organizing because the cells are not told which symbol to choose, yet they must all coordinate their choices to produce a globally coherent decision. If we told the cells which symbol to choose, the task would be very easy and no communication between cells would be necessary. This can be dubbed *centralized organization* and is in stark contrast to self- or decentralized organization. We believe that coordination in the face of more than one alternative is at the very heart of all multiagent systems.

The difference between centralized and decentralized decision making is already familiar to most people but likely not thought of in this way. It is essentially the difference between flipping a coin and using "rock, paper, scissors" when two people are trying to resolve a dilemma. In flipping a coin, both parties rely on the coin (a centralized agency) to make the decision and then abide by its ruling. In "rock, paper, scissors" a decision is also quickly achieved if both people follow the same rules, but it occurs in a decentralized manner. Note, however, in "rock, paper, scissors" it can take repeated trials to finally come to a consensus as there can be a tie. In fact, there is no guarantee that a consensus will ever be achieved. But, we may compute the (high) probability with which a decision is made after a certain number of trials.

This paper is organized as follows. Related work is described, followed by a description of the model under consideration. Results of its performance on the multiagent coordination task are presented. Analyses of the model are provided followed by discussions and conclusions.

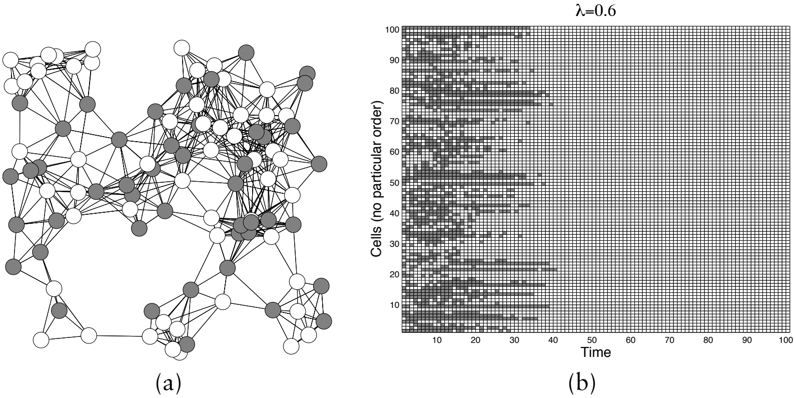

Figure 2. (a) Example of 100 SCA. Cells are indicated by circles and bidirectional connections by lines. The color of each cell represents its state chosen from an alphabet of size $K = 2$. (b) An example of how the states change over time. In this case a consensus is reached as demonstrated by all cells adopting the same color in a single time-slice (column).

2. Related work

Note that unless explicitly stated, the term *cellular automata* (CA) will imply determinism (as opposed to stochasticity) in this section.

Von Neumann [3] originally studied CA in the context of self-reproducing mechanisms. The goal was to devise local rules which would reproduce and thus spread an initial pattern over a large area of cells, in a tiled fashion. The current work can be thought of as a simple case of this where the tile size is only a single cell but there are multiple possibilities for that tile. Furthermore, we wish our rules to work starting from any random initial condition of the system.

CA were categorized by the work of Wolfram [4] in which four *universality classes* were identified. All rules were shown to belong to one of these classes: Class I (fixed point), Class II (oscillatory), Class III (chaotic), or Class IV (long transient). These universality classes can also be identified in stochastic cellular automata (SCA) and we will show that in our particular model, choosing a parameter such that the system displays long transient behavior (e.g., Class IV) results in the best performance on our decentralized coordination task. Wolfram [5] provides some examples of CA involving probabilities.

Langton [6, 7] has argued that natural computation may be linked to the universality classes. It was shown that by tuning a parameter to produce different CA rules, a phase transition was exhibited. The relation between the phase transition and the universality classes was explored. It was found that Class IV behavior appears in the vicinity of the phase transition. The current work is very comparable to Langton's study in that we also have a parameter which can be tuned to produce different CA rules. However, our parameter varies the amount of randomness that is incorporated into the system. At one end of the spectrum, completely random behavior ensues while at the other completely deterministic behavior ensues (which is simple voting). We also relate the universality classes to particular ranges of our parameter and find a correlation between performance on our decentralized coordination task and Class IV behavior. We attempt to use similar statistical measures as Langton [6] to quantify our findings.

Mitchell *et al.* [8] and Das *et al.* [9] study the same coordination task that is examined here in the case of deterministic CA. However, their approach is to use a genetic algorithm to evolve deterministic rules successful at the task whereas here hand-coded stochastic rules are described. They found that the best solutions were able to send long range *particles* [10] (similar to those in the *Game of Life*) to achieve coordination. These particles rely on the underlying structure of the connections between cells, specifically that each cell is connected to its neighbors in an identical manner. The current work assumes that no such underlying structure may be exploited and that the same mechanism should work for different connective architectures. The cost for this increased versatility is that the resulting rules are less efficient (in terms of time to coordinate) than their particle-based counterparts.

Tanaka-Yamawaki *et al.* [11] study a similar problem to that considered here. They use *totalistic* [4] rules which do not exploit the underlying structure of the connections between cells but rather rely on the intensity of each incoming symbol. They vary a parameter to produce different rules and find that above a certain threshold, "global consensus" occurs but below it does not. The connectivity between cells is regular and success was found to depend in part on the connective architecture used. However, they consider large clusters of symbols to be a successful global consensus. For our practical problem of coordinating a group of agents, clusters indicate that consensus has not been reached and thus does not permit us to use these results directly. The Tanaka-Yamawaki study does not consider all possible deterministic totalistic rules and thus is not a complete view of how such rules fair against the problem at hand.

For the problem of developing a practical coordination mechanism we decided it would not be possible to rely on the types of regular connections between nodes used in the earlier studies mentioned. For a group of mobile robots, the connections were more likely to be based on proximity as communication is typically local. Totalistic rules were attractive to us, however, because it is impossible to predict exactly how many robots will be within range of one another. Moreover, the connections change as the robots move. Dynamic connections implied to us that we could not have a period in which the structure of the connections could be learned and then coordination optimized for that particular structure. In the end we found the work of Tanaka-Yamawaki to be most similar to our problem but decided to turn to a stochastic version of their totalistic rules in an attempt to destroy the remaining clusters and complete the job of global coordination. Barfoot and D'Eleuterio describe a subset of the results presented here in [12].

The essential idea used in the model described here has been borrowed from nonlinear physics. To force the system away from a uniform density of symbols (which may be viewed as a stable equilibrium), an instability is introduced in the local update rule of each decision maker. This local instability drives the system away from a uniform density and, as we will see, towards global consensus. The notion of an instability forcing a system far-from-equilibrium thus creating "order" has been seen in nonlinear physics [13] and chemistry [14]. At the single trajectory level, an instability may be seen as breaking symmetry between more than one alternative while at the ensemble level, symmetry is once again restored.

The use of instabilities in the coordination of artificial systems is not new. Haken [15–17] led a movement to describe all structure in nature in these terms and exploit these ideas in the design of artificial systems. Instabilities have even been used to describe coordination in the wavelength population dynamics of lasers [18].

3. Stochastic cellular automata

In deterministic CA there is an *alphabet* of K symbols, one of which may be adopted by each cell. Incoming connections each provide a cell with one of these symbols. The combination of all incoming symbols uniquely determines which symbol the cell will display as output. Stochastic cellular automata (SCA) work in the very same way except at the output level. Instead of there being a single unique symbol which is adopted with probability 1, there can be multiple symbols adopted with probability less than 1. Based on this *outgoing probability density* over the K symbols, a single unique symbol is drawn to be the output of the cell. This is done for all cells simultaneously. It should be noted that deterministic CA are a special case of SCA.

We consider a specific subcase of SCA in this paper which corresponds to the totalistic rules of CA. Assume that cells cannot tell which symbols came from which connections. In this case, it is only the intensity of each incoming symbol which becomes important. Furthermore, we desire that our rules work with any number of incoming connections. Thus, rather than using the number of each of the incoming K symbols, we use this number normalized by the number of connections, which can be thought of as an *incoming probability density*. In summary the model we consider is as follows.

Totalistic SCA. Consider a system of N cells, each of which is connected to a number of other cells. Let $\mathcal{A}$ represent an alphabet of K symbols, represented by the integers 1 through K. The *state* of cell i at time step t is $x_i[t] \in \mathcal{A}$. The *input probability density* $\boldsymbol{p}_{\text{in}}$ for cell i is given by

$$\boldsymbol{p}_{\text{in}}[t] = \boldsymbol{\sigma}_i\big(x_1[t], \ldots, x_N[t]\big) \tag{1}$$

where $\boldsymbol{\sigma}_i$ accounts for the connections of cell i to the other cells. More specifically, if we define c_{ij} to be 1 if cell i is connected to cell j, and 0 otherwise, then for cell i we have the input probability density given by

$$p_{k,\text{in}}[t] = \sum_{j=1}^{N} \frac{c_{ij}}{\sum_{m=1}^{N} c_{im}} \delta(k, x_j[t]) \tag{2}$$

where the Kronecker delta $\delta(k, x_j[t])$ is 1 when $x_j[t] = k$ and 0 otherwise. The *output probability density* $\boldsymbol{p}_{\text{out}}$ is given by the map $\boldsymbol{\varphi}$:

$$\boldsymbol{p}_{\text{out}}[t+1] = \boldsymbol{\varphi}\big(\boldsymbol{p}_{\text{in}}[t]\big). \tag{3}$$

The probability densities $\boldsymbol{p}_{\text{in}}$ and $\boldsymbol{p}_{\text{out}}$ are *stochastic columns*. The new state of cell i at time step $t+1$ is randomly drawn according to the density $\boldsymbol{p}_{\text{out}}[t+1]$ and is represented by $x_i[t+1]$.

It should be noted that in equation (1) if the connections between the cells are not changing over time then the functions $\boldsymbol{\sigma}_i(\cdot)$ will not be functions of time. However, we could allow these connections to change which would make them functions of time.

Once the connections are described through the $\boldsymbol{\sigma}_i(\cdot)$ functions, the only thing that remains to be defined is the $\boldsymbol{\varphi}$-map. We assume that each cell has the same $\boldsymbol{\varphi}$-map but this need not be the case. The possibilities for this map are infinite and thus for the remainder of this paper we discuss parameterized subsets of these possibilities. One subset will be called *piecewise-$\boldsymbol{\varphi}$* and is defined as follows.

Piecewise-φ. Let

$$\boldsymbol{p}_{\text{in}} = \big[\, p_{1,\text{in}} \; \cdots \; p_{K,\text{in}} \,\big]^T. \tag{4}$$

The (unnormalized) output probabilities are given by

$$p_{k,\text{out}} = \begin{cases} 1, & \text{if } \frac{1}{K} + \beta(p_{k,\text{in}} - \frac{1}{K}) \geq 1 \\ 0, & \text{if } \frac{1}{K} + \beta(p_{k,\text{in}} - \frac{1}{K}) \leq 0 \\ \frac{1}{K} + \beta(p_{k,\text{in}} - \frac{1}{K}), & \text{otherwise} \end{cases} \tag{5}$$

where β is derived from the tunable parameter λ as follows:

$$\beta = \begin{cases} 2\lambda, & \text{if } 0 \leq \lambda \leq \frac{1}{2} \\ \frac{1}{2}(1-\lambda)^{-1}, & \text{if } \frac{1}{2} \leq \lambda < 1. \end{cases} \tag{6}$$

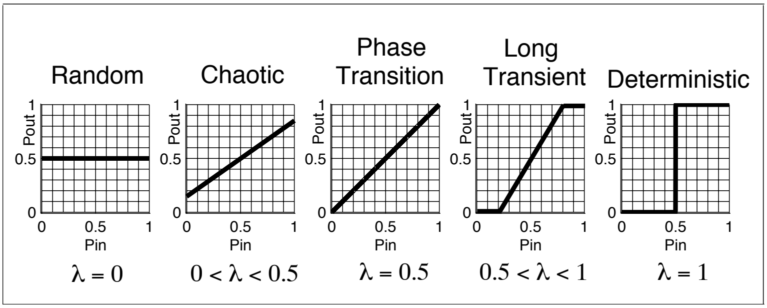

Figure 3. The piecewise-φ rule for different values of λ and $K = 2$. Note that in the diagrams $P_{\text{in}} = p_{1,\text{in}}$ since there is only one degree of freedom in $\mathbf{p}_{\text{in}}$ with $K = 2$ (by the theorem of total probability). Similarly, $P_{\text{out}} = p_{1,\text{out}}$.

The (normalized) output probability column is

$$\boldsymbol{p}_{\text{out}} = \frac{1}{p_{\Sigma,\text{out}}} \left[p_{1,\text{out}} \cdots p_{K,\text{out}} \right]^T \tag{7}$$

where $p_{\Sigma,\text{out}} = \sum_{k=1}^{K} p_{k,\text{out}}$.

Note that in equation (6), the tunable parameter, λ acts in a similar manner to a temperature parameter. When $\lambda = 1$ we have a completely deterministic rule while when $\lambda = 0$ we have a completely random rule. Figure 3 shows what the rule looks like for different λ when $K = 2$. Another possibility for the φ-map will be called *linear-φ* and is defined as follows.

Linear-φ. Let

$$\boldsymbol{p}_{\text{in}} = \left[p_{1,\text{in}} \cdots p_{K,\text{in}} \right]^T. \tag{8}$$

The (unnormalized) output probabilities are given by

$$p_{k,\text{out}} = p_{k,\text{in}}^{\beta} \tag{9}$$

where β is derived from the tunable parameter λ as follows:

$$\beta = \begin{cases} 2\lambda, & \text{if } 0 \leq \lambda \leq \frac{1}{2} \\ \frac{1}{2}(1-\lambda)^{-1}, & \text{if } \frac{1}{2} \leq \lambda < 1. \end{cases} \tag{10}$$

The (normalized) output probability column is

$$\boldsymbol{p}_{\text{out}} = \frac{1}{p_{\Sigma,\text{out}}} \left[p_{1,\text{out}} \cdots p_{K,\text{out}} \right]^T \tag{11}$$

where $p_{\Sigma,\text{out}} = \sum_{k=1}^{K} p_{k,\text{out}}$.

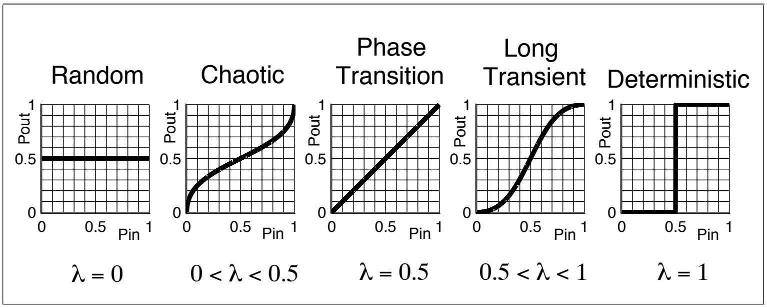

Figure 4. The linear-φ rule for different values of λ and $K = 2$. Note that in the diagrams $P_{\text{in}} = p_{1,\text{in}}$ since there is only one degree of freedom in $\mathbf{p}_{\text{in}}$ with $K = 2$ (by the theorem of total probability). Similarly, $P_{\text{out}} = p_{1,\text{out}}$.

The tunable parameter λ behaves as for the piecewise-φ map. Figure 4 shows what the rule looks like for different λ when $K = 2$. This rule has been called "linear" as there exists an algebra in which this map is simply scalar multiplication, but that topic is beyond the scope of this paper [19].

An equilibrium point $\boldsymbol{p}^*$ in a φ-map is one for which the following is true

$$\boldsymbol{p}^* = \varphi(\boldsymbol{p}^*). \tag{12}$$

The idea behind the piecewise-φ and linear-φ rules was to create an instability in the probability map at the uniform density equilibrium point

$$\boldsymbol{p}^*_{\text{uni}} = \left[\frac{1}{K} \; \frac{1}{K} \cdots \frac{1}{K} \right]^T \tag{13}$$

such that a small perturbation from this point would drive the probability towards one of the stable equilibria

$$\boldsymbol{p}^*_1 = [\, 1 \;\; 0 \; \ldots \; 0 \,]^T \tag{14}$$
$$\boldsymbol{p}^*_2 = [\, 0 \;\; 1 \; \ldots \; 0 \,]^T \tag{15}$$
$$\vdots$$
$$\boldsymbol{p}^*_K = [\, 0 \;\; 0 \; \ldots \; 1 \,]^T \tag{16}$$

It turns out that when $0 \leq \lambda < 1/2$, the equilibrium point $\boldsymbol{p}^*_{\text{uni}}$ is the only stable equilibrium. However when $1/2 < \lambda \leq 1$, $\boldsymbol{p}^*_{\text{uni}}$ becomes unstable and the other equilibria $\boldsymbol{p}^*_1, \ldots, \boldsymbol{p}^*_K$ become stable. This is similar to the classic pitchfork bifurcation as depicted in Figure 5 for $K = 2$. However, with K symbols in the alphabet the pitchfork will have K tines.

There certainly are other stochastic functions which may work as φ-maps but two will be enough for our purposes here. It is important to stress that we have designed the stability of our system at a local

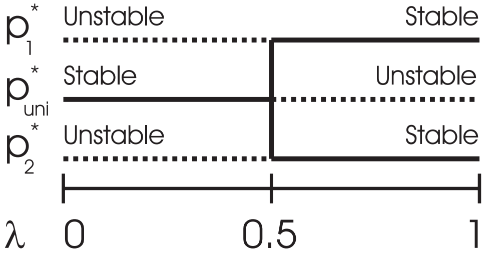

Figure 5. Pitchfork stability of the piecewise-φ rule for $K = 2$. λ is a parameter analogous to a temperature.

level. The question of global stability and success on the decentralized coordination problem does not follow directly from the local stability of each cell. As we show, it is possible to study the global stability of a large system of cells (e.g., $N = 100$) with a piecewise-φ or linear-φ rule analytically. However, there is an explosion in the number of global states as K is increased. For example, with $N = 100$ and $K = 2$ there are $2^{100} \approxeq 1.3 \times 10^{30}$ possible global states. The approach in section 4 is to study them through simulation and statistical analysis. This is an important issue for stochastic decentralized systems. If it is computationally intractable to study large systems analytically and prove they will work, then are they still useful? The hope is that by designing large decentralized systems from the bottom up, the interactions that we design on a small scale will still work on a very large scale. This is typically called *scaling up* and is investigated here through simulation.

4. Simulation

We now present simulations of cells employing the piecewise-φ rule. In order to ensure that the connections between cells are not regular, we consider each cell to exist in a cartesian box (of size 1 by 1). The N cells are randomly positioned in this box and symmetrical connections are formed between two cells if they are closer than a threshold euclidean distance d from one another. Figure 2 shows example connections between $N = 100$ cells with $d = 0.2$. Figure 6 shows example time series for different values of λ. When $\lambda < 0.5$ chaotic global behavior arises, with $0.5 < \lambda < 1$ fairly successful behavior results, but with $\lambda = 1$ clusters form. The formation of clusters means that the global system has stable equilibria which we did not predict from the local rule. However, as λ is decreased towards 0.5, these equilibria are no longer stable and the system continues to coordinate.

It would seem that there is a good correlation between the stability on the local level and the behavior type of the global system. As λ moves from below 0.5 to above, it appears there is a dramatic

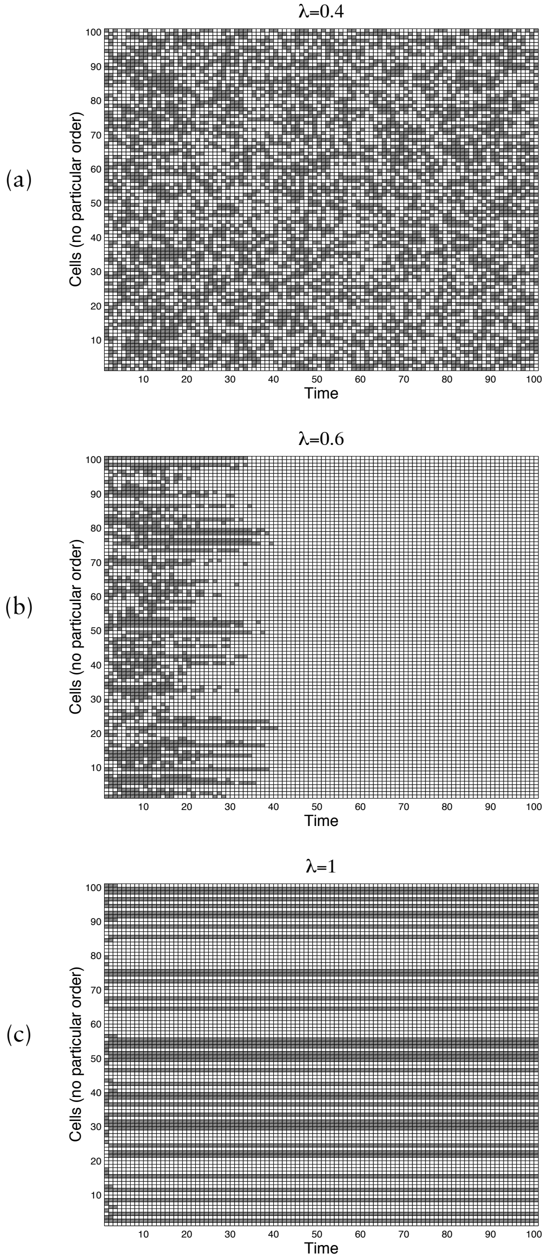

Figure 6. Example time series for different values of λ and $N = 100$, $K = 2$, and $d = 0.2$. (a) chaotic behavior, (b) successful coordination, and (c) clusters. The two colors represent the two symbols of the alphabet. The cells are listed vertically in no particular order. The piecewise-$\boldsymbol{\varphi}$ rule was used but plots are qualitatively the same for linear-$\boldsymbol{\varphi}$.

phase transition in the behavior of the system (totally chaotic to fixed point). In the neighborhood of 0.5 there is long transient behavior. It turns out that the best value for λ, from the point of view of multiagent coordination, is approximately $\lambda = 0.6$. Figures 17, 18, and 19 in the appendix show examples of SCA with random states, clusters, and consensus.

5. Statistical analysis

In an attempt to quantify the qualitative observations of section 4 a number of statistical measures were employed in the analysis of the SCA time series. These were used also by Langton in [6]. The first measure is taken from [20] and will be referred to as *entropy* (H). It is defined as follows.

Entropy. Given a sequence of M symbols

$$\mathbf{s} = \begin{bmatrix} s_1 & s_2 & \dots & s_M \end{bmatrix}^T \tag{17}$$

from an alphabet of size K, the *entropy* of the sequence may be computed as follows. First compute the frequency n_k of each of the K symbols $\forall k \in 1 \dots K$, which is simply the number of occurrences of symbol k in the sequence $\mathbf{s}$. From the frequencies, compute the probability p_k of each of the K symbols $\forall k \in 1 \dots K$ as

$$p_k = \frac{n_k}{n_\Sigma} \tag{18}$$

where $n_\Sigma = \sum_{i=1}^{K} n_i$. Finally, the entropy of sequence $H(\mathbf{s})$ is defined as

$$H(\mathbf{s}) = \frac{-\sum_{k=1}^{K} p_k \ln(p_k)}{\ln(K)} \tag{19}$$

where the $\ln(K)$ denominator is a normalization constant to make $H(\mathbf{s}) \in [0, 1]$.

This entropy function produces a value of 0 when all the symbols in $\mathbf{s}$ are identical and a value of 1 when all K symbols are equally common. The second measure is based on the first and will be referred to as *mutual information* (I). It is defined as follows.

Mutual information. Given two sequences of M symbols

$$\mathbf{s}_1 = \begin{bmatrix} s_{1,1} & s_{1,2} & \dots & s_{1,M} \end{bmatrix}^T \tag{20}$$

$$\mathbf{s}_2 = \begin{bmatrix} s_{2,1} & s_{2,2} & \dots & s_{2,M} \end{bmatrix}^T \tag{21}$$

from an alphabet of size K, the *mutual information* of the sequence $I(\mathbf{s}_1, \mathbf{s}_2)$ may be defined as

$$I(\mathbf{s}_1, \mathbf{s}_2) = H(\mathbf{s}_1) + H(\mathbf{s}_2) - H(\mathbf{s}_1, \mathbf{s}_2) \tag{22}$$

where $H(\mathbf{s}_1, \mathbf{s}_2)$ is the entropy of the two sequences considered as a joint process (i.e., with an alphabet of size $K \times K$).

The entropy and mutual information measures may be computed on any sequence of symbols. We tested them on *spatial* sequences (e.g., time series *columns* from Figure 6) and *temporal* sequences (e.g., time series *rows* from Figure 6). The most interesting measures were *average spatial entropy* (average of entropies computed from all columns in a time series) and *average temporal mutual information* (average of all Is computed from all rows in a time series: I was computed between a row and itself shifted by one time step).

Figures 7, 8, and 9 show various measures for piecewise-φ and 1000 values of λ. At each value of λ, 100 simulations were done on different random connections between cells and initial conditions. Thus, all displayed measures are actually averaged over 100 simulations. Each simulation was run for 300 time steps with $N = 100$, $K = 2$, and $d = 0.2$. Figures 10, 11, and 12 show the same measures for linear-φ.

Figures 7 and 10 show the average number of clusters at the final time step for different values of λ. The phase transition is quite obvious at $\lambda = 0.5$. The optimal value (in terms of the fewest clusters formed on average) for λ is near 0.6 for piecewise-φ and 0.53 for linear-φ. Figures 8 and 11 show average spatial entropy for different values of λ. This measure has a good correlation with average number of clusters. Again, there is a minimum occurring which corresponds to the best performance at multiagent coordination.

Figures 9 and 12 display average temporal mutual information for different values of λ. These are very interesting plots. Temporal mutual

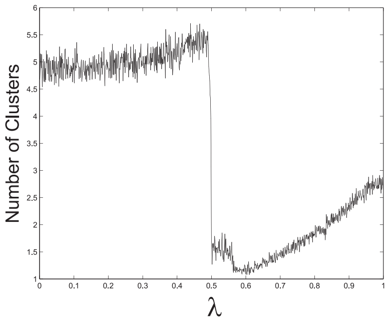

Figure 7. Average number of clusters at the final time step for 1000 values of λ. Plot shows average of 100 simulations at each value of λ. Number of clusters was computed by considering the SCA as a Markov chain with connections deleted between cells displaying different symbols. The number of clusters is then the number of eigenvalues equal to 1 from the Markov transition matrix. The piecewise-φ rule was used.

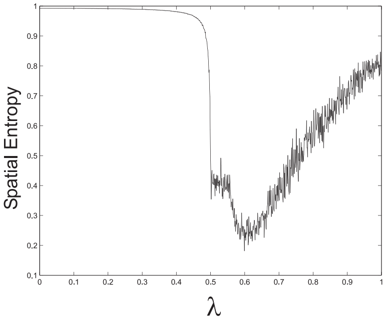

Figure 8. Average spatial entropy for 1000 values of λ. Plot shows average of 100 simulations at each value of λ. The piecewise-$\boldsymbol{\varphi}$ rule was used.

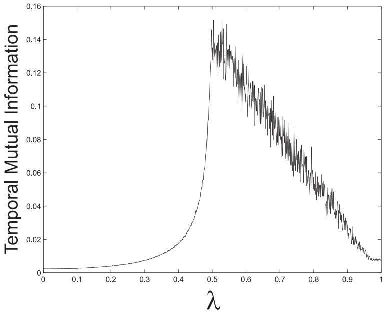

Figure 9. Average temporal mutual information for 1000 values of λ. Plot shows average of 100 simulations at each value of λ. The piecewise-$\boldsymbol{\varphi}$ rule was used.

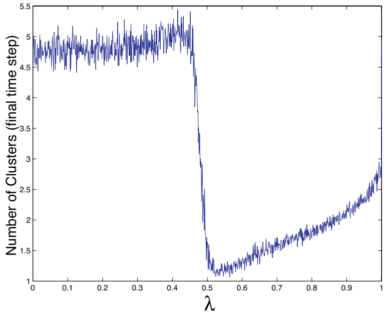

Figure 10. Average number of clusters at the final time step for 1000 values of λ. Plot shows average of 100 simulations at each value of λ. Number of clusters was computed by considering the SCA as a Markov chain with connections deleted between cells displaying different symbols. The number of clusters is then the number of eigenvalues equal to 1 from the Markov transition matrix. The linear-$\boldsymbol{\varphi}$ rule was used.

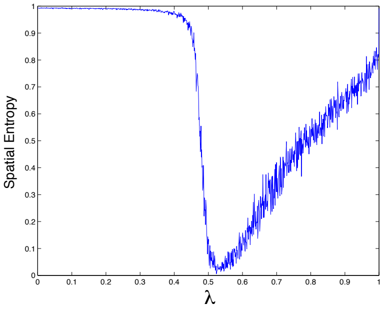

Figure 11. Average spatial entropy for 1000 values of λ. Plot shows average of 100 simulations at each value of λ. The linear-φ rule was used.

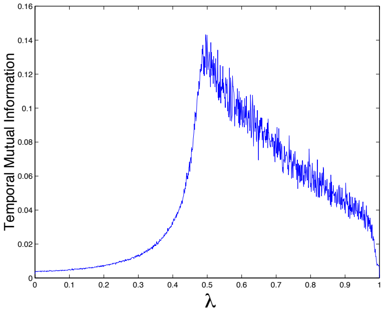

Figure 12. Average temporal mutual information for 1000 values of λ. Plot shows average of 100 simulations at each value of λ. The linear-φ rule was used.

information seems to capture the length of the global transient behavior of the system. As discussed by Langton in [6], the random pattern in the chaotic region is not considered transient but rather the steady-state behavior. The peak in temporal mutual information occurs at $\lambda = 0.5$, the phase transition, and drops away on either side (for both rules). Langton has a similar plot in [6].

Figure 13 shows how the average number of clusters at the final time step varies as the problem is scaled up from $N = 100$ cells to $N = 1000$ cells. The plot shows an average of 100 simulations, each run for 300 time steps with $K = 2$, $\lambda = 0.6$, and $d = 2/\sqrt{N}$. The parameter d was made to depend on the number of cells in order to keep the average density of connections the same. This was required as the cartesian box in which the cells live was always of size 1 by 1. As more cells are added they are closer together and thus to keep the density of connections between cells constant (on average), the factor of $2/\sqrt{N}$ was needed. The resulting relationship between number of clusters and number of

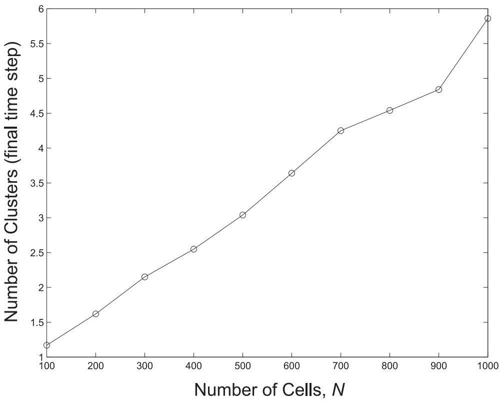

Figure 13. Average number of clusters (at the final time step) as the number of cells N is varied from 100 to 1000. The parameters were: 300 time steps, $K = 2$, $\lambda = 0.6$, and $d = 2/\sqrt{N}$. The piecewise-φ rule was used. Plot shows average from 100 simulations at each value of N.

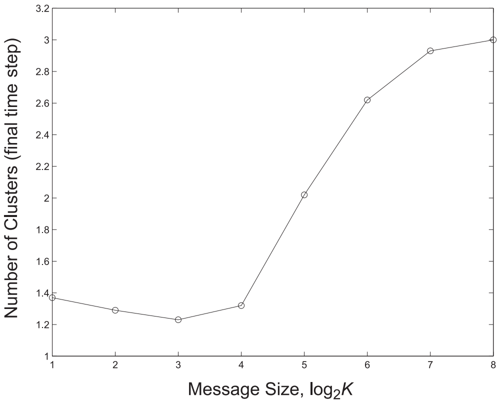

Figure 14. Average number of clusters (at the final time step) as the alphabet size K is varied from 2 (1 bit) to 256 (8 bits). The parameters were: 300 time steps, $N = 100$, $\lambda = 0.6$, and $d = 0.2$. The piecewise-φ rule was used. Plot shows average from 100 simulations at each value of K.

cells is quite linear, about 0.47 clusters for every 100 cells added. Barfoot and D'Eleuterio show a qualitatively similar scaling plot for a heap formation problem in [21].

Figure 14 shows how the average number of clusters, again at the final time step, varies as the problem is scaled up from an alphabet size K of 2 (a single bit) to 256 (8 bits or 1 byte). The plot shows an average of 100 simulations, each run for 300 time steps with $N = 100$, $\lambda = 0.6$, and $d = 0.2$.

6. Markov chain

Another approach we have employed in the analyses of SCA is to construct a Markov chain [22] for the global state of the system and look at its structure. As we will show, it is possible to predict some of the behavior seen in the simulations by examining the eigenvalues of the Markov transition matrix. We desire a Markov chain of the form

$$z[t+1] = \boldsymbol{A}z[t] \tag{23}$$

where $z = [z_i]$ is the joint probability density for all the cells and $\boldsymbol{A} = [a_{ij}]$ is the transition matrix. The joint density will be a column of size $K^N \times 1$ whose entries are probabilities which sum to 1. The transition matrix will be of size $K^N \times K^N$ whose entries are probabilities and whose columns sum to 1. The transition probabilities (using the linear-$\boldsymbol{\varphi}$ rule) may then be written as

$$a_{ij} = \frac{\alpha_{ij}^{\beta}}{\sum_{k=1}^{K^N} \alpha_{kj}^{\beta}} \tag{24}$$

$$\alpha_{ij} = \prod_{k=1}^{N} \left(\sum_{m=1}^{K} p_{k,mi} \sum_{l=1}^{N} \hat{c}_{kl} p_{l,mj} \right) \tag{25}$$

$$p_{k,ij} = \delta\left(i, \left\lfloor \frac{j-1}{K^{k-1}} \right\rfloor \bmod K + 1\right) \tag{26}$$

$$\hat{c}_{kl} = \frac{c_{kl}}{\sum_{m=1}^{N} c_{km}} \tag{27}$$

where $\delta(\cdot,\cdot)$ is the Kronecker delta, $\lfloor\cdot\rfloor$ rounds down to the nearest integer, and mod$(\cdot)$ refers to modular arithmetic. The cell connection information is captured in c_{kl}, which is 1 if there is a connection between cells k and l, otherwise 0. Although the order of the global states will not be needed for our analysis, the joint density over the global states is given by

$$z_i[t] = \frac{\zeta_i[t]}{\sum_{j=1}^{K^N} \zeta_j[t]} \tag{28}$$

$$\zeta_i[t] = \prod_{k=1}^{N} \left(\sum_{j=1}^{K} p_{k,ji} \delta\left(j, x_k[t]\right) \right). \tag{29}$$

Since we now have a Markov chain, the following properties about the eigenvalues and eigenvectors of the stochastic matrix $\boldsymbol{A}$ will be useful [23].

(a) All the eigenvalues of $\boldsymbol{A}$ are less than or equal to 1 in magnitude.

(b) There is always at least one eigenvalue equal to 1.

(c) The number of ergodic sets (attractors) will be equal to the number of eigenvalues which are exactly 1.

Using these properties it is not difficult to show, for a specific set of connections, there are exactly K eigenvalues exactly equal to 1 which correspond to the K global consensus states (all cells the same). This implies that for any value of β, if we happen to land in one of the consensus states, we will stay there. It does not, however, predict the phase transition phenomenon we have seen when β becomes just bigger than 1 (λ just bigger than 1/2).

To predict this kind of behavior, we need to look at some of the other eigenvalues to see how fast the other modes in the system will die off. It turns out the next biggest eigenvalue, after eliminating the K eigenvalues equal to 1, is a good predictor for the behavior we have seen. Figure 15 shows how this eigenvalue behaves for some SCA. The SCA used were a ring of N cells with each cell connected to itself and two neighbors on each side. N was varied from six to 10 cells. We can see that this eigenvalue dips down between $\lambda = 1/2$ and $\lambda = 1$ and the minimum is approaching the phase transition we identified earlier as the number of cells is increased. The closer this eigenvalue is to 1, the longer it will take to reach consensus since this mode will be slow to die out.

We would have liked to increase N beyond 10 but, unfortunately, it is quite computationally expensive to generate these plots. For example, when $N = 10$ and $K = 2$, our Markov transition matrix is 1024 by 1024 in size which means there are over a million elements. The step represented by equation (24) is very computationally expensive as it involves over a million exponentiations for every value of β in which we are interested. We tried increasing N to 11 but unfortunately ran into computer system memory issues.

We also tried varying the sparseness of the connections by using SCA with $N = 10$ cells which were connected to themselves and M neighbors on each side. The value of M was varied from 1 to 4. With $M = 1$ we had minimal connections and with $M = 4$ we had maximal connections (without being fully connected) as each cell was connected to all other cells except for one. Figure 16 shows how the next largest eigenvalue changes as β (or λ) is varied for the different values of M. Clearly, with more connections, this eigenvalue is smaller which is expected as this implies consensus will be reached more quickly since this mode dies out faster. Also of note is that the location of the minimum moves to the right as the number of connections is increased. This also makes sense, for in the limit of a fully connected network, we expect the minimum to be at infinite β or $\lambda = 1$ as deterministic voting will certainly be the quickest way to reach consensus.

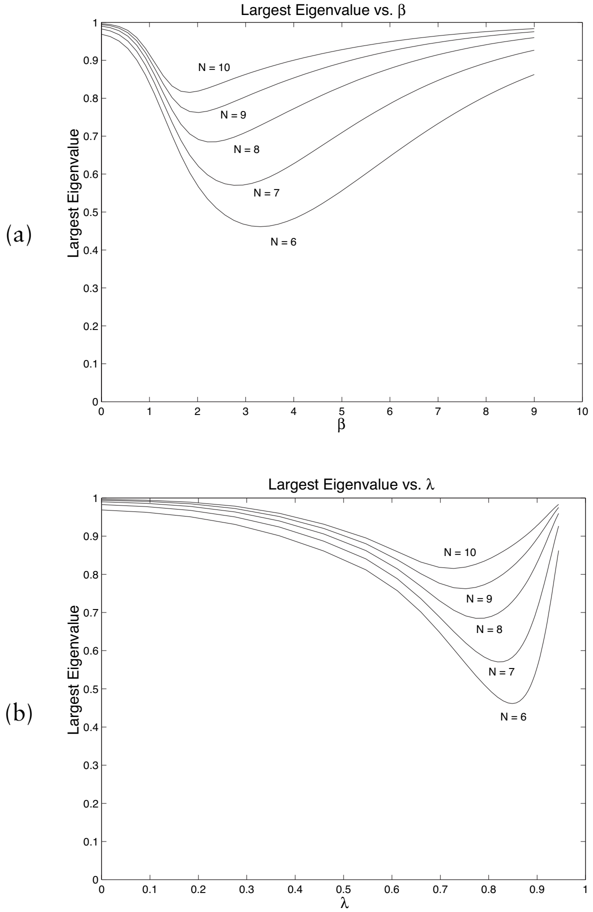

Figure 15. Next biggest eigenvalue of a Markov chain for the global dynamics of SCA, after eliminating the K eigenvalues equal to 1 which correspond to consensus. The SCA used were a ring of N cells with each cell connected to itself and two neighbors on each side. N was varied from six to 10 cells. (a) Eigenvalue *versus* β. (b) Eigenvalue *versus* λ.

7. Discussion

The strong correlation between the local stability of the piecewise-$\boldsymbol{\varphi}$ and linear-$\boldsymbol{\varphi}$ rules and the type of global behavior is quite interesting. It appears that $\lambda > 0.5$ corresponds to fixed point behavior (Wolfram's Class I), $\lambda < 0.5$ corresponds to chaotic behavior (Wolfram's Class III), and λ near 0.5 corresponds to long transient behavior (Wolfram's Class IV). Local correlation has to do with the way in which the incoming probability density is computed in equation (1). This step

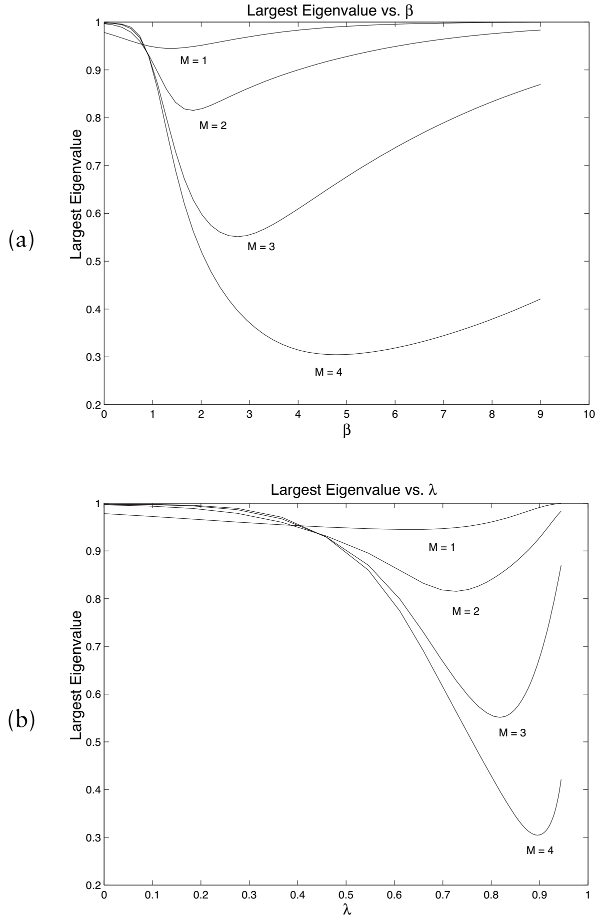

Figure 16. Next biggest eigenvalue of a Markov chain for the global dynamics of SCA, after eliminating the K eigenvalues equal to 1 which correspond to consensus. The SCA used were a ring of 10 cells with each cell connected to itself and M neighbors on each side. M was varied from one to four cells. (a) Eigenvalue *versus* β. (b) Eigenvalue *versus* λ.

delivers information averaged from all connected cells. This averaging serves to smooth out differences between connected cells. However, if this smoothing occurs too quickly (i.e., $\lambda = 1$) the system does not have time to smooth globally, resulting in the formation of clusters. This has been called *critical slowing down* [16] in other systems. As we approach the critical point ($\lambda = 0.5$ or $\beta = 1$) from above, the strength of the instability decreases, which slows down the decision-making process. The third and vital ingredient in the recipe for self-organization is the fluctuations that occur at the single trajectory level. These fluctuations allow the system to begin the process of moving away from the unstable equilibrium $\boldsymbol{p}_{\mathrm{uni}}$ in the $\boldsymbol{\varphi}$-maps. The nature of the $\boldsymbol{\varphi}$-maps is such that these fluctuations are largest when the system is near $\boldsymbol{p}_{\mathrm{uni}}$ and become smaller and smaller as a cell becomes more coordinated with its neighbors. It is a balance of these three effects which seems to be the most effective at decentralized coordination. To summarize, self-organization in this model requires the following three mechanisms.

- Instability in the $\boldsymbol{\varphi}$-map which forces each cell to move away from $\boldsymbol{p}_{\mathrm{uni}}$ (behaving randomly) and towards one of a number of deterministic decisions.
- Averaging in the $\boldsymbol{\sigma}$-map which serves to bias each cell to conform to the average behavior of its immediate (connected) neighbors.
- Fluctuations at the single trajectory level to cause each cell to move away from the unstable equilibrium $\boldsymbol{p}_{\mathrm{uni}}$. These fluctuations become smaller as the cell moves further away.

To properly balance these three effects the parameter λ was tuned. The optimal operating value of λ is not right at the phase transition but a little bit towards the deterministic end of the λ spectrum (approximately $\lambda = 0.6$ for piecewise-$\boldsymbol{\varphi}$ and 0.53 for linear-$\boldsymbol{\varphi}$).

Note that we did not find any oscillatory behavior (Wolfram's Class II) which is because the connections between the cells are symmetrical. However, if the piecewise-$\boldsymbol{\varphi}$ rule is reflected (left-right) then the system "blinks" and global coordination corresponds to all cells blinking in phase with one another. The same may be said for the linear-$\boldsymbol{\varphi}$ rule.

What is happening in the SCA model is that the boundaries between clusters are made unstable. This forces them to move randomly until they contact one another and annihilate, leaving a single cluster. This annihilation of boundaries is qualitatively the same method found to work in deterministic CA by Mitchell *et al.* [8] and Das *et al.* [9]. In those studies the boundaries were made to move in very specific ways by exploiting the nature of the connections between cells. They found that the boundaries could be made to travel long distances. This allowed coordination to occur more quickly than the method presented here. However, their mechanism was not immediately portable to different connective architectures. By not exploiting the underlying connections between cells, the best we can do is to make the boundaries move randomly and wait for them to contact one another and annihilate. The benefit is that this method is independent of the connective architecture.

The simulation results presented here used $N = 100$ cells and required on average 150 time steps to get to a single cluster with $d = 0.2$, $K = 2$, and $\lambda = 0.6$. Clearly, the time required to form a single cluster will increase with the number of cells in the system. This is reflected in Figure 13 which shows how the number of clusters after 300 time steps varies as the number of cells is increased. The larger systems are not able to finish coordinating in the allowed time, thus resulting in more clusters.

Figure 14 shows how the number of clusters at the final time step varies with the alphabet size K. This is more difficult to explain as the curve first goes down a little and then up as K is increased from 2 to 256 in factors of 2. This would have been difficult to predict analytically. In some ways the $K = 2$ case is very difficult as there can be two equally large clusters whose boundary fluctuates but is never annihilated to leave a single cluster. There is effectively a stalemate, no further progress is being made. Having more clusters that are smaller in size can make the fluctuations relatively bigger, enabling boundaries to be annihilated more quickly. This explains the initial decline in Figure 14. The eventual rise in the plot (increasing from $K = 16$) is similar to that in Figure 13. Although the system is still making progress, it is not able to complete coordination in 300 time steps. As the alphabet size becomes larger than the number of cells (here 100), the plot levels off. This may be explained by the fact that N cells cannot represent more than N different symbols in the random initial condition, regardless of how large we make the alphabet size K. Note that if the system becomes too inefficient at very large K (i.e., too time consuming), it is possible to use more than one coordination mechanism and combine the results. For example, two of the $K = 8$ mechanisms could be combined to produce messages of size 64. Depending on the parameters this may or may not improve coordination efficiency. The last point which should be mentioned is that the same value of $\lambda = 0.6$ was used at all values of K. There could, however, be different optimal values for this parameter for each K.

The piecewise-$\boldsymbol{\varphi}$ and linear-$\boldsymbol{\varphi}$ rules are not the only maps that can be used to achieve decentralized coordination in SCA. Replacing it with other monotonically increasing functions (i.e., in Figure 3) with the same equilibria will likely work. This was tried for $K = 2$ with the relation

$$p_{k,\text{out}} = 3(p_{k,\text{in}})^2 - 2(p_{k,\text{in}})^3 \qquad \forall k = 1 \ldots K \tag{30}$$

which has equilibria at $[\,1\ 0\,]^T$, $[\,0\ 1\,]^T$, and $[\,1/2\ 1/2\,]^T$. It is fairly successful but the piecewise-$\boldsymbol{\varphi}$ and linear-$\boldsymbol{\varphi}$ rules are easier to parameterize. Equation (30) is similar in form to equations studied by Haken *et al.* [15, 17], particularly the cubic term.

The equilibria are the most important features to consider in the design of $\boldsymbol{\varphi}$-maps. Creating an instability at the uniform density $\boldsymbol{p}_{\text{uni}}$ is crucial. However, there are other features which can be incorporated. For example, the linear-$\boldsymbol{\varphi}$ rule is appealing as it provides a smooth route to a deterministic decision. Essentially, less and less noise is added as the incoming probability gets closer to one of the deterministic equilibria. The piecewise-$\boldsymbol{\varphi}$ is different in that it has regions of saturation for values of λ greater than 1/2. In the saturated regions of the curve (i.e., the horizontal flat parts) normal voting results. This feature greatly increases the strength of the stability of the deterministic equilibria. Thus any perturbation to the system is less likely to drive the system away from one of these regions. For example, if a group of 100 cells were already coordinated and a few more cells were introduced into the system at a later time, this would increase the likelihood of the new cells to conform. Similarly, if one cell were to begin malfunctioning it would be less likely to cause the other cells to become uncoordinated. The size of the saturated region may be tuned as circumstances require.

Finding the optimal value of λ for a particular set of parameters may not actually be necessary. As a future direction of research, a "cooling schedule" could be developed. We could start with λ near the phase transition (e.g., λ just larger than 0.5) and then slowly "cool" the system by bringing λ gradually towards 1. The system would certainly pass through the optimal value for λ. This form of cooling schedule has been used, for example, in simulated annealing, a global optimization method [24]. This would require each cell having some form of internal clock in order to time the cooling. Another possibility is to allow each cell to program its own λ using feedback. λ would get larger in periods of inactivity and smaller in periods of high activity. Removing the need for a centralized designer to program λ is one more step towards fully autonomous decentralized decision making. Another future direction of work is to consider the addition of noise to the communication between cells. It is likely that a small amount of communication noise will not cause the system to catastrophically stop working as it has been built on fluctuation and noise to begin with.

The model considered here does not require knowledge of the underlying structure of the connections between cells. This was a design requirement as it was originally motivated by a network of communicating mobile robots whose connections might be changing over time and thus difficult to exploit. It is thus natural to question whether the model still works as the connections are varied over time. To this end, a small amount of gaussian noise was added to the positions of the cells in the cartesian box at each time step. As the cells moved, the connections between them changed (since they are limited by the range d). The SCA model was still able to form single clusters. This was possible even when $\lambda = 1$, which makes sense since there is still some noise being added. However, the nature of the noise is at the connection level rather than the signal level. Over a long period of time it is as though the system were fully connected. However, the assumption of completely random movement is probably not a good one for a system of mobile robots. The coordination mechanism described here has tried to make as few assumptions as possible about the nature of connections between cells.

8. Conclusion

A mechanism for decentralized coordination has been presented based on stochastic cellular automata (SCA). This is an example of self-organizing behavior in that global coordination occurs in the face of more than one alternative. It was shown that by using stochastic rules, sparsely communicating agents can come to a global consensus. A common piece of information may be generated to which each cell has access using a stochastic approach. A parameter in the coordination mechanism was tuned and it was found that coordination occurred best when the system was near a phase transition between chaotic and ordered behavior (the optimum was a little bit towards the ordered side). It is hoped that this model will shed light on self-organization as a general concept while at the same time providing a simple algorithm to be used in practice.

Appendix

A. Stochastic cellular automata examples

Figures 17, 18, and 19 show examples of stochastic cellular automata with random initial conditions, clusters, and consensus, respectively.

References

[1] Erick Hoyt, *The Earth Dwellers: Adventures in the Land of Ants* (Simon and Schuster, New York, 1996).

[2] T. D. Barfoot, E. J. P. Earon, and G. M. T. D'Eleuterio, "A New Breed: Development of a Network of Mobile Robots for Space Exploration," in *Proceedings of the Sixth International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*, Montréal, Canada, June 19–21 2001.

[3] Jon von Neumann, *Theory of Self-Reproducing Automata* (University of Illinois Press, Urbana and London, 1966).

[4] Stephen Wolfram, "Universality and Complexity in Cellular Automata," *Physica D*, **10** (1984) 1–35.

[5] Stephen Wolfram, *A New Kind of Science* (Wolfram Media, Inc., 2002).

[6] Chris G. Langton, "Computations at the Edge of Chaos: Phase Transitions and Emergent Computation," *Physica D*, **42** (1990) 12–37.

[7] Chris G. Langton, "Life at the Edge of Chaos," in *Artificial Life II: Proceedings of the Workshop on Artificial Life*, edited by Chris G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, Santa Fe, NM, February 1990.

[8] Melanie Mitchell, Peter T. Hraber, and James P. Crutchfield, "Revisiting the Edge of Chaos: Evolving Cellular Automata to Perform Computations," *Complex Systems*, 7 (1993) 89–130; SFI Working Paper 93-03-014.

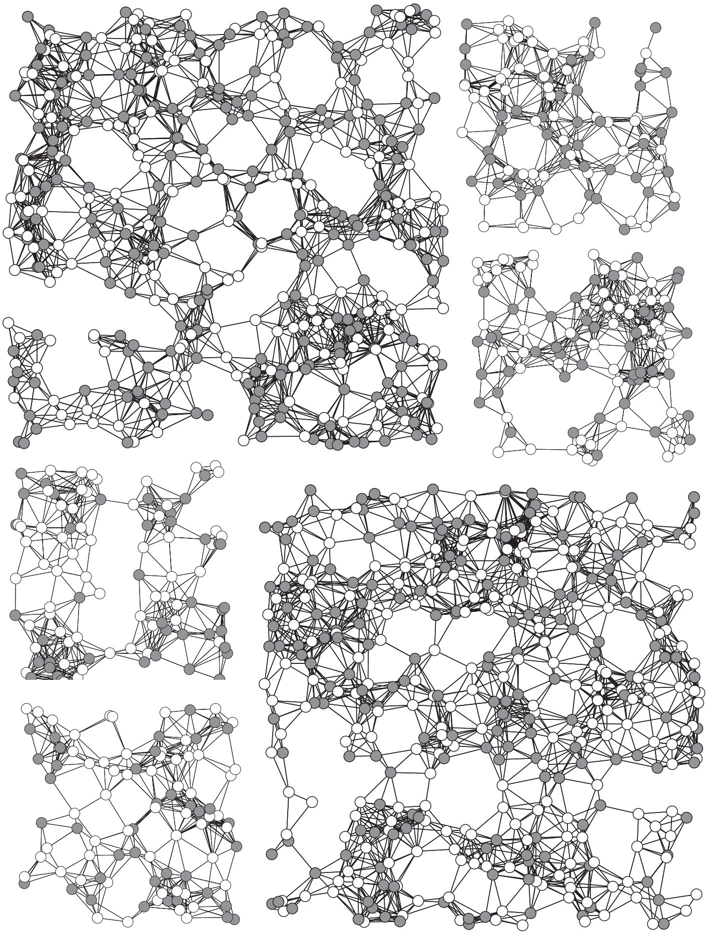

Figure 17. Six examples of random initial conditions for alphabet size $K = 2$. The two colors represent the two symbols of the alphabet. The larger examples have $N = 400$ and $d = 0.1$ while the smaller ones have $N = 100$ and $d = 0.2$.

[9] Rajarshi Das, James P. Crutchfield, Melanie Mitchell, and James E. Hanson, "Evolving Globally Synchronized Cellular Automata," in *Proceedings of the Sixth International Conference on Genetic Algorithms*, edited by L. J. Eshelman (Morgan Kaufmann Publishers, San Fransisco, CA, 1995).

[10] David Andre, Forrest H. Bennett, and John R. Koza, "Evolution of Intricate Long-distance Communication Signals in Cellular Automata using Genetic Programming," in *Artificial Life V, Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living*

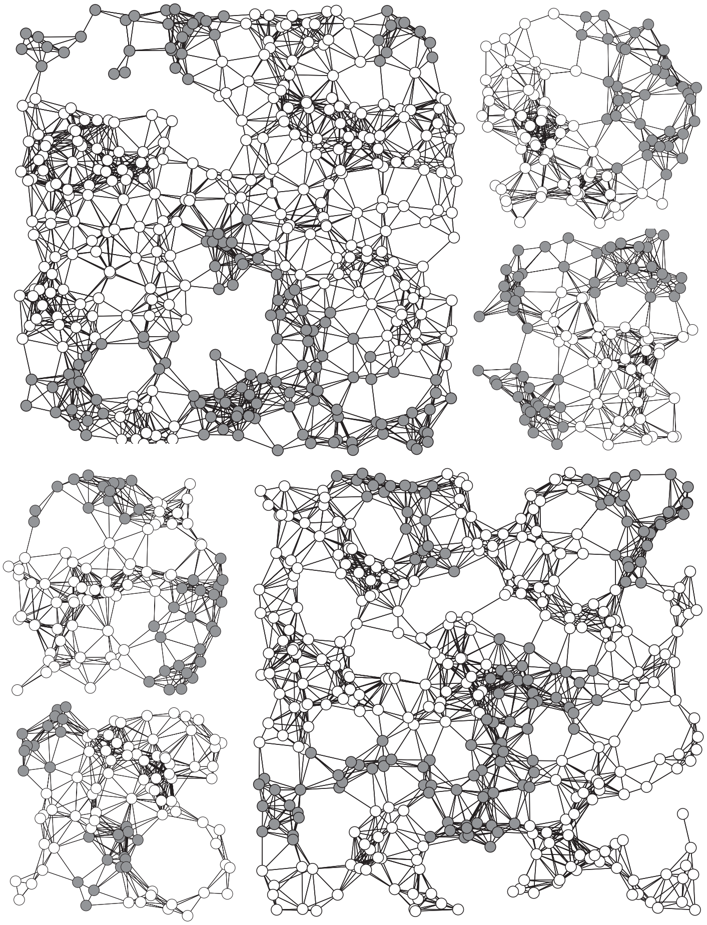

Figure 18. Six examples of undesirable clusters forming for alphabet size $K = 2$. The two colors represent the two symbols of the alphabet. The larger examples have $N = 400$ and $d = 0.1$ while the smaller ones have $N = 100$ and $d = 0.2$.

Systems, edited by Chris G. Langton and Katsunori Shimohara, May 16–18, 1996, Nara, Japan.

[11] Mieko Tanaka-Yamawaki, Sachiko Kitamikado, and Toshio Fukuda, "Consensus Formation and the Cellular Automata," *Robotics and Autonomous Systems*, **19** (1996) 15–22.

[12] T. D. Barfoot and G. M. T. D'Eleuterio, "Multiagent Coordination by Stochastic Cellular Automata," in *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, edited by Bernhard Nebel, Seattle, USA, August 4–10 2001 (Morgan Kaufmann, 2001).

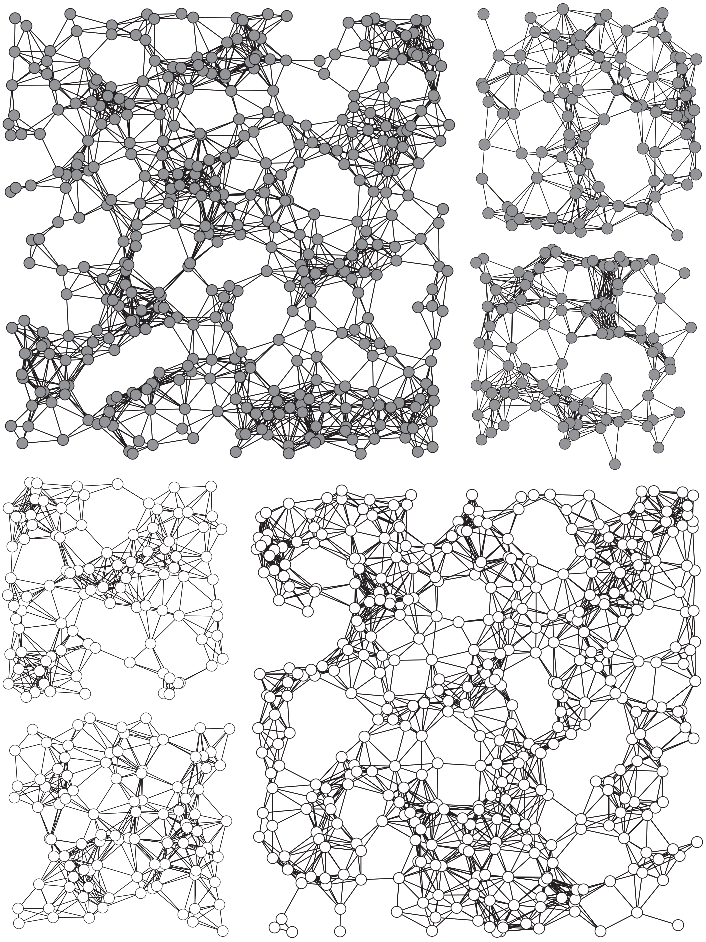

Figure 19. Six examples of consensus for alphabet size $K = 2$. The two colors represent the two symbols of the alphabet. The larger examples have $N = 400$ and $d = 0.1$ while the smaller ones have $N = 100$ and $d = 0.2$.

[13] G. Nicolis and I. Prigogine, *Self-Organization in Non-Equilibrium Systems* (John-Wiley and Sons, Inc., New York, 1977).

[14] G. Nicolis and F. Baras, *Chemical Instabilities* (D. Reidel Publishing Company, Dordrecht, Holland, 1984).

[15] H. Haken and M. Wagner, *Cooperative Phenomena* (Springer-Verlag, Berlin, 1973).

[16] H. Haken, *Synergetics, An Introduction, Third Edition* (Springer-Verlag, Berlin, 1983).

[17] H. Haken, *The Science of Structure: Synergetics* (Van Nostrand Reinhold Company, New York, 1984).

[18] R. Graham, A. Wunderlin, and H. Haken, *Lasers and Synergetics: A Colloquium on Coherence and Self-Organization in Nature* (Springer-Verlag, Berlin/New York, 1987).

[19] T. D. Barfoot, "Stochastic Decentralized Systems," Ph.D. Thesis, Institute for Aerospace Studies, University of Toronto, 2002.

[20] C. E. Shannon, "A Mathematical Theory of Communication," *The Bell System Technical Journal*, **27** (July 1948) 379–423.

[21] T. D. Barfoot and G. M. T. D'Eleuterio, "An Evolutionary Approach to Multiagent Heap Formation," in *Proceedings of the Congress on Evolutionary Computation*, edited by J. P. Angeline *et al.*, Washington DC, USA, July 6–9 1999.

[22] Andrei Andreevich Markov, "Essai d'une recherche statistique sur le texte du roman 'eugene onegin' illustrant la liaison des epreuve en chain," *Izvestia Imperatorskoi Akademii Nauk (Bulletin de l'Academie Imperiale des Sciences de St-Petersbourg)*, **7** (1913) 153–162.

[23] J. G. Kemeny and J. L. Snell, *Finite Markov Chains* (Springer, New York, 1976).

[24] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, **220**(4598) (1983) 671–680.