

Multiagent Coordination by Stochastic Cellular Automata

T D Barfoot and G M T D'Eleuterio

tim.barfoot@utoronto.ca, gde@utias.utoronto.ca

University of Toronto Institute for Aerospace Studies

4925 Dufferin Street, Toronto, Ontario, Canada, M3H 5T6

Abstract

A coordination mechanism for a system of sparsely communicating agents is described. The mechanism is based on a stochastic version of cellular automata. A parameter similar to a temperature can be tuned to change the behaviour of the system. It is found that the best coordination occurs near a phase transition between order and chaos. Coordination does not rely on any particular structure of the connections between agents, thus it may be applicable to a large array of sparsely communicating mobile robots.

1 Introduction

The term *multiagent system* encompasses large bodies of work from engineering, computer science, and mathematics. Examples include networks of mobile robots [Mataric, 1992], software agents [Bonabeau *et al.*, 1994], and cellular automata [Wolfram, 1984]. 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 behaviour 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 behaviour. It is possible for millions of ants to act as a *super-organism* through local pheromone communication. We seek to reproduce this ability on a fundamental level in order to coordinate artificial systems.

It can be argued that *cellular automata* (CA) are the simplest example of a multiagent system. Originally studied by [von Neumann, 1966], the term CA is used to describe systems of sparsely coupled difference equations. Despite their simple mechanics, some extremely interesting behaviours have been catalogued (e.g., Conway's *Game of Life*). 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. The goal will be to have all cells choose the same symbol from a number of possibilities using only sparse communication. 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.

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. Statistical analysis of the rule are provided followed by discussions and conclusions.

2 Related Work

In the following note that typically cellular automata do not operate in a stochastic but rather a deterministic manner. Unless explicitly stated (e.g., *stochastic cellular automata* (SCA)), the term cellular automata will imply determinism.

[von Neumann, 1966] originally studied cellular automata 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.

Cellular automata were categorized by the work of [Wolfram, 1984] in which four *universality classes* were identified. All rules were shown to belong to one of class I (fixed point), class II (oscillatory), class III (chaotic), or class IV (long transient). These universality classes can also be identified in SCA and we will show that in our particular model, choosing a parameter such that the system displays long transient behaviour (e.g., class IV) results in the best performance on our multiagent coordination task.

[Langton, 1990] 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 behaviour appeared in the vicinity of the phase transition. The current work is very comparable to this study in

that we also have a parameter which can be tuned to produce different CA rules. However, our parameter tunes the amount of randomness that is incorporated into the system. At one end of the spectrum, completely random behaviour ensues while at the other completely deterministic behaviour ensues. We also relate the universality classes to particular ranges of our parameter and find a correlation between performance on our multiagent coordination task and class IV behaviour. We attempt to use similar statistical measures to [Langton, 1990] to quantify our findings.

[Mitchell *et al.*, 1993], [Das *et al.*, 1995] study the same coordination task as will be examined here in the case of deterministic CA. However, their approach is to use a genetic algorithm to evolve rules successful at the task whereas here hand-coded rules are described. They found that the best solutions were able to send long range *particles* (similar to those in the *Game of Life*) [Andre *et al.*, 1997] in order to achieve coordination. These particles rely on the underlying structure of the connections between cells, specifically that each cell is connected to its neighbours 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.*, 1996] studies the same problem to that considered here. They use *totalistic* [Wolfram, 1984] rules which do not permit exploitation of the underlying structure of the connections between cells but rather rely on the intensity of each incoming symbol. They also vary a parameter to produce different rules and find that above a certain threshold, “global consensus” occurs but below it does not. However, they consider large clusters of symbols to be a successful global consensus. We do not and thus turn to a stochastic version of their totalistic rules to destroy these clusters and complete the job of global coordination.

3 The Model

In deterministic cellular automata 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 distribution* 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 sub-case 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 sym-

bols, we use this number normalized by the number of connections which can be thought of as an *incoming probability distribution*. 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. The state of Cell i at time-step t is $x_i[t] \in \mathcal{A}$. The input probability distribution, $\mathbf{p}_{\text{in}}$, for Cell i is given by

$$\mathbf{p}_{\text{in}}[t] = \sigma_i(x_1[t], \dots, x_N[t]) \quad (1)$$

where σ_i accounts for the connections of Cell i to the other cells. The output probability distribution $\mathbf{p}_{\text{out}}$ is given by the map, φ ,

$$\mathbf{p}_{\text{out}}[t+1] = \varphi(\mathbf{p}_{\text{in}}[t]) \quad (2)$$

The probability distributions $\mathbf{p}_{\text{in}}$ and $\mathbf{p}_{\text{out}}$ are stochastic columns. The new state of Cell i at time-step $t+1$ is randomly drawn according to the distribution $\mathbf{p}_{\text{out}}[t+1]$ and is represented by $x_i[t+1]$.

It should be noted that in (1) if the connections between the cells are not changing over time then the functions, $\sigma_i(\cdot)$, 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 $\sigma_i(\cdot)$ functions, the only thing that remains to be defined is the φ -map. We assume that each cell has the same φ -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 a parameterized subset of these possibilities. This subset will be called *piecewise- φ* and is defined as follows.

Piecewise- φ . Let

$$\mathbf{p}_{\text{in}} = [p_{1,\text{in}} \cdots p_{K,\text{in}}]^T \quad (3)$$

The (unnormalized) output probabilities are given by

$$p_{k,\text{out}} = \begin{cases} 1, & \text{if } \frac{1}{K} + \beta(p_{k,\text{in}} - \frac{1}{K}) \geq 1 \\ 0, & \text{if } \frac{1}{K} + \beta(p_{k,\text{in}} - \frac{1}{K}) \leq 0 \\ \frac{1}{K} + \beta(p_{k,\text{in}} - \frac{1}{K}), & \text{otherwise} \end{cases} \quad (4)$$

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

$$\beta = \begin{cases} 2\lambda, & \text{if } 0 \leq \lambda \leq \frac{1}{2} \\ \frac{1}{2}(1 - \lambda)^{-1}, & \text{if } \frac{1}{2} \leq \lambda < 1 \end{cases} \quad (5)$$

The (normalized) output probability column is

$$\mathbf{p}_{\text{out}} = \frac{1}{p_{\Sigma,\text{out}}} [p_{1,\text{out}} \cdots p_{K,\text{out}}]^T \quad (6)$$

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

Note that in (5), 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 1 shows what the rule looks like for different λ when $K = 2$.

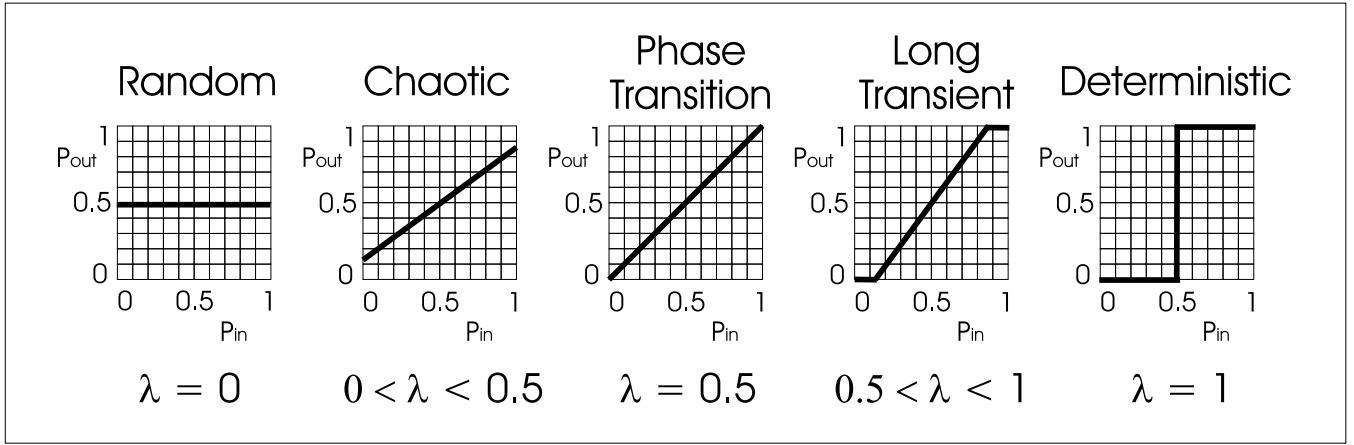


Figure 1: The *piecewise-φ* rule for different values of λ and $K = 2$.

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

$$\mathbf{p}^* = \varphi(\mathbf{p}^*) \quad (7)$$

The idea behind the *piecewise-φ* rule was to create an instability in the probability map at the uniform distribution equilibrium point

$$\mathbf{p}_{\text{uni}}^* = \left[\frac{1}{K} \quad \frac{1}{K} \quad \cdots \quad \frac{1}{K} \right]^T \quad (8)$$

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

$$\mathbf{p}_1^* = [1 \ 0 \ \cdots \ 0]^T \quad (9)$$

$$\mathbf{p}_2^* = [0 \ 1 \ \cdots \ 0]^T \quad (10)$$

$$\vdots \quad (11)$$

$$\mathbf{p}_K^* = [0 \ 0 \ \cdots \ 1]^T \quad (12)$$

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

It is important to stress that we have designed the stability of our system at a local level. The question of global stability and success on the multiagent coordination problem does not follow directly from the local stability of each cell. It might be possible to study the global stability of a system of cells with the *piecewise-φ* rule analytically. The approach in this paper has been to study it through simulation and statistical measures.

4 Simulation

We now present simulations of cells running 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

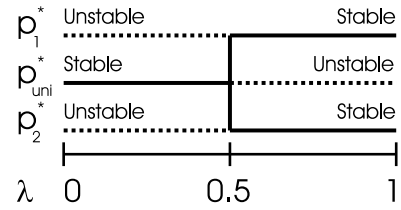


Figure 2: Pitchfork stability of the *piecewise-φ* rule for $K = 2$. λ is a parameter analogous to a temperature.

box and symmetrical connections are formed between two cells if they are closer than a threshold Euclidean distance, d , from one another. Figure 4 shows example connections between $N = 100$ cells with $d = 0.2$. Figure 3 shows example time series for different values of λ . When $\lambda < 0.5$, chaotic global behaviour arises, with $0.5 < \lambda < 1$ fairly successful behaviour 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 behaviour type of the global system. As λ moves from below 0.5 to above, it appears there is a dramatic phase transition in the behaviour of the system (totally chaotic to fixed point). In the neighbourhood of 0.5 there is long transient behaviour. It turns out that the best value for λ , from the point of view of multiagent coordination, is approximately $\lambda = 0.6$.

5 Statistics

In an attempt to quantify the qualitative observations of the previous section a number of statistical measures were employed in the analysis of the SCA time series. These were used also by [Langton, 1990]. The first measure is taken from [Shannon, 1948] and will be referred to as *entropy* (H). It is defined as follows.

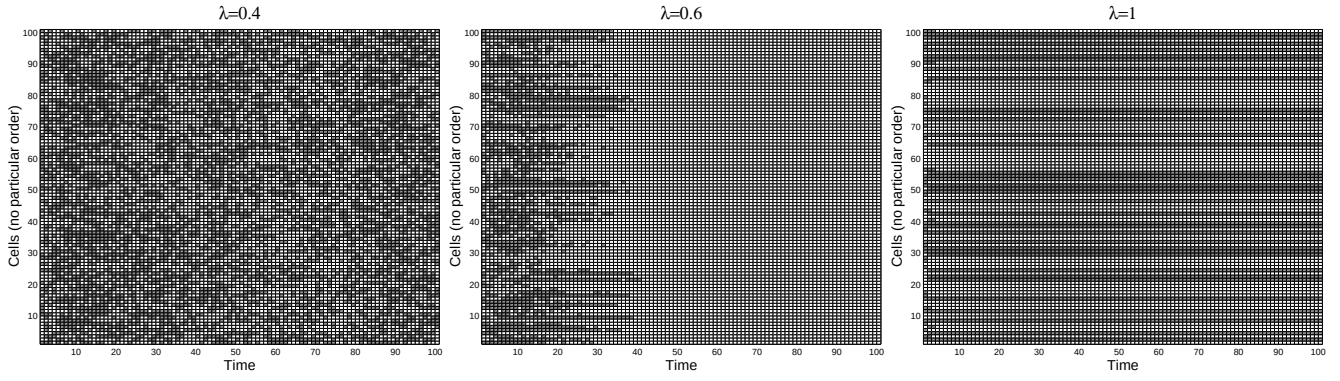


Figure 3: Example time series for different values of λ and $N = 100$, $K = 2$, $d = 0.2$. (left) chaotic behaviour, (middle) successful coordination, (right) clusters. The 2 colours represent the 2 symbols of the alphabet.

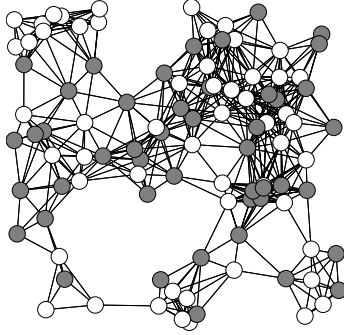


Figure 4: Example connections between $N = 100$ cells with $d = 0.2$. K colours represent the K symbols of the alphabet. Here an initial random condition is displayed for $K = 2$.

Entropy Given a sequence of M symbols

$$\mathbf{s} = [s_1 \ s_2 \ \dots \ s_M]^T \quad (13)$$

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

$$p_k = \frac{n_k}{n_\Sigma} \quad (14)$$

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

$$H(\mathbf{s}) = \frac{-\sum_{k=1}^K p_k \ln(p_k)}{\ln(K)} \quad (15)$$

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

Mutual Information Given two sequences of M symbols

$$\mathbf{s}_1 = [s_{1,1} \ s_{1,2} \ \dots \ s_{1,M}]^T \quad (16)$$

$$\mathbf{s}_2 = [s_{2,1} \ s_{2,2} \ \dots \ s_{2,M}]^T \quad (17)$$

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

$$I(\mathbf{s}_1, \mathbf{s}_2) = H(\mathbf{s}_1) + H(\mathbf{s}_2) - H(\mathbf{s}_1, \mathbf{s}_2) \quad (18)$$

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 $K \times K$).

These two measures may be computed on any sequence of symbols. We tested them on *spatial* sequences (e.g., time series columns from figure 3) and *temporal* sequences (e.g., time series rows from figure 3). 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 I s computed from all rows in a time series. I was computed between a row and itself shifted by one time-step).

Figure 5 show various measures for 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 $d = 0.2$.

Figure 5 (left) shows the average number of clusters¹ at the final time-step for different values of λ . Clearly there is an optimal value of λ near 0.6. Figure 5 (middle) shows 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 at approximately $\lambda = 0.6$ which corresponds to the best performance at multiagent coordination.

Figure 5 (right) displays average temporal mutual information for different values of λ . This is a very interesting plot.

¹Number 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.

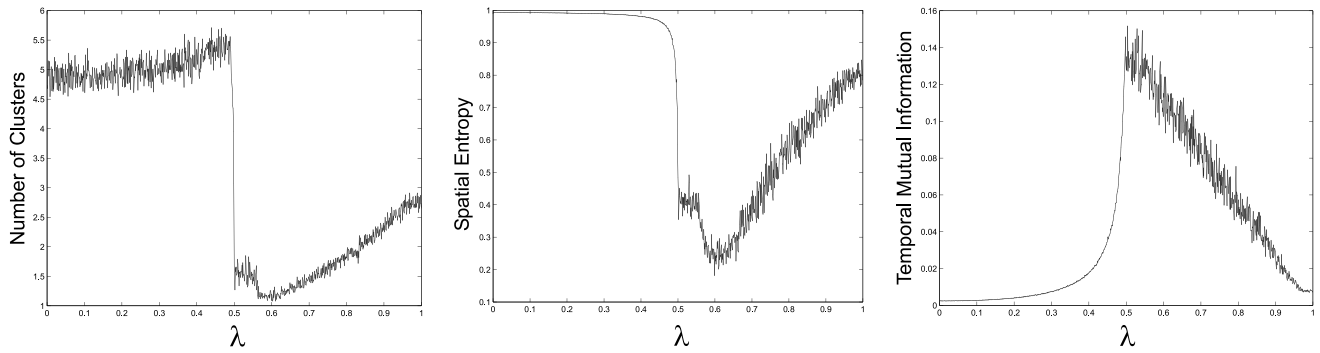


Figure 5: (left) Average number of clusters at final time-step for 1000 values of λ . (middle) Average spatial entropy for 1000 values of λ . (right) Average temporal mutual information for 1000 values of λ . All plots show average of 100 simulations at each value of λ .

Temporal mutual information seems to capture the length of the global transient behaviour of the system. As discussed in [Langton, 1990], the random pattern in the chaotic region is not considered transient but rather the steady state behaviour. The peak in temporal mutual information occurs at $\lambda = 0.5$, the phase transition, and drops away on either side. [Langton, 1990] has a similar plot. Figure 6 shows how the number of clusters at the final time-step (300 time-steps) changes as the problem scales up to more cells; it appears to be a linear relationship. Figure 7 shows how the number of clusters at the final time-step changes for different message sizes.

6 Discussion

The strong correlation between the local stability of the *piecewise- φ* rule and the type of global behaviour is quite interesting. It appears that $\lambda \gg 0.5$ corresponds to fixed point behaviour (class I), $\lambda \ll 0.5$ corresponds to chaotic behaviour (class III), and λ near 0.5 corresponds to long transient behaviour (class IV). The correlation most likely has something to do with the way in which the incoming probability distribution is computed in (1). This step 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. The addition of noise in the particular form of the *piecewise- φ* rule aids in slowing the smoothing process thus destroying the clusters. This has been called *critical slowing down* [Haken, 1983] 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. It is a balance of these two effects which seems to be the most effective at multiagent coordination. 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 $\lambda = 0.6$).

Note that we did not find any oscillatory behaviour (class II) which is likely because the connections between the cells are symmetrical. However, if the *piecewise- φ* rule in figure 1 is reflected (left-right) then the system ‘blinks’ and global coordination corresponds to all cells blinking in phase with one another.

In this model of multiagent coordination, the boundaries between clusters have purposely been made unstable. This forces them to move randomly until they contact one another and annihilate, leaving a single cluster. The 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. Figure 6 confirms this by showing that at the end of 300 time-steps, increasing the number of cells, N , results in more clusters. The linear relationship suggests that scaling-up may be possible but more in depth studies are required. Figure 7 shows how the system scales to different message sizes, K . Here the relationship between number of clusters (after 300 time-steps) to message size is a bit surprising, first dropping then increasing as K increases. It levels off again as the number of symbols exceeds the number of cells (since at most N symbols can be represented in the random initial condition). Again, the nature of this scaling should be studied more closely.

The *piecewise- φ* rule is not the only map that can be used to achieve multiagent coordination in SCA. Replacing it with other monotonically increasing functions (i.e., in figure 1) with the same equilibria also works. We had comparable success to the *piecewise- φ* map using

$$p_{k,\text{out}} = (p_{k,\text{in}})^\beta \quad \forall k \in 1 \dots K \quad (19)$$

with the outgoing probability column normalized as in (6).

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 of figure 4 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 does make 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. This aspect is currently under further investigation.

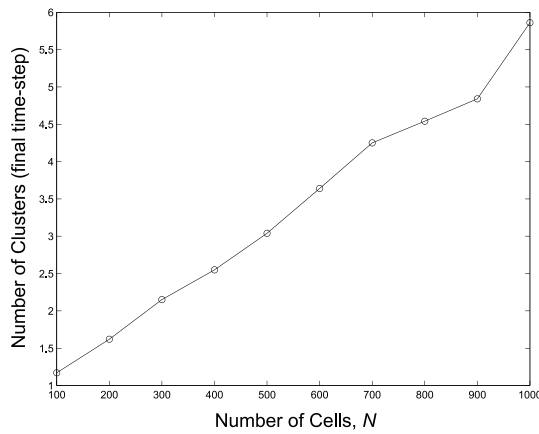


Figure 6: Number of clusters at final time-step as the number of cells is varied from $N = 100$ to $N = 1000$. Parameters for each run were 300 time-steps, $\lambda = 0.6$, $K = 0.2$ and $d = \frac{0.2}{\sqrt{N}}$ to keep the density of connections constant. Plot shows average of 100 simulations at each value of N .

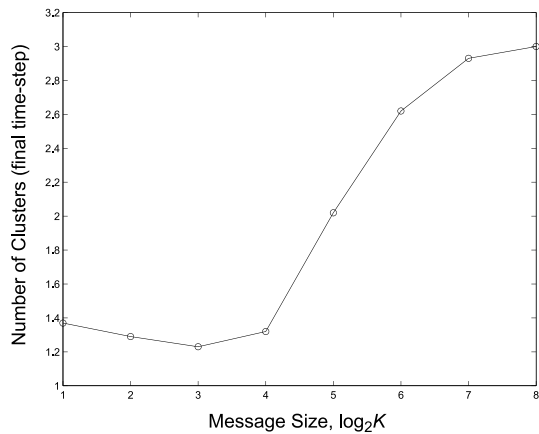


Figure 7: Number of clusters at the final time-step as the message size is varied from $K = 2$ (1 bit) to $K = 256$ (8 bits). Parameters for each run were 300 time-steps, $\lambda = 0.6$, $N = 100$ and $d = 0.2$. Plot shows average of 100 simulations at each value of K .

7 Conclusion

A mechanism for multiagent coordination has been presented based on stochastic cellular automata. We consider this to be an example of self-organizing behaviour in that global coordination occurs in the face of more than one alternative. It was shown that by using stochastic rules, sparsely communicating agents could come to a global consensus. 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 behaviour (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.

Acknowledgements

We would like to acknowledge support of this work by NSERC (Natural Science and Engineering Research Council) and CSA (Canadian Space Agency).

References

- [Andre *et al.*, 1997] David Andre, Forrest H. Bennett, and John R. Koza. Evolution of intricate long-distance communication signals in cellular automata using genetic programming. In Chris G. Langton and Katsunori Shimohara, editors, *Artificial Life V, Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems*. MIT Press, 1997.
- [Bonabeau *et al.*, 1994] Eric Bonabeau, Guy Theraulaz, Eric Arpin, and Emmanuel Sardet. The building behaviour of lattice swarms. In Rodney A. Brooks and Pattie Maes, editors, *Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*. MIT Press, 1994.
- [Das *et al.*, 1995] Rajarshi Das, James P. Crutchfield, Melanie Mitchell, and James E. Hanson. Evolving globally synchronized cellular automata. In L.J. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 336–343, San Francisco, CA, April 1995.
- [Haken, 1983] H. Haken. *Synergetics, An Introduction, 3rd Edition*. Springer-Verlag, Berlin, 1983.
- [Langton, 1990] Chris G. Langton. Computations at the edge of chaos: Phase transitions and emergent computation. *Physica D*, 42:12–37, 1990.
- [Matarić, 1992] Maja J. Matarić. Designing emergent behaviours: From local interactions to collective intelligence. In J. A. Meyer, H.L. Roitblat, and S. W. Wilson, editors, *Simulation of Adaptive Behaviour: from Animals to Animats 2*. MIT Press, 1992.
- [Mitchell *et al.*, 1993] Melanie Mitchell, Peter T. Hraber, and James P. Crutchfield. Revisiting the edge of chaos: Evolving cellular automata to perform computations. *Complex Systems*, 7:89–130, 1993. SFI Working Paper 93-03-014.
- [Shannon, 1948] C E Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, July 1948.
- [Tanaka-Yamawaki *et al.*, 1996] Mieko Tanaka-Yamawaki, Sachiko Kitamikado, and Toshio Fukuda. Consensus formation and the cellular automata. *Robotics and Autonomous Systems*, 19:15–22, 1996.
- [von Neumann, 1966] Jon von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, Urbana and London, 1966.
- [Wolfram, 1984] Stephen Wolfram. Universality and complexity in cellular automata. *Physica D*, 10:1–35, 1984.

Identifying the Scope of Modeling for Time-Critical Multiagent Decision-Making*

Sanguk Noh

Computer Science Department
University of Missouri-Rolla
nohs@umr.edu

Piotr J. Gmytrasiewicz

Dept. of Computer Science and Engineering
University of Texas at Arlington
piotr@cse.uta.edu

Abstract

Decision-making in multiagent settings requires significant computational resources. Agents need to model each other to decide how to coordinate – this sometimes may require solving nested models of many other agents and may be impractical to perform in an acceptable time. In this paper, we investigate ways in which the agents can be equipped with flexible decision-making procedures to allow multiagent decision-making under time pressure. One of the techniques we implemented uses iterative deepening algorithm guided by performance profiles generated offline. When the interaction involves many agents, the algorithm iteratively enhances the quality of coordinated decision-making by incrementally adding the levels of nesting considered, but with the additional penalty of increased running time. To identify the appropriate scope of modeling online we use the concept of urgency which represents cost of delaying decisions. We validate our framework with experiments in a simulated anti-air defense domain. The contribution of our framework is that it endows our autonomous agents with flexibility to cope with time pressure in complex multiagent settings.

1 Introduction

In multiagent settings, deliberative decision-theoretic methods [Gmytrasiewicz *et al.*, 1998; Gmytrasiewicz and Durfee, 2000; Noh and Gmytrasiewicz, 1999a; 1999b; Vidal and Durfee, 1995] allow an agent to compute its best action, i.e., with the highest expected utility, given its available knowledge. The agent's knowledge may contain information about environment, about the agent itself, about other agents present, about what the other agents know, and so on. Therefore, when making coordinated decisions in complex scenarios with many agents

present, the agent may have to consider large and complex knowledge structures, which may lead to it failing to decide an action in an acceptable time.

In response to these issues we investigate methods which allow the agents to perform multiagent decision-making in time-critical situations. In our earlier work [Noh and Gmytrasiewicz, 1999b] we concentrated on compiling results of deliberative decision-making using machine learning and creating reactive rules that improved the agent's reactivity. This current paper focuses on limiting deliberation time by identifying the proper scope of knowledge used for decision-making. We wish to establish which information, assuming it is in the agent's possession in the first place, can be neglected and at what cost. In general, there are two approaches which we call simplification and approximation techniques:

- *Simplifications* involve neglecting a certain part of available information and computation associated with processing this information, while preserving the accuracy of deliberation. In general, there is no guarantee that a simplification technique will be applicable in any particular situation but, if it is, it can be used without sacrificing the quality of the result.
- *Approximations* involve limiting information considered while making decisions without the guarantee that the solution obtained is optimal.

In our framework, the simplification techniques remove computations that are found not to influence the quality of output. For example, if the agent finds that one of its actions is better than all other actions no matter what (we call it the dominant action) then any further computations of the values of actions become unnecessary. Other examples include memoization, and pruning certain parts of structured knowledge representation (defined rigorously in [Gmytrasiewicz and Durfee, 2000] and presented in Figure 1), if it can be proven that they do not influence the solution. All simplifications reduce the computation time needed for deliberation without significant overhead. The approximation techniques determine what computational resources should be used for agent's decision making. These methods consider a tradeoff between decision quality and computation time.

*This research has been sponsored by the Office of Naval Research Artificial Intelligence Program under contract N00014-95-1-0775 and by the National Science Foundation CAREER award IRI-9702132.

In other words, the objective of approximation techniques is to provide autonomous agents with the boundedly rational decision, given the real-time demands of the situation at hand. The approximation techniques include: (1) using compiled rules to constrain alternative actions, on which we reported in [Noh and Gmytrasiewicz, 1999b], and (2) using performance profiles and urgency of situation to determine the appropriate scope of modeling.

In this paper, we will explore the second of the approximation techniques. We will identify the appropriate scope of modeling of other agents based on performance profiles and the calculation of urgency. Toward this end, we focus on the amount of information that can be included in modeling other agents. We are inspired by techniques used by humans in real-life situations when they make coordinated decisions with a large number of individuals. Frequently, we consider and think about only a few agents in our immediate vicinity for the purpose of coordination. Similarly, our agents start by considering their closest neighbors first, and then broaden the scope by including more distant agents (in Figure 1 this would correspond to broadening of, what we call the modeling structure.) At the same time, our agents need to decide on how deep the modeling should proceed. Should one consider how the other agents model others, for example? While reasoning at deeper levels results in better decisions, it also requires more computation time.

Thus, the entire modeling space has the dimension of the *breadth*, which corresponds to the number of agents considered for coordination, and the *depth*, which corresponds to the reasoning level. Based on the structure of the modeling space, we present an iterative deepening algorithm which provides our agents with preemptible results according to different breadths and depths of the modeling space. This algorithm can be used offline to generate *performance profiles* [Dean, 1990; Zilberstein and Russell, 1996], which illustrate the performance vs. computation time required by processing increasingly more information. Further, we introduce the concept of urgency which represents the value of time given a situation at hand. For example, if a situation is very urgent, the agent should be ready to execute an action quickly, without much reasoning efforts. When the situation is not urgent, on the other hand, the agent can afford to consider more information and spend more deliberation time looking for better decisions. Our agents decide whether to act or think by computing urgency.

2 Related Work

We explore the ways to reduce the complexity of deliberative decision-making procedures that might cause autonomous agents to fail in accomplishing their tasks under time pressure, and, at the same time, to preserve the accuracy of deliberation as much as possible. Due to the impracticality of an agent with perfect rationality, it has been proposed that bounded optimality can be a way to yield a solution within a specific time in spite

of sacrificing the optimality of the decision to some degree. Thus, our work builds on efforts by several other researchers who focus on making autonomous systems tractable when there is no sufficient time to produce optimal solutions in complex environments.

One approach uses a purely reactive reasoning procedure. The goal of reactive systems is to directly respond to the condition with a predefined action. Although such a system is able to react very quickly to environments, plans will be useless if problem varies and re-planning is needed. Further, if no protocol or overall plan can be established explicitly in advance, this reactive system is unrealistic or likely to lock the agents into suboptimal forms of behavior. However, our deliberative reasoning procedure provides rational behaviors in dynamic environments even when plans are not pre-defined.

Other rigorous efforts have focused on bounding the rationality of deliberative agents, which limits the resources available to agents. These approaches include work on the use of meta-reasoning procedures to control inference [Horvitz, 1988; Russell and Wefald, 1991], and anytime algorithms [Dean, 1990; Zilberstein and Russell, 1996]. Meta-reasoning and anytime algorithms balance the costs and the benefits of continued deliberation [Good, 1971]. They perform online computation of expected utilities of additional deliberation. In contrast, our approach incorporates offline empirical data (performance profiles) into the decision of whether to act or think. By using the information calculated offline, our meta-reasoning is faster than their meta-reasoning.

In the field of multiagent coordination, there have been several approaches to support practical decision-making. Lesser [Lesser, 1998] proposes a domain independent agent architecture for a large-scale multiagent environment, which deals with adaptation to resource availability, tradeoffs between resources and quality, complex interdependencies among agents, and so forth. For further research, the components of their software infrastructure should be implemented and tested in sophisticated, real-time multiagent applications. Durfee [Durfee, 1999] suggests strategies for making coordination practical when agents use the Recursive Modeling Method (RMM) [Gmytrasiewicz and Durfee, 2000] as a nested model framework. These strategies include simplifying utility computations, limiting possible choices, treating multiple agents as one, and so on. This work is similar to our current work in that both approaches use RMM to make agents' rational decisions. However, we identify the scope of modeling knowledge processed by an agent considering the urgency of the situation at hand whereas Durfee simplifies (or ignores) computational resources as much as possible. Further, our work concentrates on offline efforts to make real-time decision tractable but most of Durfee's strategies for practical coordination have to be executed online. Lastly, in our own work [Noh and Gmytrasiewicz, 1999b], we propose an adaptive and deliberative agent architecture consisting of compiled and deliberative decision procedures. We compile the results of deliberative decision-making into a set of reactive

condition-action rules with numerous machine learning algorithms. An autonomous agent can use the compiled knowledge and constrain the number of alternative actions considered by excluding the ones that are likely to be suboptimal.

3 Control of Reasoning: Act or Think

To show the correlation among the amount of information, decision quality, and computation time within our framework, we compare expected utilities of actions arrived at using different amounts of computational resources, and executed with various time delays. Let N be the set of agents and $A_i, i \in N$, be the set of actions of agent i . The expected utility of the best action, α , arrived at using the body of information E and executed at time t , is

$$EU(\alpha^E, t) = \max_{a_i^j \in A_i} \sum_k P(S_k | E, t, a_i^j) U(S_k) \quad (1)$$

where

- $P(S_k | E, t, a_i^j)$ is the probability over the possible outcome states, S_k , given the available evidence E and the action a_i^j executed at time t .
- $U(S_k)$ is the utility of the resulting state S_k .

Having arrived at the best action, the agent i can execute it immediately, or keep on computing by increasing the body of knowledge considered, and including additional information, e , hoping to arrive at better decision. If it were to keep computing with information $E + e$, the result, α^{E+e} would be available for execution at time $t + t'$, where t' is the computation time used. The decision of whether to keep computing with extra information or whether to execute currently best action comes down to the following comparison:

$$EU(\alpha^{E+e}, t + t') > EU(\alpha^E, t) \quad (2)$$

If the expected utility of possibly higher quality, but delayed, action is greater than the utility of acting on the current best, the agent should keep on computing based on the increased body of information. The challenge is to determine whether the above inequality holds before the computation of α^{E+e} is actually executed. Our approach to flexible decision-making is to use the performance profiles, generated offline, and the notion of value of time, or urgency, to determine whether to act or think. Using performance profiles one can estimate $EU(\alpha^{E+e}, t)$, i.e., the expected value of decision based on information $E + e$, executed currently. Since the actual α^{E+e} is not available, we postulate that the currently available α^E be used to estimate the $EU(\alpha^{E+e}, t + t')$ as follows:

$$EU(\alpha^{E+e}, t + t') = EU(\alpha^E, t) - [EU(\alpha^E, t) - EU(\alpha^E, t + t')] \quad (3)$$

In the above, the value of delayed action α^{E+e} , is estimated based on the assumption that its quality

deteriorates¹ at the rate similar to that of α^E , which has been computed. Substituting the above to Equation 2 and rearranging terms, we get:

$$\begin{aligned} EU(\alpha^{E+e}, t) - EU(\alpha^E, t) > \\ EU(\alpha^E, t) - EU(\alpha^E, t + t') \end{aligned} \quad (4)$$

The left hand side of the above inequality is the benefit of further computation, if it can be carried out instantaneously. This quantity can be estimated from the performance profiles. The right hand side is the current estimate of the value of the extra computation time t' . It is closely related to what we define as *urgency*, or the *value of time*, expressed as the rate with which the expected utility of the currently best alternative deteriorates with time. For instance, if the system slowly processes the data and the urgency is high, then the cost of time needed for computation is greater than the benefit of computation, and it is better not to compute further and act immediately. But if the computation time is short, the value of a unit of time is moderate, and the performance profile rises sharply, the extra computation is preferred to immediate action.

4 Performance Profiles of Iterative Deepening Algorithm

Our deliberative decision-theoretic method is the Recursive Modeling Method (RMM). We have implemented a full-blown version of RMM which allows an agent to compute its best action given all that is known about the other agents, about their states of knowledge, and so on. An agent using RMM represents the hierarchy of its modeling knowledge as a nested modeling structure with: (1) the agent's own decision-making situation (represented as a payoff matrix or as an influence diagram), (2) the next level containing the agent's view of other agents' situations (also payoff matrices or equivalent influence diagrams), (3) the deeper level representing the agent's view of how other agents model others, and so forth (as depicted in Figure 1).

The RMM algorithm, operating on the structure represented in Figure 1, can iteratively deepen its scope by including lower modeling levels. Doing so enhances the quality of decision-making as models nested deeper decrease uncertainties about the agents' beliefs about others in a multiagent domain. In the worst case, when none of the simplification methods is applicable, the complexity and run time of decision-making grow exponentially with the number of agents and the level of reasoning. That is why we consider the tradeoff between decision quality and computation time.

Using iterative deepening RMM algorithm, we generate performance profiles, which plot expected output quality versus computation time in proportion to the

¹In an urgent situation it is better to execute action sooner rather than later. I.e., the expected utility of action decreases as the time passes.

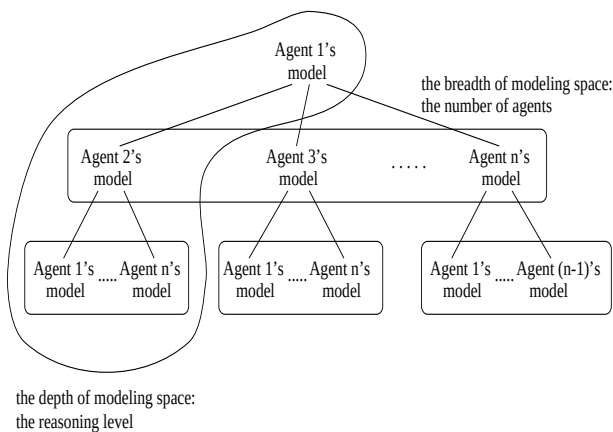


Figure 1: The space of recursively nested models.

breadth and the depth of modeling space. The data for performance profiles is collected offline through simulation of many specific decision-making situations.

Our model of multiagent decision-making under time pressure is the anti-air domain, which consists of a number of attacking targets and coordinating defense units, as depicted in Figure 2. The mission of the defense units is to attempt to intercept the attacking targets so as to minimize damages to own territory. The anti-air defense simulator [Noh and Gmytrasiewicz, 1999a] is written in Common LISP and built on top of the MICE simulator, running on a LINUX platform.

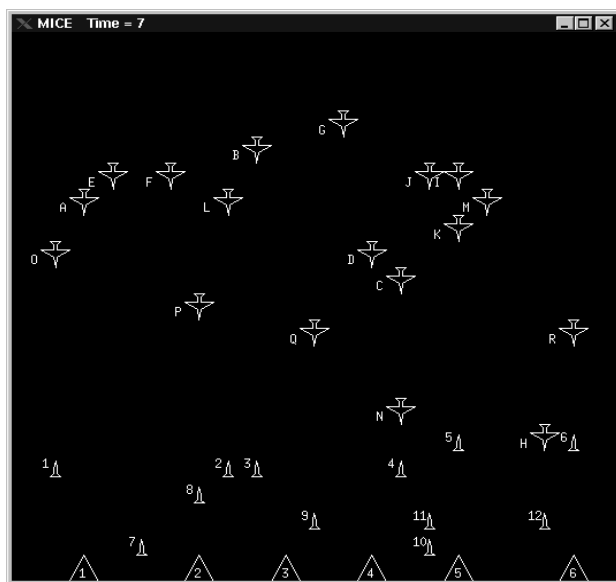


Figure 2: A complex anti-air defense scenario. There are six defense units and 18 incoming targets.

To construct performance profiles using the iterative deepening RMM, we consider six agents and six incoming targets.² The positions and warhead sizes of

²We simplified the scaled-up scenarios for clarity of pre-

targets are randomly generated, and the warhead sizes vary between 100 and 300. We measure the agent's performance in terms of total threat removed. The *total threat removed* is defined as the expected value of a sum of removed warhead sizes of attacking targets. Thus, if a target was chosen for interception, then it contributed (*interception-probability* $\times$ *warhead-size*) to the total threat removed. If a target was not intercepted, it did not contribute at all. The interception probability, $P(H_{ij})$, is dependent on the angle a_{ij} between the target j 's direction of motion and the agent i 's line of sight, as follows:

$$P(H_{ij}) = e^{-\mu a_{ij}} \quad (5)$$

where μ is an interceptor-specific constant (assumed here to be 0.01).

In the experiment, the breadth of modeling space ranges from two to six and the depth of modeling space varies between one and four. The algorithm begins with the simplest modeling space, that is, two agents and reasoning depth one and incrementally adds the amount of information up to six agents and depth four. For baseline performance, we include the performance of agents controlled by random and independent target selection (i.e., without modeling other agents) strategies. The experimental results of average performance (*total threat removed*) and computation time (*seconds* on a 500 MHz PentiumTM III machine) after 100 runs are summarized into Table 1 as discrete entries. In Table 1, the more total threat removed, the better performance.

The random agents, ignoring all information, showed the worst performance, 516.2, and spent the shortest computation time, 0.013. The performance (527.9) and the computation time (0.017) of the independent agents were similar to those of the random agents. The independent agent, maximizing its own expected utility without considering others, did not score well in this coordination experiment and obtained next to the worst performance. Compared the performance of the independent agent, the performance of the agent coordinating by considering the other neighboring agent, the amount of threat removed, 809.7, was drastically increased up by 53.4%. In this case, we clearly see that modeling other agents is advantageous for effective coordination.

As the depth or the breadth of modeling space increased, the performance of agents was improved and the computation time increased. When the number of agents considered for coordination was two or three, the performance gradually improved; the threat removed from our defense area increased from 809.7 to 859.9 when each defense agent considered the other agents, and increased from 906.2 to 1055.0 when it modeled two other agents. However, as more agents and further reasoning depth were included, the result of performance increase leveled off. The amount of improvement from 3 with depth 4 to

sentation here. This simplification is made possible by compiled rules we reported on earlier [Noh and Gmytrasiewicz, 1999b].

Table 1: Performance and computation time of the agents using random, independent and iterative deepening algorithm

Methods	Performance	Computation Time
Random	516.2 $\pm$ 126.6	0.013 $\pm$ 0.007
Independent	527.9 $\pm$ 175.3	0.017 $\pm$ 0.005
2 w/ depth 1	809.7 $\pm$ 105.1	0.040 $\pm$ 0.015
2 w/ depth 2	816.7 $\pm$ 108.9	0.056 $\pm$ 0.020
2 w/ depth 3	841.4 $\pm$ 82.1	0.058 $\pm$ 0.021
2 w/ depth 4	859.9 $\pm$ 57.9	0.062 $\pm$ 0.043
3 w/ depth 1	906.2 $\pm$ 98.3	0.364 $\pm$ 0.047
3 w/ depth 2	1001.3 $\pm$ 60.2	0.374 $\pm$ 0.053
3 w/ depth 3	1050.8 $\pm$ 46.3	0.381 $\pm$ 0.051
3 w/ depth 4	1055.0 $\pm$ 47.9	0.373 $\pm$ 0.054
4 w/ depth 1	1055.2 $\pm$ 48.9	3.036 $\pm$ 0.380
4 w/ depth 2	1055.4 $\pm$ 48.0	3.129 $\pm$ 0.501
4 w/ depth 3	1055.7 $\pm$ 50.8	3.168 $\pm$ 0.556
4 w/ depth 4	1055.9 $\pm$ 47.4	3.202 $\pm$ 0.570
5 w/ depth 1	1057.2 $\pm$ 48.0	32.533 $\pm$ 11.064
5 w/ depth 2	1057.4 $\pm$ 46.6	38.400 $\pm$ 7.026
5 w/ depth 3	1057.7 $\pm$ 46.3	37.675 $\pm$ 6.813
5 w/ depth 4	1058.1 $\pm$ 49.0	37.975 $\pm$ 7.108
6 w/ depth 1	1060.6 $\pm$ 45.3	144.776 $\pm$ 97.834
6 w/ depth 2	1060.8 $\pm$ 44.4	691.043 $\pm$ 166.245
6 w/ depth 3	1061.2 $\pm$ 44.3	689.943 $\pm$ 164.606
6 w/ depth 4	1061.5 $\pm$ 44.0	692.443 $\pm$ 164.804

6 with depth 4 in Table 1 was only 6.5(= 1061.5 – 1055.0) due to more information.

As more breadth and more depth were considered, our agents also spent more computation time. However, because of garbage collection in Lisp, the computation time in some cases did not increase in proportion to the reasoning depth: 3 with depth 4, 5 with depth 3, and 6 with depth 3. As more agents were added the run-times ranged over [0.040–0.062], [0.364–0.381], [3.036–3.202], and [32.533–38.400], respectively. When the six agents were considered for coordination, it should be noted that the computation time at reasoning depth two (691.043) took five times as long as that of reasoning depth one (144.776). Proceeding to more than depth one, the agents (6 with depth 2 – 4) needed to compute additionally five six-dimensional payoff matrices and so used relatively more runtime to construct and evaluate the payoff matrices. In that the dimension of payoff matrix increased according to the number of agents, the computation time spent depending on the number of agents was saliently different.

5 Determining the Scope of Modeling Using Urgency

Our definition of the urgency, as defined in Equation 4, gets at the intuition that an agent stands to lose utility as time progresses in an urgent situation if the agent is not doing anything to remedy the situation. For example, an air defense unit's not trying to intercept the incoming threats may cause the increase in the probability of getting hit as the time progresses, because the

unit's inaction may diminish its chances to successfully intercept the threats. Thus, if the agent's expected utility is inversely proportional to the damages expected in the attack, the situation is urgent in that the defense agent loses utility as time progresses.

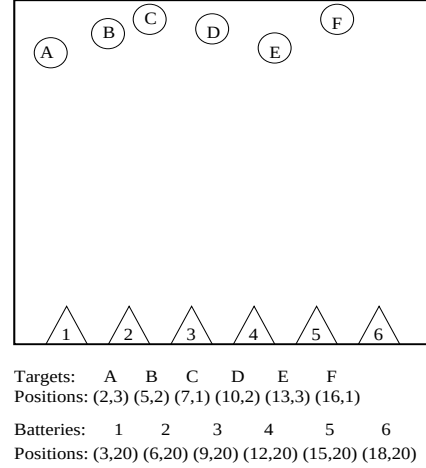


Figure 3: A simplified anti-air defense scenario. There are six defense units and six incoming targets.

As a simple example consider the scenario depicted in Figure 3. Assume that the targets proceed two grids down for one second. In this example, we focus on the agent 1's decision-making situation. At the current time the average interception probability is

$$\text{At time } t, \Sigma_j P(H_{1j})/6 = 0.9499$$

After one second, the targets are coming closer to the defense agents by two grids, the angles between the battery's line of sight and the targets increase, and the interception probability from the agent 1's perspective decreases as follows:

$$\text{At time } t + 1, \Sigma_j P(H_{1j})/6 = 0.9483$$

In our settings as described in the previous section, the average total warhead size is 1,200(= 200 $\times$ 6). Given the interception probabilities at different time steps, the average total warhead size and the computation time, the urgency, or the cost of computation, is

$$\begin{aligned} EU(\alpha^E, t) - EU(\alpha^E, t + 0.017) &= \\ (0.9499 - 0.9483) \times 1,200 \times 0.017 &= 0.0326 \end{aligned}$$

Given the performance profiles in Table 1, the benefit of computation (the left hand side of Equation 4) is

$$EU(\alpha^{E+e}, t) - EU(\alpha^E, t) = 527.9 - 516.2 = 11.7$$

In this example scenario, therefore, since the benefit of computation is greater than the cost of computation, the agent should not take the immediately available action α of shooting at a randomly picked target at the initial time t , but it should consider the extra information

e (consisting of the agent's own decision-making situation), compute $EU(\alpha^{E+e}, t+t')$, and be ready to execute the best action at time $t + 0.017$.

It turns out that the break-even point, at which the agent should cease computation, is to model 3 agents with depth 3 in the example scenario. If the parameters of the particular situation at hand change, however, say when the missiles move faster, they are positioned closer, or if the computation is slower, the optimal scope of reasoning is lower and less computation will take place. In general, the factors of the calculation of value of time include the capability of data processing machine and the external factors imparting urgency to the situation.

We tested the notion of urgency in the anti-air defense settings which contain six defense units and six incoming targets. To identify the appropriate scope of modeling given the offline pre-computed performance profiles (Table 1), each defense agent calculated the benefit and the cost of computation for a level. As a result, after 100 runs, the defense agents stopped reasoning when the average breadth of modeling space was 3 and the average depth of modeling space was 3.017. That is, they considered the adjacent two agents with the reasoning depth just over three, on the average, for their coordinated decision-making. When the agents were under time pressure, thus, they could decide the proper modeling space by calculating their urgency, and reduce the scope and processing time to react faster.

6 Conclusions

The iterative deepening algorithm can be a useful tool that generates partial results according to the breadth and the depth of the modeling space. We generated offline performance profiles by using the iterative deepening algorithm in an anti-air defense domain. We think that the performance profiles for other coordination domains may be similar, but should be obtained separately offline. As the amount of breadth and depth of modeling information increased, the quality of decision was improved. We also extended our multiagent interaction model by introducing the concept of urgency. This notion presents the value of time in a specific situation, when decisions have to be made under time pressure and uncertainty. We integrated this concept into the flexible decision-making framework for identifying the proper scope of modeling information. The most promising aspect of this method is that the reasonable model structure can be employed to avoid wasteful computations and produce a practical decision during real-time.

We experimented with the iterative deepening algorithm and the concept of urgency in the anti-air defense domain to assess agent's decision-making quality. Anti-air defense is certainly a real-time task, and any overhead or loss of timely response can result in the destruction of coordinating agents and areas they are defending. The iterative deepening algorithm and the calculation of value of time prevented catastrophic failure and provided the best decisions in the high-stakes anti-air defense.

We will address the problem of what and how much offline performance profiles are necessary to gain the real-time advantage. For a more systematic evaluation of the framework, the performance profiles using the iterative deepening algorithm will be generated in various domains. In parallel, when insufficient data is available for offline simulation, we will test whether or not our framework produces deterioration in performance of the real-time deliberation.

Another problem to be answered is that we need to validate how each flexible decision-making method proposed can be integrated into a whole procedure. Even though we already tested whether the respective methods can be successfully applied to the time critical situation, we still need to experiment with the whole procedure selectively combined for various situations.

References

- [Dean, 1990] T. Dean. Decision-theoretic control of inference for time-critical applications. *Artificial Intelligence*, pages 1–28, November 1990.
- [Durfee, 1999] E. H. Durfee. Practically coordinating. *AI Magazine*, 20(1):99–116, 1999.
- [Gmytrasiewicz and Durfee, 2000] P. J. Gmytrasiewicz and E. H. Durfee. Rational coordination in multi-agent environments. *Autonomous Agents and Multiagent Systems Journal*, 3:319–350, 2000.
- [Gmytrasiewicz et al., 1998] P. J. Gmytrasiewicz, S. Noh, and T. Kellogg. Bayesian update of recursive agent models. *User Modeling and User-Adapted Interaction: An International Journal*, 8(1/2):49–69, 1998.
- [Good, 1971] I. J. Good. Twenty-seven principles of rationality. In V. P. Godambe and D. A. Sprott, editors, *Foundations of Statistical Inference*, pages 108–141. Holt, Rinehart, and Winston, 1971.
- [Horvitz, 1988] E. J. Horvitz. Reasoning under varying and uncertain resource constraints. In *Proceedings of the 7th AAAI*, pages 111–116, August 1988.
- [Lesser, 1998] V. R. Lesser. Reflections on the nature of multi-agent coordination and its implications for an agent architecture. *Autonomous Agents and Multi-Agent Systems*, 1(1):89–111, 1998.
- [Noh and Gmytrasiewicz, 1999a] S. Noh and P. J. Gmytrasiewicz. Implementation and evaluation of rational communicative behavior in coordinated defense. In *Proceedings of the 3rd Autonomous Agents*, pages 123–130, May 1999.
- [Noh and Gmytrasiewicz, 1999b] S. Noh and P. J. Gmytrasiewicz. Towards flexible multi-agent decision-making under time pressure. In *Proceedings of the 16th IJCAI*, pages 492–498, August 1999.
- [Russell and Wefald, 1991] S. J. Russell and E. H. Wefald. Principles of metareasoning. *Artificial Intelligence*, 49:361–395, 1991.
- [Vidal and Durfee, 1995] J. Vidal and E. Durfee. Recursive agent modeling using limited rationality. In *Proceedings of the 1st ICMAS*, June 1995.
- [Zilberstein and Russell, 1996] S. Zilberstein and S. J. Russell. Optimal composition of real-time systems. *Artificial Intelligence*, 82(1):181–213, 1996.