

# Coevolving Communication and Cooperation for Lattice Formation Tasks

Jekanthan Thangavelautham, Timothy D Barfoot, and  
Gabriele M T D'Eleuterio

Institute for Aerospace Studies  
University of Toronto  
4925 Dufferin Street  
Toronto, Ontario, Canada  
M3H 5T6

`thangav@ecf.utoronto.ca,`  
`tim.barfoot@utoronto.ca,gabriele.deleuterio@utoronto.ca`

**Abstract.** Reactive multiagent systems are shown to coevolve with explicit communication and cooperative behavior to solve lattice formation tasks. Comparable agents that lack the ability to communicate and cooperate are shown to be unsuccessful in solving the same tasks. The agents without any centralized supervision develop a communication protocol with a mutually agreed upon signaling scheme to share sensor data between a pair of individuals. The control system for these agents consists of identical cellular automata handling communication, cooperation and motion subsystems. Shannon's entropy function was used as a fitness evaluator to evolve the desired cellular automata. The results are derived from computer simulations.

## 1 Introduction

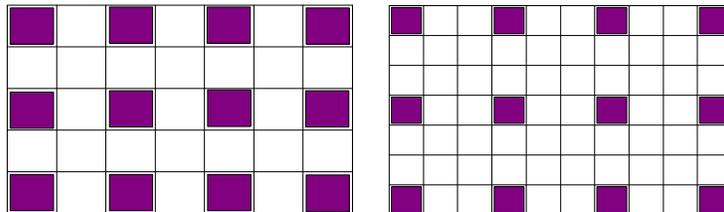
In nature, social insects such as bees, ants and termites collectively manage to construct hives and mounds, without any centralized supervision [1]. A decentralized approach offers some inherent advantages, including fault tolerance, parallelism, reliability, scalability and simplicity in agent design [2]. All these advantages come at a price, the need for multiagent coordination. Adapting such schemes to engineering would be useful in developing robust systems for use in nanotechnology, mining and space exploration.

In an ant colony, each individual is rarely independently working away without explicitly communicating with other individuals [1]. In fact, it is well known that ants and termites use chemicals to communicate information short distances [3]. Cooperative effort often requires some level of communication between agents

to complete a task satisfactorily [6]. The agents in our simulation can take advantage of communication and cooperation strategies to produce a desired ‘swarm’ behavior. The emphasis in this approach is to gain a better understanding of the underlying system dynamics of self-organization.

Our initial effort has been to develop a homogenous multiagent system able to construct simple lattice structures (as shown in fig. 1). The lattice formation task involves redistributing a preset number of randomly scattered objects (blocks) in a 2-D grid world into a desired lattice structure. The agents move around the grid world and manipulate blocks using reactive control systems with input from simulated vision sensors, contact sensors and inter-agent communication. Genetic algorithms are used to coevolve the desired control subsystems to achieve a global consensus. A global consensus is achieved when the agents reach a consensus among the collective and arrange the blocks into one observable lattice structure. This is analogous to the heap formation task in which a global consensus is reached when the agents collect numerous piles of objects into one pile [4].

The paper is organized as follows. Related work is described followed by description of the lattice formation task and the agent model. Results of the genetic run and simulation results are presented, followed by discussion and conclusions.



**Fig. 1.** The lattice structures shown include the  $2 \times 2$  tiling pattern (left) and the  $3 \times 3$  tiling pattern (right).

## 2 Related Work

Balch and Arkin [3] measured the impact of localized communication on multiagent systems based on three simple tasks that include grazing, foraging and consumption. Complex tasks are considered a combination of these simpler tasks. Balch and Arkin focused on two forms of inter-agent communication: state-based communication (state information is transmitted between individuals similar to animal display behavior) and goal-based communication (transmission of goal oriented information such as distance and direction from a source). It was found

that goal and state-based communication have negligible impact on system performance when implicit communication is already present. As in the heap formation task in [4], we expect implicit communication (blocks being channel of communication) to be inherently present within our pre-defined task and expect goal and state based communication to provide little or no benefit [3].

The object of our study has been to determine if localized communication combined with cooperation would produce useful ‘swarm’ behavior to complete a predefined task. Often cooperative tasks involving coordination between numerous individuals such as table-carrying, hunting or tribal survival depend on explicit communication [6]. Communication is required for such cooperative tasks when each individual’s actions depend on knowledge that is accessible to others. Like the heap forming agents, our agents can detect objects over a limited area [4]. It was anticipated that optimal cooperative and communication schemes would allow for sharing of simulated vision data among two or more agents. Earlier works into communication and cooperation were based on a fixed communication language, which may be difficult to develop and may not even be an optimal solution [8–10].

Adaptive communication protocols have been developed combining a learning strategy such as genetic algorithms to develop a desired set of behaviors [6, 11]. MacIennan and Burghardt [11] evolved a communication system in which one agent observed environmental cues and in turn ‘informed’ other agents. Yanco and Stein [6] used two robots, with the ‘leader’ robot receiving environmental cues and informing the ‘follower’ robot. They used genetic algorithms to evolve a communication scheme, equivalent to ‘grunts’ and ‘gestures’ with mutually agreed upon meaning between the two agents.

In considering a decentralized approach, it appears more consistent to have an adaptive language in which changing environmental scenarios will automatically dictate changes in the communication protocol and control system. Coevolving the physical, cooperative and communication behaviors together seemed a natural step.

To our advantage, coevolution has been shown in [7] to be a good strategy in incrementally evolving a solution which combines various distinct behaviors. Initially in a coevolutionary process, two competing populations or, in this case, subagents (in the form of control systems) are unfit. Then one population tries to adapt to the ‘environment’ created by the other and vice-versa until a stable solution is reached. The effect of this parallel evolution is a mutually beneficial end result, which is usually a desired solution [7].

### 3 Lattice Pattern Formation

The multiagent system discussed in this paper consists of agents on a 2-D grid world, with a discrete control system composed of cellular automata. Cellular automata (CA), it has been shown, provide a simple discrete, deterministic model of many systems including physical, biological and computational systems [16,

17]. Each CA agent could be considered a parallel processing computer, in which a set of deterministic rules are used to provide a discrete output for a set of input.

Determining a set of local rules by hand that would exhibit a useful emergent behavior is somewhat difficult and a tedious process. By comparison, evolving such characteristics would produce desired results, provided the right fitness function is found [12]. The CA lookup table for each agent could be considered a gene consisting of  $X$  bits. The key to evolving a good solution is a suitable fitness function, which allows for CA lookup table to be competitively ranked. Using Shannon’s entropy function, we devised a system able to form the  $3 \times 3$  tiling pattern. The 2-D grid world is divided into  $M$   $3 \times 3$  cells,  $A_j$ , where the fitness value,  $f_i$ , for one set of initial condition is given as follows :

$$f_i = s \cdot \frac{\sum_{j=1}^J p_j \ln p_j}{\ln J} \quad (1)$$

where,  $s = 100$  and is a constant scaling factor,  $i$  is an index over many sets of random initial conditions and

$$p_j = \frac{n(A_j)}{\sum_{j=1}^J n(A_j)} \quad (2)$$

where  $n(A_j)$  is the number of blocks in cell  $A_j$ . When the blocks are uniformly distributed over  $J$  cells, we have  $f_i = 100$ . The total fitness,  $f_{total}$ , used to compare competing CA lookup tables is computed as follows

$$f_{total} = \frac{\sum_{i=1}^I n(f_i)}{I} \quad (3)$$

where  $f_i$  is calculated after  $T$  time steps and  $I$  is the number of simulations.

## 4 The Agent

Cooperative behavior between a pair of agents could be visualized as two agents being physically bolted together. To verify whether evolutionary pressure would encourage such a configuration, the agents have the ability to decide whether to stay paired or separate after each time step. The difficulty in this scheme is defining a set of behaviors that would ensure the ‘paired’ behavior would be useful in forming the  $3 \times 3$  tiling pattern.

Each agent is equipped with 3 bumper sensors, 4 spatially modal vision sensors [1, 4] and 2 contact sensors wired to an accompanying trolley. The vision sensors are fitted to allow agents, blocks and empty space to be distinguished. Additional sets of filters are used to distinguish between blocks and other objects for use during a communication session.

### 4.1 Basis Behaviors

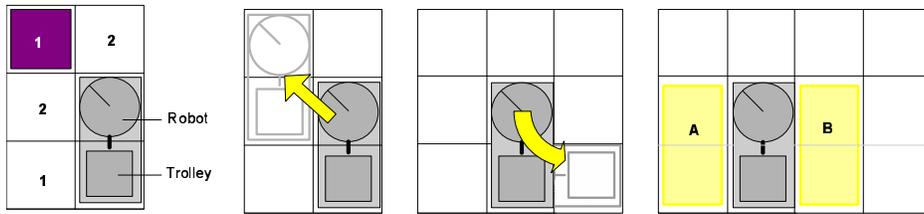
Once the robot has chosen to ‘pair up’ with a neighboring agent, the physical behavior is looked up based on the input from the vision sensors and the data

received from the communication session. When the agent remains 'separated', the received data is taken to be 0.

There are four physical behaviors which are defined as follows:

**I Move:** The agent moves diagonally to the upper left corner if the front and left-side trolley bumper detect no obstacles otherwise the agent rotates left as shown in fig. 2 (center).

**II Manipulate Object:** The choice of whether to put down or pick up a block is made based on whether the agent is already in possession of a block. The agent puts down a block to the left-side diagonal cell if possible, otherwise puts down the block directly to bottom left- side diagonal (if possible). When the agent is not in possession of a block, the agent will attempt to pick up a block by checking directly to the left, if not directly to the right, otherwise directly ahead (see fig. 2 ). If the agent is unable to pick up or put down a block, the 'move' command is activated.

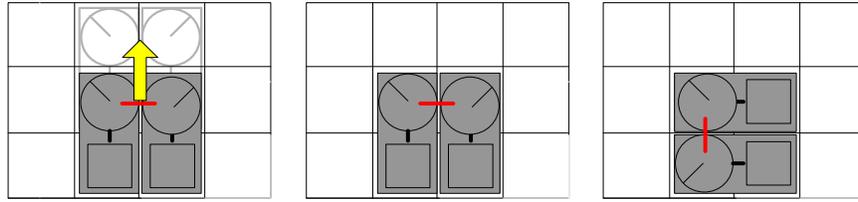


**Fig. 2.** (left) Each agent can detect objects (blocks, other agents or empty space) in the four surrounding squares as shown. The agent can put down a block from squares labelled (1) and pick up a block from squares labelled (2). (center left) Robot shown moving forward. (center right) Robot rotates counter-clockwise. (right) Contact sensors can detect other agents or obstructions in regions marked (A) and (B).

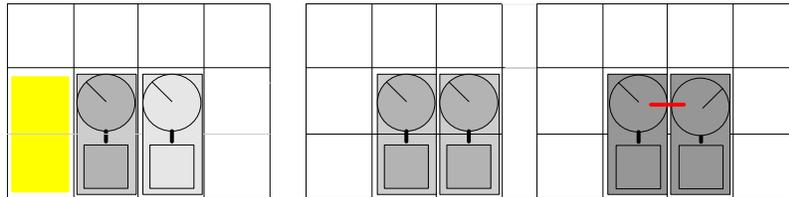
When a pair of agents chooses to move forward, the collective movement is the vector sum of the diagonal movement of each individual agent. The paired entity 'move' and 'rotate' as a single agent from the heap forming problem [4]. This cooperative strategy is employed in the box pushing problem, in which numerous agents are evolved to work cooperatively and push the box in a desired path.

The contact sensors can detect a block, agent (in correct position) or empty space in two equally spaced regions next to the agent (fig. 2). The signals from the pair of contact sensors would undergo a Boolean 'AND' and the combined signal will be used to lookup a response from a Link Lookup Table (LLT). The Link Lookup Table entries consist of two basis behaviors: 'Link' (paired up) and 'Unlink' (separated) , which are defined as follows :

**III Link:** An agent will link to a neighboring agent once the following conditions are met :



**Fig. 3.** (left) A pair of robots moving forward. (center),(right) Paired robots shown rotating counterclockwise.



**Fig. 4.** (left) (1) and (2) show the two regions used to detect if a neighboring agent is in position to ‘link’. (Center) Agents in position to ‘link’ and configuration afterwards (right).

- The neighboring agent is aligned in one of two position (shown in fig. 4).
- Neither agent is already paired up.
- The neighboring agent has also chosen to link.
- If the agent is already paired, then the ‘partner’ must also to choose to link at each time step in order to remain paired up otherwise the agent will undergo the ‘unlink’ behavior. If the agent remains paired, then a basis behavior from the Physical Response Lookup Table (PRLT) will be looked up.

**IV Unlink:** A pair of agents will ‘unlink’, provided the agents are already linked and either one of the agent has chosen to ‘unlink’. If the agent already remains unlinked, the agent will lookup a basis behavior from the Physical Response Lookup Table.

## 4.2 Communication

The agents communicate one bit of information depending on what is detected using the vision sensors. The vision sensors can distinguish between an agent, a block and an empty space. A filter is applied to the vision data to distinguish between a block and an agent or empty space.

The agent as shown in fig. 2 is equipped with four vision sensors with 3 possible outcomes each (block, agent, empty space), two possible outcomes for the agent (carrying a block or not) and an additional two states during a communication session (receive a 1 or 0). The total number of entries in the Physical

Response Lookup Table is:  $3^4 \times 2 \times 2 = 324$  entries. The Communication Lookup Table is connected to the four vision sensors, with 2 possible outcomes (block or no block), resulting in  $2^4 = 16$  entries. The Link Lookup Table has two sets of sensors on each side of the agent with three possible outcomes each (obstacle, agent, empty space), which leads to  $3^2 = 9$  entries. In total there are 349 lookup table entries defined for the cellular automata-based control system.

### 4.3 Initial Conditions

During training, the agents were randomly scattered as ‘unlinked’ pairs throughout the 2-D world. Owing to the discrete nature of the 2-D grid world, it would have been difficult to develop a set of basis behavior such as ‘seek out another agent’, which may extend beyond a single time step. With these limitations in place, some additional constraints had to be imposed to ensure all the agents had a fair chance in pairing up.

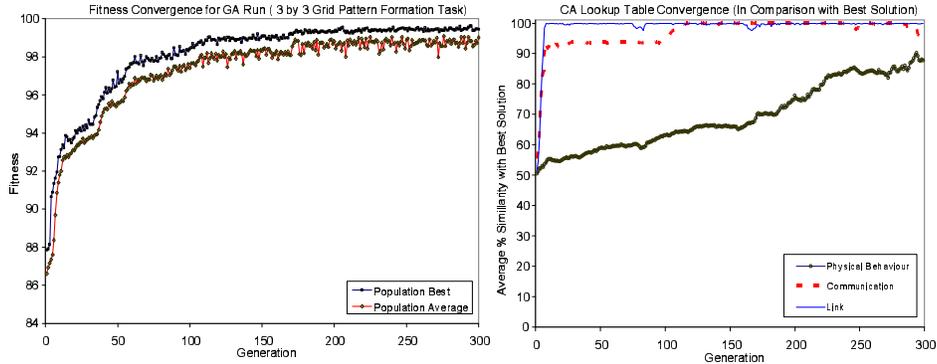
## 5 Simulation Results

In our simulations the GA population size was  $P = 50$ , number of generations  $G = 300$ , crossover probability  $p_c = 0.7$ , mutation probability  $p_m = 0.005$  and tournament size of 5 (for tournament selection). For the GA run, the 2-D world size was a  $16 \times 16$  grid with 24 agents, 36 blocks and a training time of 3000 time steps, where,  $J = 49$  and  $I = 30$  (number of initial conditions per fitness evaluation).

After 300 generations, the GA run converged to a reasonably high average fitness value (about 99). The best individual from the entire GA run was taken to be the solution for the task (fig. 5). The agents learn to pair up and stay paired during the entire training time within the first 5-10 generations. Individuals that failed to pair up were less successful in moving around the grid and manipulating objects, while individuals considered ‘half developed’ would constantly pair up and separate which also retarded the potential for moving around the grid and manipulating objects. Fig. 5 (right) shows the average similarity between the best individual and the population during each generation. As observed from the plot, an optimal Link Lookup Table (LLT) was found within the first ten generations.

The fitness time series averaged over 1000 simulations shows a smooth curve (fig. 7), which is used to calculate the emergence time. The emergence time is defined to be the number of time steps it takes for the system to have organized itself [13, 4]. At a fitness value of 99, the blocks were well organized into the  $3 \times 3$  tiling pattern and more importantly a global consensus (one observable lattice) was formed. For the  $16 \times 16$  world, the emergence time was 2353 time steps. Fig. 6 shows some snapshots from a typical simulation at various time steps.

One of the most cited advantages of decentralized control is the ability to scale the solution to a much larger problem size. To keep the comparison simple and meaningful, constraints had to be imposed to ensure a fair chance in arranging



**Fig. 5.** (left) Convergence history for a typical GA run. (right) CA lookup table Convergence. (in comparison with Best Solution)

a perfect lattice. Based on these constraints, the number of blocks  $B$  is defined as (4) where  $x$  is the length of the grid and  $y$  is the width of the grid:

$$B = \frac{1}{9}(x + 2)(y + 2) \quad (4)$$

The only other parameter left to be varied is the number of agents for each problem size. It was found when the ratio of agents to blocks was low, a global consensus took much longer to occur or never occurred at all. When the agents to blocks ratio is high, the collective effort is hindered by each individual (known to as *antagonism*) [15] and a global consensus is never achieved. Based on these observations, an optimal ratio of agents to blocks was found which minimized the emergence time. A lower bound relationship for this optimal ratio shown is in (5), where  $n_{opt}$  is the optimal number of agents

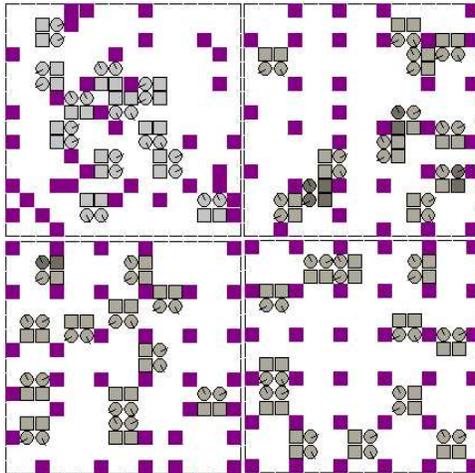
$$n_{opt} \geq 4\sqrt{B} \quad (5)$$

and  $B$  is the number of blocks.

### 5.1 Scaling Up the Solution

Using the optimal ratio of agents to blocks, the simulation was performed for an extended 600,000 time steps to determine the maximum fitness for various problem sizes (fig. 8). The maximum fitness value remained largely constant, as expected, due to our decentralized approach.

By scaling up the problem size, it was naturally expected for the problem to be considered harder to solve and as a result more time steps would be required before the system could reach a global consensus. The fitness time series was averaged over 100 simulations and the emergence time was taken to be 99.6 % of the observed maximum fitness value. It was nevertheless remarkable, that an



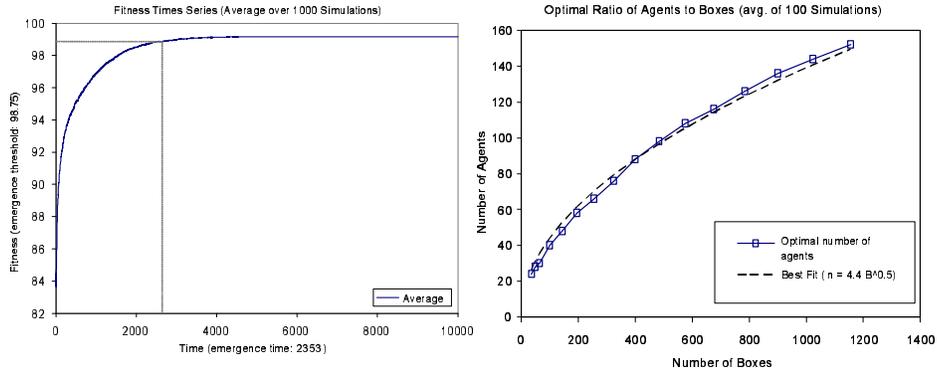
**Fig. 6.** Snapshot of the system taken at various time steps (0, 100, 400, 1600 ). The 2-D world size is a  $16 \times 16$  grid with 28 agents and 36 blocks. At time step 0, neighboring agents are shown 'unlinked' (light gray) and after 100 time steps all 28 agents manage to 'link' (gray or dark gray). Agents shaded in dark gray carry a block. After 1600 time steps (bottom right snapshot), the agents come to a consensus and form one lattice structure.

evolved solution worked well for scaled up problem size (up to  $100 \times 100$  grid) similar to the CA heap forming agents [4]. Comparing the world size (grid area) with the emergence time, the relationship was nicely linear.

## 6 Discussion

It is interesting that a framework for communication between agents is evolved earlier than pair coordination. With the number of lookup table entries for the Communication Lookup Table (CLT) being far fewer than the Physical Response Lookup Table (PRLT), it would be expected for a good solution to be found in fewer generations. Within a coevolutionary process it would be expected for competing populations or in this case subsystems to spur an 'arms race' [7]. The steady convergence in PRLT appears to exhibit this process. The communication protocol that had evolved from the GA run consists of a set of non-coherent signals (equivalent to grunts and gestures), with a mutually agreed upon meaning.

By statistical probability such solutions are easier to evolve, since many more suitable solutions are possible. Yanco and Stein [6] note that such forms of communication are exhibited by "gregarious animals, small children and adult humans lacking common language". Simple cooperative tasks such as table carrying and hunting have been shown to rely more on effective use of signalling than a formal language.



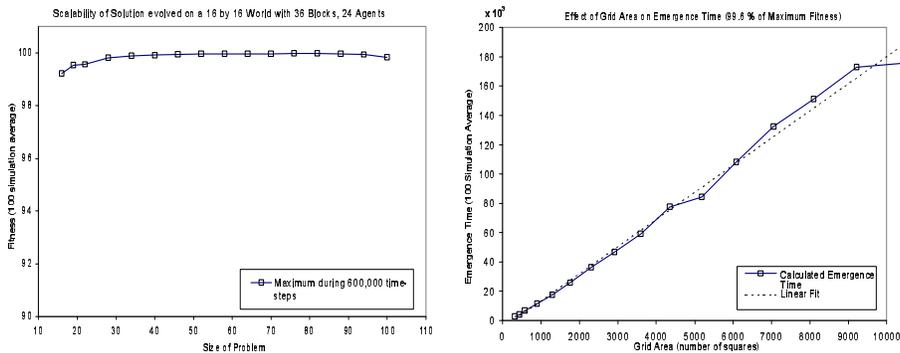
**Fig. 7.** (left) Average fitness time series over 1000 simulations for the  $16 \times 16$  grid with 28 agents and 36 blocks. The calculated emergence time is also indicated (right) Optimal ratio of agents to blocks for problem size of up to  $100 \times 100$  grid.

It was encouraging to witness our cellular automaton-based multiagent systems evolve a non-coherent communication protocol, similar to what had been observed by Yanco and Stein [6] for a completely different task. With their experiment, one of the two robots was always able to provide orders based on environmental cues to the ‘follower robot’. From our findings, it appears cooperative tasks even under decentralized control (where all the members of the system are equal) would require mutually agreed upon signals rather than a formal language to complete a task satisfactorily.

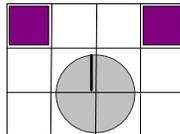
As part our effort to find optimal methods to solving the  $3 \times 3$  tiling pattern formation task, a comparable agent was developed which lacked the ability to communicate and cooperate. As a result each agent had 7 vision sensors, which meant 4374 lookup table entries compared to the 349 entries for the agent discussed in the paper. After having tinkered with various genetic parameters, it was found the GA run never converged.

In this particular case, techniques employing communication and cooperation have reduced the lookup table size by a factor 12.5 and have made the GA run computational feasible. The significant factor is a correlation between the number of lookup table entries and the number of generation required to reach convergence. With the search space being too large, it is suspected the genetic algorithm was unable to find an incremental path to an optimal solution.

It is still a matter of debate whether coevolution alone or whether communication and cooperation were more significant in reaching convergence. To add to the doubt, Paredis showed that coevolution alone failed to produce successful strategies for a cellular automaton-based classification task [14]. Although the classification task bears little resemblance to the pattern formation task, further work would be necessary to account for the significance of both coevolution and communication/cooperation for this task. Evolving agents for a more complex



**Fig. 8.** (left) Maximum fitness values (averaged over 100 simulations) after an extended 600,000 time steps (right) Effect of grid area on emergence time with number of blocks constrained by (4).



**Fig. 9.** The alternate configuration considered for solving the  $3 \times 3$  tiling pattern formation task . The agent occupies 4 squares and can detect objects, agents and empty spaces in 7 squares as shown.

pattern forming task may sound quite formidable but schemes incorporating explicit communication, cooperation and coevolution may provide some hope.

## 7 Conclusion

Our approach to designing a decentralized multiagent system uses genetic algorithms to develop a set of local behaviors to produce a desirable global consensus. A decentralized approach presents some inherent advantages in agent design including scalability and simplicity, which was shown in our simulations. The agents coevolved with localized communication and cooperative behavior can successfully form the  $3 \times 3$  lattice structure. Comparable agents which have bigger lookup tables and lack the ability to communicate and cooperate are unable to perform the same tasks. Our findings show strategies employing cooperation, communication and coevolution can be used to significantly reduce the size of CA lookup tables and make a genetic search more feasible. Interestingly, the agents in our simulations evolve a signaling scheme with a mutually agreed upon meaning to share vision data. A formal language appears to be unnecessary for agents to cooperate even for a decentralized approach (where all the members of the system are equal).

It is hoped our approach will provide some insight into the dynamics of ‘swarm’ behavior and present new alternatives to engineering reactive multiagent systems for self organization tasks.

## References

1. Kube, R., Zhang, H.: Collective Robotics Intelligence : From Social Insects to robots. In Proc. Of Simulation of Adaptive Behavior (1992) 460–468
2. Cao, Y.U., Fukunaga, A., Kahng, A. : Cooperative Mobile Robotics : Antecedents and Directions. In : Arkin, R.C., Bekey, G.A. (eds.): Autonomous Robots, Vol. 4. Kluwer Academic Publishers, Boston (1997) 1-23
3. Balch, T., Arkin, R.: Communication in Reactive Multiagent Robotic Systems. Journal of Autonomous Robots, Vol. 1 (1994) 27–52
4. Barfoot, T., D’Eleuterio, G.M.T.: An Evolutionary Approach to Multi-agent Heap Formation. . In proceedings of the Congress on Evolutionary Computation (1999)
5. Goldberg, D.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Pub. Co., Reading, Mass., (1989)
6. Yanco, H., Stein L.: An adaptive communication protocol for cooperating mobile robots. From Animals to Animats: Proceedings of the Second International Conference on the Simulation of Adaptive Behavior. MIT Press/Bradford Books (1993) 478–485
7. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. The MIT Press. Cambridge, MA, (1992)
8. Matsumoto, A., Asama, H., Ishida, Y.: Communication in the autonomous and decentralized robot system ACTRESS. In Proceeding of the IEEE international workshop on Intelligent Robots and System. (1990) 835–840
9. Shin, K., Epstein, M.: Communication Primitives for Distributed Multi-robot system. In Proceedings of the IEEE Robotics and Automation Conference, (1985) 910–917
10. Fukuda, T., Kawauchi, Y.: Communication and distributed intelligence for cellular robotics system CEBOT. In Proceedings of Japan-USA symposium on Flexible Automataion, (1990) 1085–1092
11. Maclennan, B., Burghardt, G. M.: Synthetic ethology and the evolution of cooperative communication. Adaptive Behaviour, (1994) 161–188
12. Mitchell, M., Crutchfield, J.P., Das, R.: Evolving cellular Automata with genetic algorithms, a review of recent work. In Proceedings of the First International Conference on Evolutionary Computation and Its Application, Russian Academy of Sciences. (1996)
13. Hanson, J.E, Crutchfield, J.P.: Computational mechanics of Cellular Automata : An Example. Working Paper 95-10-095, Santa Fe Institute. Submitted to Physica D, Proceedings of the International Workshop on Lattice Dynamics. (1995)
14. Paredis, J.: Coevolving cellular automata : Be aware of the red queen. In Proceedings of 7th International Conference on Genetic Algorithms. (1997)
15. Dagneff, T., Chantemargue, F., Hirsburnner, B.: Emergence-based cooperation in a multi-agent system. Technical report, University of Fribourg, Computer Science Department. (1997)
16. von Neumann, J.: Theory of self reproducing Automata. Univ. Illinois Press, Urbana and London
17. Wolfram, S. : A New Kind of Science. Wolfram Media, Champaign, IL