

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. Cooperative effort often requires some level of communication between agents to complete a task satisfactorily [5]. The agents in our simulation can take advantage of communication and cooperation strategies to produce a desired 'swarm' behavior.

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 [3].

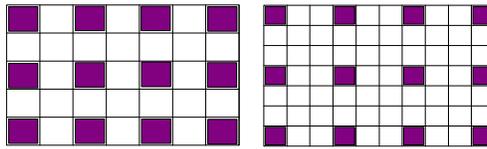


Fig. 1. The lattice structures shown include the 2×2 tiling pattern (left) and the 3×3 tiling pattern (right).

2 Related Work

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 [5]. 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 [3]. 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 [7–9].

Adaptive communication protocols have been developed combining a learning strategy such as genetic algorithms to develop a desired set of behaviors [5, 10]. MacIennan and Burghardt [10] evolved a communication system in which one agent observed environmental cues and in turn ‘informed’ other agents. Yanco and Stein [5] used two robots, with the ‘leader’ robot receiving environmental cues and informing the ‘follower’ robot. To our advantage, coevolution has been shown in [6] to be a good strategy in incrementally evolving a solution which combines various distinct behaviors. Within a coevolutionary process competing populations (or subsystems) spur an ‘arms race’ where 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 [6].

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 [13, 14].

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. Using Shannon’s entropy function, we devised a system able to distribute the objects (blocks) uniformly (a necessary step in forming the 3×3 tiling lattice pattern). The 2-D grid world is divided into M 3×3 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:

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

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 option to stay paired or separate after each time step. Each agent is equipped with 3 bumper sensors, 4 spatially modal vision sensors [1, 3] and 2 contact sensors wired to an accompanying trolley. The vision sensors are fitted to allow agents, blocks and empty space to be distinguished. Once the agent 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.

There are four physical behaviors which are defined as follows:

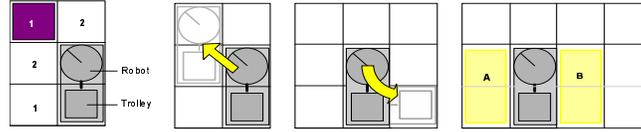


Fig. 2. (left) Each agent can detect objects (blocks, other agents or empty space) in the four surrounding squares as shown. The agent can put down a block from squares labelled (1) and pick up a block from squares labelled (2). (center left) Robot shown moving forward. (center right) Robot rotates counter-clockwise.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

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. If the agent is unable to pick up or put down a block, the ‘move’ command is activated.

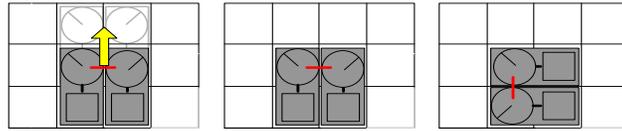


Fig. 3. (left) A pair of agents moving forward. (center),(right) Paired agents shown rotating counterclockwise.

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. 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).

When a pair of agents chooses to move forward, the collective movement is the vector sum of the diagonal movement of each individual agent (fig. 4). 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 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 aligned in in one of two position (shown in fig. 4). The agents are paired only when neither agent is already paired and both have agreed to ‘link’.

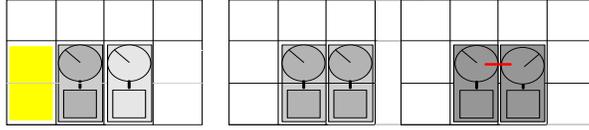


Fig. 4. (left) (1) and (2) show the two regions used to detect if a neighboring agent is in position to ‘link’. (Center) Agents in position to ‘link’ and configuration afterwards (right).

IV Unlink: A pair of agents will ‘unlink’, provided the agents are already linked and either one of the agent has chosen to ‘unlink’.

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.

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 16×16 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 (fig. 5), the GA run converged to a reasonably high average fitness value (about 99). The agents learn to pair up and stay paired during the entire training time within the first 5-10 generations. Fig. 5 (right) shows the average similarity between the best individual and the population during each generation. The fitness time series averaged over 1000 simulations shows a smooth curve (fig. 6), 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 [11,3]. At a fitness value of 99, the blocks were well organized into the 3×3 tiling pattern and more importantly a global consensus (one observable lattice) was formed. For the 16×16 world, the emergence time was 2353 time steps. Fig. 7 shows some snapshots from a typical simulation at various time steps. To keep the comparison simple and meaningful, constraints had to be imposed to ensure a fair chance in arranging a perfect lattice. 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*) [12] and a global consensus is never achieved.

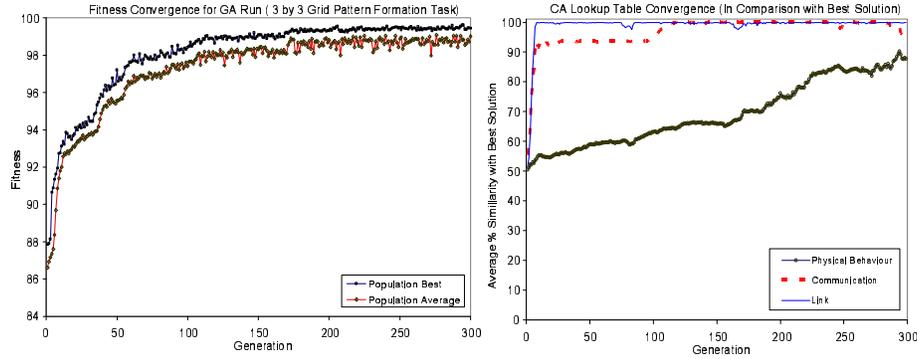


Fig. 5. (left) Convergence history for a typical GA run. (right) CA lookup table Convergence. (in comparison with Best Solution)

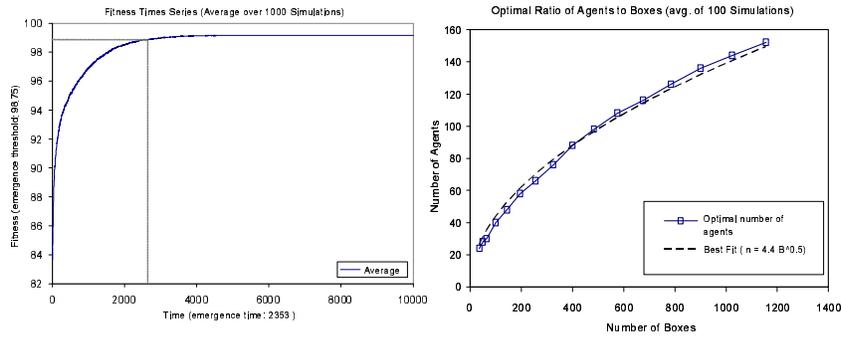


Fig. 6. (left) Average fitness time series over 1000 simulations for the 16×16 grid with 28 agents and 36 blocks. The calculated emergence time is also indicated (right) Optimal ratio of agents to blocks for problem size of up to 100×100 grid.

One of the most cited advantages of decentralized control is the ability to scale the solution to a much larger problem size. 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 (up to 100×100 grid). The maximum fitness value remained largely constant, as expected, due to our decentralized approach. However, further simulations will need to be conducted to confirm the scalability of the evolved solutions.

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

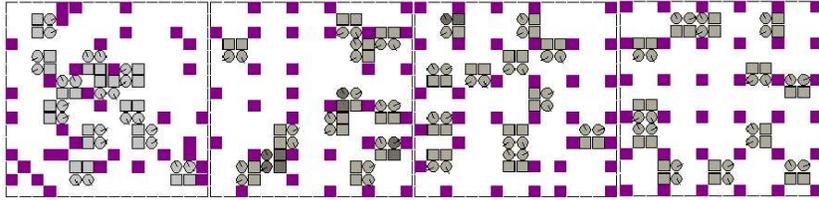


Fig. 7. Snapshot of the system taken at various time steps (0, 100, 400, 1600). The 2-D world size is a 16×16 grid with 28 agents and 36 blocks. At time step 0, neighboring agents are shown 'unlinked' (light gray) and after 100 time steps all 28 agents manage to 'link' (gray or dark gray). Agents shaded in dark gray carry a block. After 1600 time steps (far right), the agents come to a consensus and form one lattice structure.

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' [6]. The steady convergence in PRLT appears to exhibit this process. 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 [5] 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'.

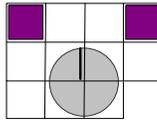


Fig. 8. The alternate configuration considered for solving the 3×3 tiling pattern formation task. The agent occupies 4 squares and can detect objects, agents and empty spaces in 7 squares as shown.

As part our effort to find optimal methods to solving the 3×3 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 generations 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.

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. The agents coevolved with localized communication and cooperative behavior can successfully form the 3×3 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.

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. Barfoot, T., D'Eleuterio, G.M.T.: An Evolutionary Approach to Multi-agent Heap Formation. Proc. of the Congress on Evolutionary Computation (1999)
4. Goldberg, D.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Pub. Co., Reading, Mass., (1989)
5. Yanco, H., Stein L.: An adaptive communication protocol for cooperating mobile robots. From Animals to Animats: Proc. of the Second Int. Conference on the Simulation of Adaptive Behavior. MIT Press/Bradford Books (1993) 478–485
6. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. The MIT Press. Cambridge, MA, (1992)
7. Matsumoto, A., Asama, H., Ishida, Y.: Communication in the autonomous and decentralized robot system ACTRESS. In Proc. of the IEEE international workshop on Intelligent Robots and System. (1990) 835–840
8. Shin, K., Epstein, M.: Communication Primitives for Distributed Multi-robot system. In Proc. of the IEEE Robotics and Automation Conference, (1985) 910–917
9. Fukuda, T., Kawachi, Y.: Communication and distributed intelligence for cellular robotics system CEBOT. In Proceedings of Japan-USA symposium on Flexible Automataion, (1990) 1085–1092
10. Maclennan, B., Burghardt, G. M.: Synthetic ethology and the evolution of cooperative communication. Adaptive Behaviour, (1994) 161–188
11. 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, Proc. of the Int. Workshop on Lattice Dynamics. (1995)
12. Dagneff, T., Chantemargue, F., Hirsburnner, B.: Emergence-based cooperation in a multi-agent system. Tech. report, Univ. of Fribourg. (1997)
13. von Neumann, J.: Theory of self reproducing Automata. Univ. Illinois Press, London
14. Wolfram, S. : A New Kind of Science. Wolfram Media, Champaign, IL