Talking Helps: Evolving Communicating Agents for the Predator-Prey Pursuit Problem

Kam-Chuen Jim[*,‡]
C. Lee Giles[*,†]

Keywords
multi-agent communication, agent communication languages, multi-agent simulation, evolution of agents

Abstract We analyze a general model of multi-agent communication in which all agents communicate simultaneously to a message board. A genetic algorithm is used to evolve multi-agent languages for the predator agents in a version of the predator-prey pursuit problem. We show that the resulting behavior of the communicating multi-agent system is equivalent to that of a Mealy finite state machine whose states are determined by the agents' usage of the evolved language. Simulations show that the evolution of a communication language improves the performance of the predators. Increasing the language size (and thus increasing the number of possible states in the Mealy machine) improves the performance even further. Furthermore, the evolved communicating predators perform significantly better than all previous work on similar prey. We introduce a method for incrementally increasing the language size, which results in an effective coarse-to-fine search that significantly reduces the evolution time required to find a solution. We present some observations on the effects of language size, experimental setup, and prey difficulty on the evolved Mealy machines. In particular, we observe that the start state is often revisited, and incrementally increasing the language size results in smaller Mealy machines. Finally, a simple rule is derived that provides a pessimistic estimate on the minimum language size that should be used for any multi-agent problem.

1 Introduction

An important decision that needs to be made when designing a learning multi-agent system is choosing the sensory information for the system. Providing too little information will result in faster learning but will not allow the system to find an optimal solution. On the other hand, providing too much information can significantly increase the learning time because of the larger search space, though the optimal solution becomes possible. Allowing agents to communicate and to learn what to communicate can significantly ease the burden on the designer. This article studies an ideal case in which each agent has access to a small set of local information and through experience learns to communicate only the additional information that is important.

* NEC Research Institute, Inc., 4 Independence Way, Princeton, NJ 08540
‡ Physiome Sciences, Inc., Princeton, NJ 08540. kamjim@research.nj.nec.com,giles@ist.psu.edu
† School of Information Sciences & Technology and Computer Science and Engineering, Pennsylvania State University, University Park, PA 16801

While many researchers have shown the emergence of beneficial communication in multi-agent systems, very few have looked into how communication affects the behavior or representational power of the multi-agent system. The results of this article contribute further to this area by looking at the relationship between the communication behavior of a multi-agent system and the finite state machine that completely describes this behavior. With this knowledge we can better understand how communication increases the representational power of a multi-agent system.

The role of communication in multi-agent systems remains one of the most important open issues in multi-agent system design [4]. There have been several efforts to standardize communication protocols and languages to facilitate coordination between agents, although these efforts are still relatively immature. The knowledge query and manipulation language (KQML) is a communication protocol for exchanging knowledge and information. A description of KQML can be found in Labrou and Finin [12], but KQML is still a work in progress and its semantics have not been completely defined. The knowledge interchange format (KIF) is a formal syntax for representing knowledge. The KIF language is a prefix version of first-order predicate calculus and has been proposed as a standard for describing knowledge in expert systems and intelligent agents. Speech act theory [18] views human natural language as actions, such as requests, replies, and commitments. Speech act theory standardizes the types of communication acts available to agents. To a receiver agent understanding speech act protocols, the message contained within the communication act may be nonstandard, but there is no ambiguity as to the type of message sent.

Previous work has shown that beneficial communication can emerge in a multi-agent system. Ackley and Littman [1] have shown that agents can evolve to communicate altruistically in a track world even when doing so provides no immediate benefit to the individual. MacLennan and Burghardt [13] used genetic algorithms to evolve finite state machines that cooperate by communicating in a simple abstract world. Walker and Wooldridge [22] studied the emergence of conventions in multi-agent systems as a function of various hard-coded strategy-update functions, including update functions in which agents communicate to exchange memories of observed strategies of other agents. Luc Steels [19] showed that vocabulary can evolve through the principle of self-organization. A set of agents create their own vocabulary in a random manner, yet self-organization occurs because the agents are coupled in the sense that they must conform to a common vocabulary to cooperate through communication. Saunders and Pollack [16] allowed agents to communicate real-valued signals through continuous communication channels. The signals decay over distance and an agent's input on a channel reflects the summation of all the other agents' signals along that channel. Saunders and Pollack assigned these agents to a task in which they need to follow a broken trail of food and showed that it was possible to evolve agents that communicate the presence of food. Balch and Arkin [2] assigned robot agents to three tasks (foraging, consuming, and grazing) and showed that communication significantly improves performance on tasks with little environmental communication, and that more complex communication strategies provide little or no benefit over low-level communication.

While many researchers have shown the emergence of beneficial communication, very few have analyzed the nature of the communication and how communication affects the behavior or representational power of the multi-agent system. Gmytrasiewicz and Durfee developed a "Recursive Modeling Method" to represent an agent's state of knowledge about the world and the other agents in the world [6]. Furthermore, Gmytrasiewicz, Durfee, and Rosenschein [7] used this framework to compute the expected utility of various speech acts by looking at the transformation the speech act induces on the agents' state of knowledge. Hasida, Nagao, and Miyata [9] showed that with certain assumptions, communication can be treated as an n-person game, and the optimal encoding of content by messages is obtained as an equilibrium maximizing the sum of the receiver's and speaker's expected utilities.

Finally, a description of some previous work on the predator-prey pursuit problem is provided in the next section.

2 The Predator-Prey Problem

The predator-prey pursuit problem is used in this article because it is a general and well-studied multi-agent problem that still has not been solved. The predator-prey pursuit problem was introduced by Benda, Jagannathan, and Dodhiawalla [3]. It comprises four predator agents whose goal is to capture a prey agent by surrounding it on four sides in a grid-world. This problem has been used to study phenomena such as competitive coevolution [10, 15, 17], multi-agent strategies, and multi-agent communication. Several researchers have studied the latter two phenomena using the predator-prey pursuit problem. Haynes and Sen [10] used genetic programming to evolve predator strategies and showed that a linear prey (pick a random direction and continue in that direction for the rest of the trial) was impossible to capture reliably in their experiments because such prey avoids locality of movement. Korf [11] studied a version of the predator-prey problem in which the predators were allowed to move diagonally as well as orthogonally and the prey moved randomly. Tan [21] used reinforcement learning and showed that cooperating agents that share sensations and learned policies with each other significantly outperform noncooperating agents in a version of the predator-prey problem. Nishimura and Ikegami [14] observed random swarming and other collective predator motions in a predator-prey game. Stephens and Merx [20] studied a simple noncommunicating predator strategy in which predators move to the closest capture position and showed that this strategy is not very successful because predators can block each other by trying to move to the same capture position. Stephens and Merx also present another strategy in which three predators transmit all their sensory information to one central predator agent who decides where all predators should move. This central single-agent strategy succeeded for 30 test cases, but perhaps the success rate would be much lower if the agents were to move simultaneously instead of taking turns.

Our study uses an implementation that is probably more difficult for the predators than those used in all previous work:

1. In our configuration, all agents are allowed to move in only four orthogonal directions. The predators cannot take shortcuts by moving diagonally to the prey, as they do in [11].
2. All agents have the same speed. The predators do not move faster than the prey, nor do they move more often than the prey, as they do in [10].
3. All agents move simultaneously. Because the agents do not take turns moving (e.g. [20]) there is some uncertainty in anticipating the result of each move. In addition, moving the agents concurrently introduces many potential conflicts; for example, two or more agents may try to move to the same square.
4. The predators cannot see each other and do not know each other's location. If this type of information is essential to getting successful captures then the predators will have to evolve a language that can represent such information.

The world is a two-dimensional torus discretized into a 30×30 grid. Since the world is toroidal, if an agent runs off the left edge of the grid it will reappear on the right edge of the grid, and a similar behavior would be observed vertically. No two agents are

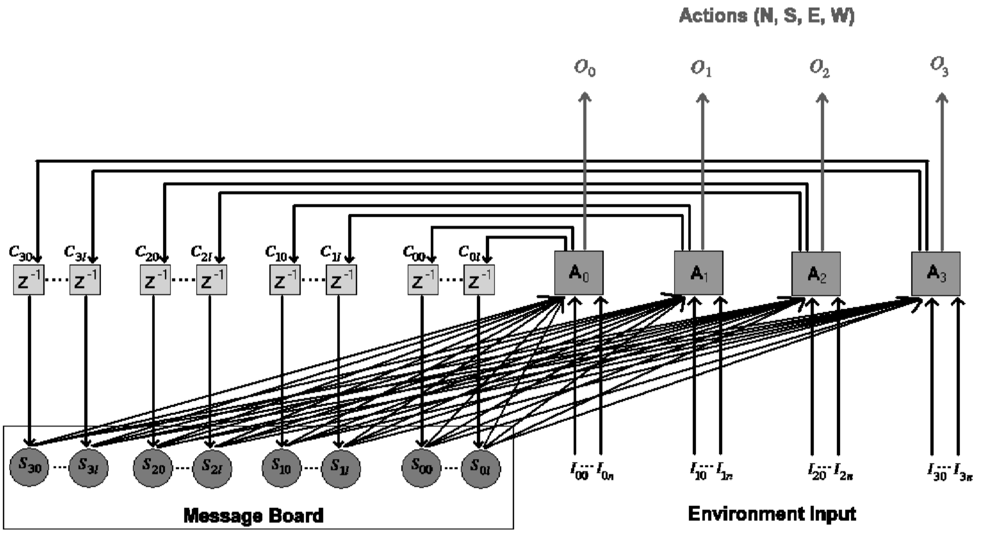


Figure 1. Multi-agent communication as a single finite state machine (FSM). The length of the communication strings is represented by *l*.

allowed to occupy the same cell at the same time. Agents cannot move through each other. If two or more agents try to move to the same square they are all blocked and remain in their current positions. At the beginning of each scenario the predators and prey are randomly placed on different squares. Each scenario continues until either the prey is captured, or until 5,000 time steps have occurred without a capture.

Two prey strategies are used in the simulations. The *random* prey chooses its next action at each time step from the set N, S, E, W using a uniform random distribution. The *linear* prey picks a random direction at the beginning of a trial and continues in that direction for the duration of the scenario. It has been shown that the linear prey can be a difficult prey to capture [10, 19] because it does not stay localized in an area. In our simulations this is an even more difficult prey to capture because the prey and predators move at the same speed.

3 Communication

We studied a simple framework in which all predator agents communicate simultaneously to a *message board* (see Figure 1). At every iteration, each predator agent speaks a string of symbols from a binary alphabet {0, 1}. The communicated symbols are placed on the message board. Each agent then reads all the strings communicated by all the predator agents and determines the next move and what to say next. The strings are restricted to have equal length l. We vary the length l of the strings and study the effect on performance.

3.1 Equivalence to a Finite State Machine

This type of communication can be represented as shown in Figure 1, where $\{A_m\}$ is the set of homogenous predator agents, $\{O_m\}$ is the set of actions of the predators, and $\{I_{mn}\}$ is the set of environmental inputs, where n is the number of inputs and m is the number of communicating agents. The message board can be interpreted as a set of *state nodes*.

The entire set of agents can be viewed as one finite state machine (FSM) with the set of possible states specified by the state nodes $\{S_{ml}\}$. The whole multi-agent system is equivalent to a finite state automaton with output, otherwise known as a finite state transducer. One type of finite state transducer is the Mealy finite state machine, in which the output depends on both the state of the machine and its inputs. A Mealy machine can be characterized by a quintuple $M = (\Sigma, Q, Z, \delta, \lambda)$, where Σ is a finite nonempty set of input symbols, Q is a finite nonempty set of states, Z is a finite nonempty set of output symbols, δ is a "next-state" function that maps $Q \times \Sigma \rightarrow Q$, and λ is an output function that maps $Q \times \Sigma \rightarrow Z$.

It is easy to show that the multi-agent system is a Mealy machine by describing the multi-agent system in terms of the quintuple M. The input set Σ is obtained from the set $\{I_{00}I_{01} \cdots I_{0n}I_{10}I_{11} \cdots I_{mn}\}$ of all possible concatenated sensor readings for the predator agents (for all possible values of I). A description of the sensor readings is provided later in this article. The states Q are represented by concatenation of all symbols in the message board. Since the communication strings comprise binary symbols $\{0, 1\}$, the maximum number of states N_{states} in the Mealy machine is therefore determined by the number of communicating agents m and by the length l of the communication strings: $N_{\text{states}} = 2^{lm}$. The output set Z is obtained from the set $\{O_{00}O_{01} \cdots O_{0p}O_{10}O_{11} \cdots O_{mp}\}$ of all possible concatenated actions for all the communicating agents, where p is the number of bits required to encode the possible actions for each agent (for all possible values of O). In the general case where the actions do not have to be encoded as binary bits, the output set is simply the set $\{O_0 O_1 \cdots O_m\}$ of all possible concatenated actions for the m communicating agents. The next state function δ and output function λ are determined by the agents' action and communication policies. The policies themselves may be FSMs or something with even more representational power; in such a case the multi-agent FSM is a hierarchical FSM.

3.2 Communication Can Help in Partially Observable Environments

From Figure 1 it is clear that communication allows the agents to use state information. This state information is contributed by all communicating agents and represents the state of the entire multi-agent system.

Although each individual agent may maintain its own state information, such information will be limited by the available sensors of the agent. When an agent's next optimal action depends on information that is hidden from an agent's sensors, we say that the agent suffers from the *hidden state problem*. Figure 2 shows an example of a typical hidden state problem that is very common in the predator-prey simulations reported in this article. In this figure, predator 1 sees the same sensory information for two different scenarios because predators cannot sense each other directly. In scenario *a*, predator 1 attempts to move south but is blocked by predator 0 in its path, while in scenario *b* predator 1 is attempting to move south and is not blocked.

Communication allows agents to "tell" each other environmental information that may have been observable only to a subset of the agents. Obviously, communication will be of little use in this respect in the limit when the same set of information is observable to all agents. The message board can be viewed as part of the environment. With this equivalent interpretation, the message board disambiguates the environmental states observed by each agent by providing information that may have been hidden otherwise—assuming the agents are able to communicate effectively.

It is very rare for all agents to have access to the same amount of information. This is because an individual agent will usually have its own internal state that is not observable by other agents. If an agent's state helps determine its behavior, communication may be instrumental in allowing the agents to converge on an optimal plan of action. Thus,

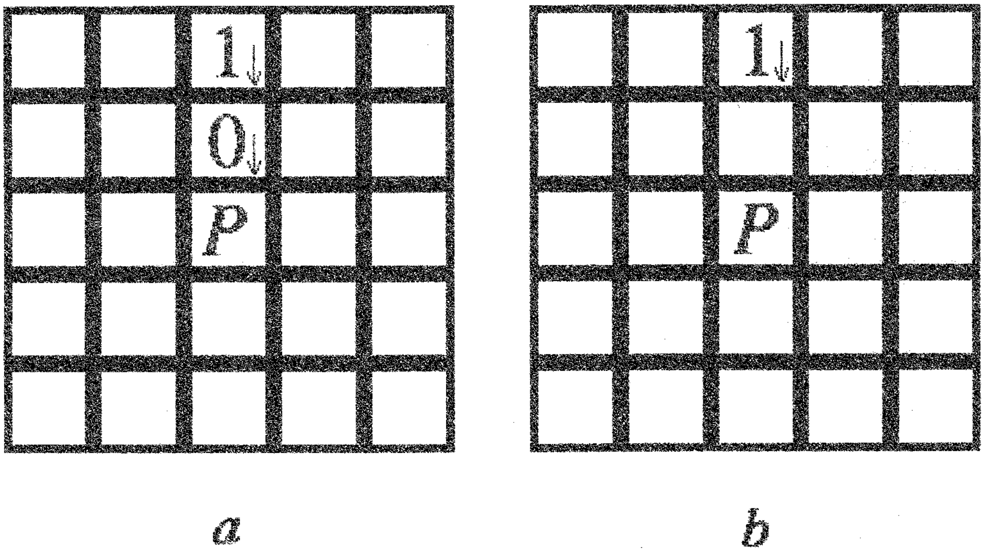


Figure 2. An example hidden state problem. Predator 1 sees the same sensory information for both scenarios *a* and *b*, but in fact scenario *b* is very different from *a*: In scenario *a* predator 1 is blocked, while in scenario *b* it is not. This hidden state problem is due to the fact that the predators cannot sense each other's locations.

even if all agents have access to all possible environmental information, communication may still be helpful by allowing agents to communicate their internal state information.

4 Experimental Setup

A genetic algorithm is used to evolve predators that communicate. A set of experiments is performed with communication strings of varying length l. As the length l increases, the number of strings that are available for communicative acts increases exponentially.

In the sections that follow, genetic algorithm (GA) predators are labeled as GaPredator(l), where l is the length of the communication strings. A communication string of length zero means the predators are not communicating.

The performances of *grown* predators (see Section 4.2 below) are also compared. These predators are labeled as GaPredator($l_0 \rightarrow l_1$), where l_0 is the string length before the agent is grown, and l_1 is the length it was grown to.

Separate populations of GaPredator(0), GaPredator(1), GaPredator(2), GaPredator (0 → 1), and GaPredator(1 → 2) predators are matched against the random and linear preys. The initial GaPredator(0 → 1) population is grown from the GaPredator(0) population with the best average fitness, and similarly the initial GaPredator(1 → 2) population is grown from the GaPredator(0 → 1) population with the best average fitness.

4.1 Encoding Predator Strategies

The behavior of each evolved predator is represented by a binary chromosome string. The length c of the chromosome string is a function of the number of possible states N_{states} observable by the predator based on its sensory information, and the number of actions b_{actions}.

The sensory information available to the predators comprises the range and bearing of the prey, and the contents of the message board. The range (measured in terms of Manhattan distance) and bearing are discretized into $N_{\text{range}} = 4$ and $N_{\text{bearing}} = 8$ sectors,

Table 1. Discretization of predator-prey range and bearing. Range is measured in Manhattan distance.

Distance of prey from predator (# of cells)	Range sector	Bearing b of prey from predator (radians)	Bearing sector
0	0	$-\frac{\pi}{8} < b \leq \frac{\pi}{8}$	0
1	1	$\frac{\pi}{8} < b \leq \frac{3\pi}{8}$	1
2	2	$\frac{3\pi}{8} < b \leq \frac{5\pi}{8}$	2
3+	3	$\frac{5\pi}{8} < b \leq \frac{7\pi}{8}$	3
		$\frac{7\pi}{8} < b \leq \frac{9\pi}{8}$	4
		$\frac{9\pi}{8} < b \leq \frac{11\pi}{8}$	5
		$\frac{11\pi}{8} < b \leq \frac{13\pi}{8}$	6
		$\frac{13\pi}{8} < b \leq \frac{15\pi}{8}$	7

as detailed in Table 1. The number of symbols on the message board is ml, where m is the number of predator agents. The message board can have $N_{\text{messages}} = 2^{ml}$ possible messages. The total number of states that can be sensed by a predator is therefore $N_{\text{states}} = N_{\text{range}} N_{\text{bearing}} N_{\text{messages}}$. The actions comprise the moves $\{N, S, E, W\}$, and speak a string of length l at each iteration. The number of binary bits required to represent the four moves is $b_{\text{moves}} = 2$. Thus, the total number of action bits is $b_{\text{actions}} = b_{\text{moves}} + l$. We arrive at the following equation for the chromosome length c_{ml} of a GA predator that communicates with strings of length l in a team of m predators:

$$c_{ml} = b_{\text{actions}} N_{\text{states}}$$
$$c_{ml} = (b_{\text{moves}} + l) N_{\text{range}} N_{\text{bearing}} 2^{ml} \tag{1}$$

so the chromosome length increases exponentially with communication string length l and number of agents m.

4.2 Growing GA Predators—Coarse-to-Fine Search

To improve efficiency, it would be useful to *grow* the predators. Growing means taking a population of predators that have already evolved a language from a set of possible strings and evolving them further after increasing the set of possible strings they are allowed to communicate. This re-uses the knowledge acquired by predators that were limited to a smaller language. This is effectively a coarse-to-fine search; as we increase the search space by increasing the number of possible strings, the agents can refine the language and communicate other useful, but possibly less critical, information.

By growing the language in these experiments we are making it adaptive. Luc Steels [19] defines an adaptive language as one that "expands or changes in order to cope with new meanings that have to be expressed."

When a population of GA predators with chromosome length c_{ml} is grown to a length of $c_{m(l+1)}$, each new chromosome is encoded such that the behavior of the new predator is initially identical to that of the chromosome it was grown from. The portions of the larger chromosome that are new are not visited initially because the predator is making exactly the same decisions as before and will therefore see the same set of sensory states. During the evolutionary process new sensory states will be visited and the agent will evolve accordingly.

In addition, the population size of the grown $c_{m(l+1)}$ predators is always twice the population size of the c_{ml} predators they were grown from. Half of the population of $c_{m(l+1)}$ predators are grown from the c_{ml} predators; the other half are generated randomly. In this manner the grown predators do not rely solely on mutation for introducing new genetic material to the genes that were copied from the predators with chromosome length c_{ml}. They can obtain new genetic material through crossover with the randomly generated individuals.

4.3 Evaluating the Fitness of Evolved Predators

The fitness of each evolved strategy is determined by testing it on 100 randomly generated scenarios with different starting locations for the predator and prey agents. The maximum number of cycles per scenario is 5,000, after which the predators are considered to have failed. Since the initial population is randomly generated, it is very unlikely that the first few generations will be able to capture the prey. We attempt to speed up the evolution of fit strategies by rewarding those strategies that at least stay near the prey and are able to block the prey's path. The fitness f_i of individual i is computed at the end of each generation as follows, where $N_{\max} = 5000$ is the maximum number of cycles per scenario, $T = 100$ is the total number of scenarios for each individual, and n_c is the number of captures:

- If $n_c = 0$, $f_i = \frac{0.4}{d_{\text{avg}} + 0.6\frac{n_b}{N_{\max} T}}$ where d_{avg} is the average distance of all four predators from the prey during the scenarios, and n_b is the cumulative number of cycles that the prey's movement was blocked by an adjacent predator during T scenarios. The fitness of noncapture strategies can never be greater than 1.
- If $0 < n_c < T$, $f_i = n_c$.
- If $n_c = T$, $f_i = T + \frac{10000T}{\sum_{j=0}^{T} t_j}$, where t_j is the number of cycles required to capture the prey at scenario j.

4.4 GA Setup

The following GA parameters were found experimentally to be most effective. We use two-point crossover with a crossover probability of 0.4. The idea behind multi-point crossover is that parts of the chromosome that contribute to the fit behavior of an individual may not be in adjacent substrings. Also, the disruptive nature of multi-point crossover may result in a more robust search by encouraging exploration of the search space rather than early convergence to highly fit individuals. For a discussion of two-point crossover and generalized multi-point crossover schemes see [5]. A tournament selection scheme [8] with a tournament size Tour of 5 is used to select the parents at each generation. In Tournament selection, Tour individuals are chosen randomly from the population and the best individual from this group is selected as a parent. This is repeated until enough parents have been chosen to produce the required number of offsprings for the next generation. The larger the tournament size, the greater the selection pressure, which is the probability of the best individual being selected compared to the average probability of selection of all individuals. The population size p and mutation rate depend on the length of the communication string because the search space increases exponentially with the communication string length. The larger search space translates into longer chromosome lengths. As a general rule, longer chromosome lengths warrant a larger population size and smaller mutation rate. The population sizes and mutation rates used in the experiments are listed in Table 2.

Ten trials are performed, with the population initialized randomly at the beginning of each trial. The following is a brief description of the algorithm:

1. Repeat the following for 10 trials on selected prey:
   (a) Randomly generate a population of p individuals.

Table 2. Population size and mutation rate GA parameters used in the simulations.

Predator	Population size	Mutation rate
GaPredator(0)	100	0.01
GaPredator(0 → 1)	200	0.001
GaPredator(1)	200	0.001
GaPredator(1 → 2)	800	0.0005
GaPredator(2)	800	0.0005

(b) Repeat until there is no improvement after 200 generations:

i. Simulate each predator strategy on 100 scenarios and evaluate its fitness based on the performance on those scenarios.

ii. Select p individuals from the current population using tournament selection, pair them up, and create a new population by using two-point crossover with mutation.

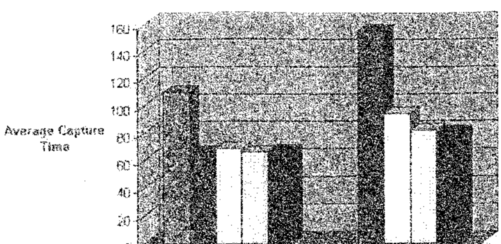
(c) The best strategy found over all generations is used as the solution of this trial. The fitness of this strategy is then recomputed by testing on 1,000 new randomly generated scenarios.

2. The strategy that performed best over all 10 trials is used as the solution.

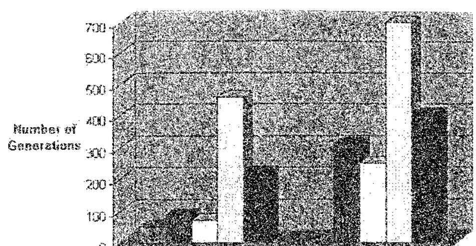
5 Results

Figure 3 shows the best average capture times (over 1,000 randomly generated scenarios) and the cumulative number of evolutionary generations that were needed to achieve such capture times. If $G(l)$ is the number of generations that a GaPredator(l) population was evolved, and $G(l_0 \to l_1)$ is the number of generations that a GaPredator($l_0 \to l_1$) population was further evolved after it was grown from l_0 to l_1, then the cumulative generations for the best GaPredator(0 → 1) and GaPredator(1 → 2) populations are computed as follows:

$$G_{\text{cumulative}}(0 \to 1) = G(0) + G(0 \to 1)$$
$$G_{\text{cumulative}}(1 \to 2) = G_{\text{cumulative}}(0 \to 1) + G(1 \to 2)$$



	Random	Linear
GaPredator(0)	110	no captures
GaPredator(1)	66	155
GaPredator(0->1)	69	94
GaPredator(2)	66	82
GaPredator(1-->2)	67	80



	Random	Linear
GaPredator(0)	50	no captures
GaPredator(1)	65	310
GaPredator(0->1)	70	250
GaPredator(2)	460	700
GaPredator(1-->2)	220	405

Figure 3. Best capture times and the corresponding number of evolutionary generations required to evolve the communicating predators against random and linear prey, at communication string lengths 0, 1, and 2.

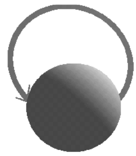


Figure 4. Finite state machine of noncommunicating GA predators. All the links have been combined into one meta-link for simplicity. In other words, when the predators are not communicating they act like an FSM with one state and many links, one link for each possible input combination.

Below is a summary of the performance and convergence results:

- As the length of the communication string increases, the capture time decreases. However, the best capture performance of GaPredator(1) against the random prey is comparable to the best performance of GaPredator(2) and GaPredator($1 \rightarrow 2$), which indicates that a communication string of length 1 was sufficient against the random prey.
- The evolutionary generations required increases with the length of the communication string.
- The capture performance of grown predators is comparable to the performance of the equivalent nongrown predators but requires significantly less evolution time. Thus, incrementally increasing the language size is an effective coarse-to-fine method that reduces the search time.
- The evolved communicating predators perform better than all previously published work to our knowledge. The experimental setup most similar to our work is perhaps that of Haynes and Sen [10], although their setup makes the predators' job easier because they are allowed to move more frequently than the prey. Haynes and Sen and others [11] working on similar prey report results as a percentage of trials that lead to capture, whereas the results reported here show a 100% capture rate when the predators are allowed to communicate.

5.1 Analysis of Evolved FSMs and Communication

This section describes and analyzes the differences in the evolved FSMs as a result of the communication language size. The Mealy machines were obtained by "listening" to the predators talking during actual trials, as opposed to analyzing the predators' GA string to determine what they would say for each possible sensory permutation. This way we only account for states and links on the multi-agent Mealy machine that are ever visited and ignore states and links that do not contribute to the behavior of the predators because they are never visited anyway.

After obtaining the communication activity of the predators, the states of the Mealy machine are constructed by concatenating the words spoken by all predators on the message board. A different multi-agent state is associated with each unique concatenation. The links represent transitions between multi-agent states (i.e., transitions in the content of the message board) at each time step as a result of the inputs sensed from the environment.

Figures 4–7 show the best evolved Mealy machines for noncommunicating and communicating predators that were evolved against the linear prey. The Mealy machines are depicted using what we call scaled finite state diagrams (SFSD). SFSDs provide more information than standard finite state diagrams by representing the relative importance of links and nodes in a visual manner. A scaled finite state diagram is described as

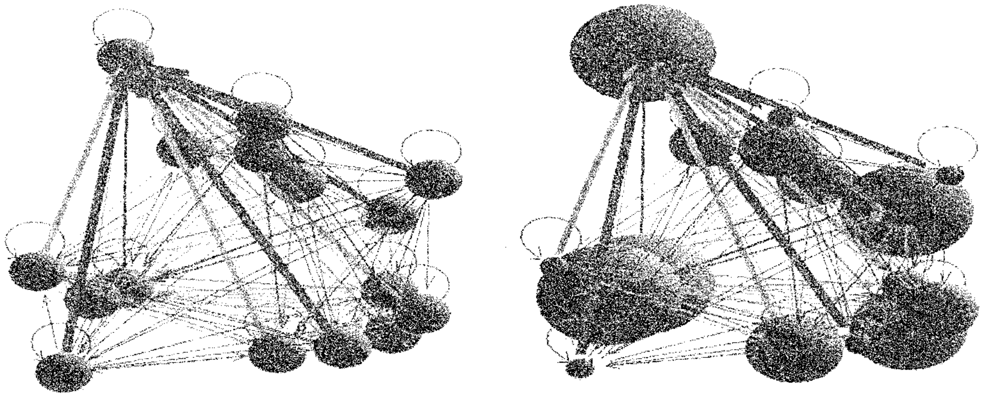


Figure 5. Finite state machine of best communicating GAPredator(1) evolved against the linear prey. All 16 possible states are used. The FSM on the right is the same machine as on the left, except the machine on the right is represented as a scaled finite state diagram (SFSD). States 0, 2, 4, 5, 6, and 8 are more significant than the other states.

follows:

- Links are combined to *meta-links*. A meta-link is an aggregate of all links that connect the same two nodes together, irrespective of their input/output pairs. This simplifies the figures because otherwise the individual links are so numerous that they would completely fill all the space. Also, note that the links are directional, and the end with the arrow points to the next state.
- The thickness of a meta-link indicates the number of individual links that were combined to form the meta-link. A thick meta-link means that many individual links with different input/output pairs were combined to form that meta-link.
- The size of a node indicates its attractiveness and significance. This is measured by the number of incoming links that are connected to that node. A large state node indicates that many environmental input combinations from various states would move the multi-agent system to this state.

Each node is labeled by a number, which is computed by concatenating all the communicated words on the message board and using the language size as the base power. The start node is labeled "0" because at the start of each scenario the message board is initialized to all zeroes.

Observation of the evolved Mealy machines indicate the following:

- The start state is always very significant in the evolved Mealy machines.
- Growing a language results in a Mealy machine with fewer states than an evolved language that was not grown. Compare Figures 6 and 7 and see Table 3, which shows the average number of states in the best Mealy machines over 10 trials as a function of communication size. For example, the average number of states in the evolved GaPredator(2) machines was 252, while for the grown predators GaPredator($1 \rightarrow 2$) the average was only 87. Intuitively this makes sense: The grown FSMs were forced to make do initially with fewer possible states, and as new states became available they were added only when doing so improved performance, or at least did not detract from the performance.

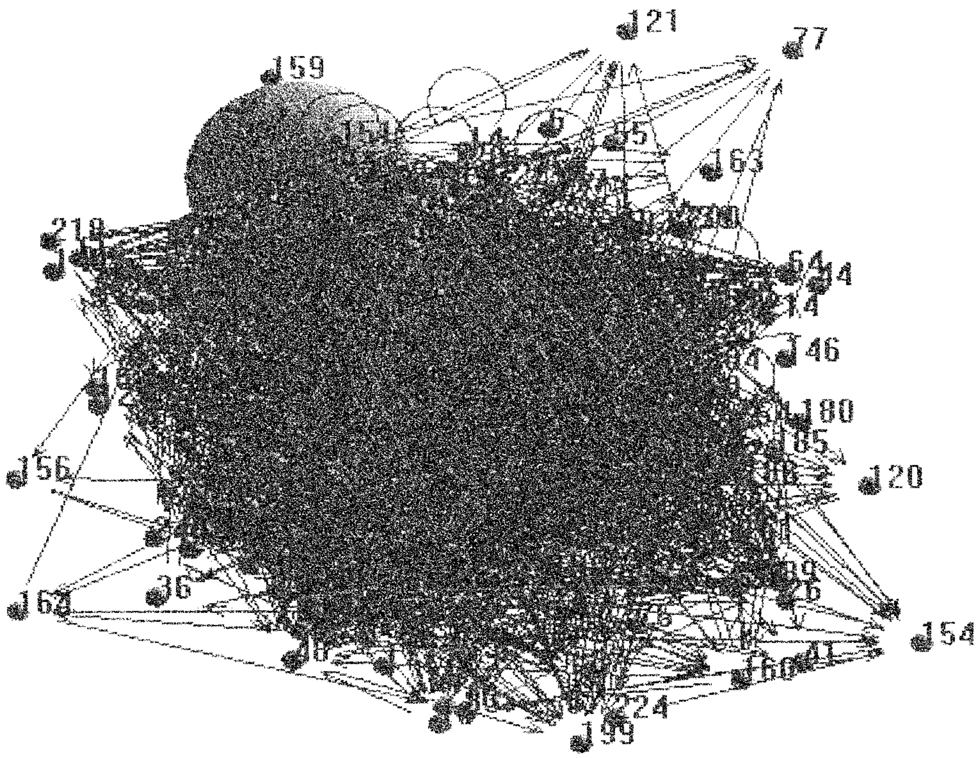


Figure 6. Finite state machine of best communicating GAPredator(2) evolved against the linear prey. All 256 possible states are used.

Table 3. Average number of states in best predators' multi-agent Mealy machine over 10 trials as a function of prey and communication size.

Predator	**Prey**	
	Random	Linear
GaPredator(0)	0	0
GaPredator(0 → 1)	8	16
GaPredator(1)	12	16
GaPredator(1 → 2)	8	87
GaPredator(2)	12	252

- The size of the Mealy machine appears to increase with the difficulty of the problem. See Table 3. For example, the Mealy machines evolved against the random prey are smaller than the Mealy machines evolved against the more difficult linear prey. Also, note that the Mealy machine for GaPredator(1 → 2) (see Figure 7) only uses 87 out of 256 possible states, which indicates that increasing the language size (and thus the number of possible states) would not improve results; it would only increase the number of required evolutionary generations unnecessarily.

5.2 Evolved Languages

Table 4 shows an excerpt of the language evolved by the best GaPredator(0 → 1) agents. This excerpt was obtained by clustering the observed communication activity

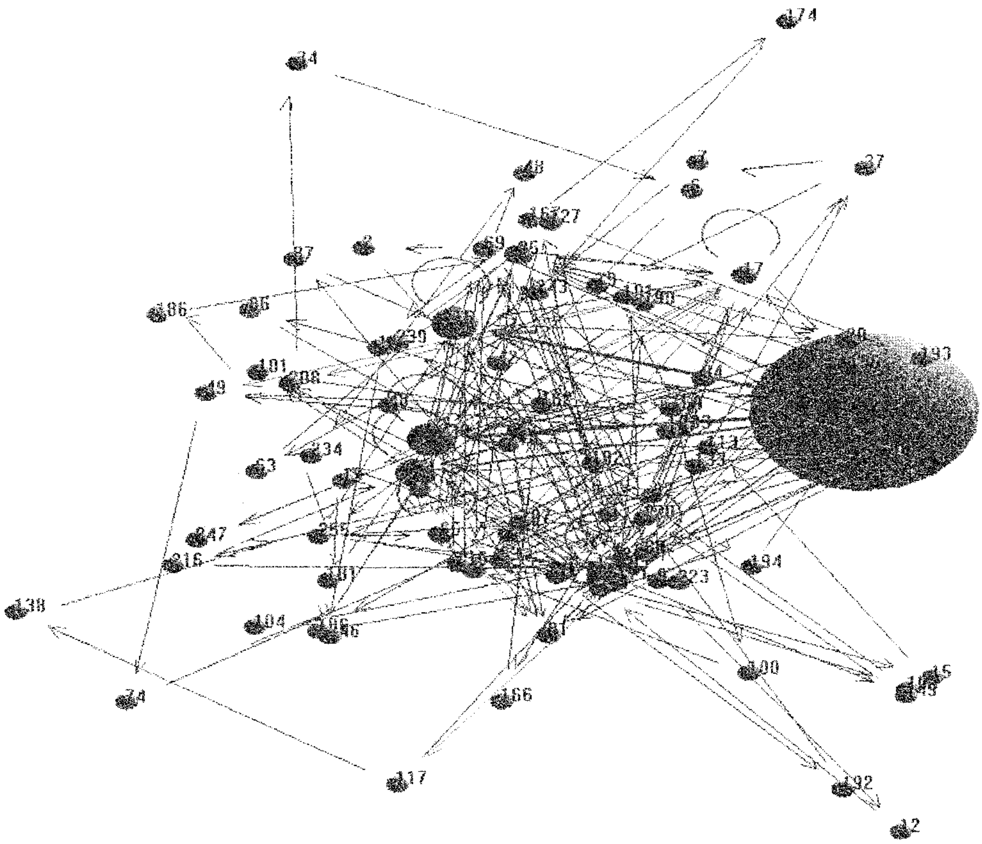


Figure 7. Finite state machine of best communicating GAPredator(1 → 2) evolved against the linear prey. Only 87 out of 256 possible states are used, but the start state (state 0) is much more significant than all other states.

Table 4. Excerpt of the language evolved by best GaPredator(0 → 1) agents.

Input (Range/Bearing)	Message board	Say	Move
2/2	0 0 0 0	0	North
2/3	0 0 1 0	0	West
2/1	0 0 0 1	0	East
1/6	1 0 0 1	1	South
1/1	1 1 1 1	1	South
1/6	0 1 1 1	1	West

using the *minimal spanning tree* algorithm and displaying some of the larger clusters. As an example, the first line is interpreted as follows: "If the prey is to the far north of me (range of 2, bearing 2) and the message board consists of the symbols (0,0,0,0), speak the symbol "0" and move north."

The minimal spanning tree (MST) algorithm is a hierarchical clustering method. Initially, each distinct communication instance is assigned a separate cluster. A communication instance consists of the following information: the agent's sensory information, the contents of the message board, what the agent decides to say, and how the agent decides to move. The MST algorithm proceeds iteratively, at each stage joining the two most similar clusters until a stopping criteria (usually until there is only one cluster left).

The similarity between clusters was measured using a distance metric that weighted the agents' move and sensor information more than it weighted the contents of the message board.

An important observation from the evolved languages is that it is very difficult, if not impossible, to explain the evolved languages. Looking at Table 4, one would be hard-pressed to say, for example, what the symbol "0" means to the predators since there does not appear to be a pattern to its usage. However, the evolved languages are obviously very suitable because they allow the predators to outperform all previous work on similar prey. We thus conclude that allowing the agents to evolve their own communication language is very useful, since it would have been very difficult for a human designer to construct a similar language that can perform as well.

Also, the evolved languages are tightly coupled with the learning problem and cannot be re-used on a different problem. The languages are integrated with the strategies and available actions of the agents in their environment. Therefore, the portability of the evolved languages is dependent on the portability of the evolved multi-agent strategies.

5.3 Semantic Density

Let us define the *semantic density* of a language as the average number of meanings assigned to each word of the language. The semantic density δ can be computed as

$$\delta = \frac{\gamma}{\kappa},$$

where γ is the total number of meanings represented by the language, and κ is the number of words in the language.

We can compute an upper bound γ_{U} on the number of possible useful meanings that the predator agents can communicate. We make the following simplifying assumption: The space of useful meanings that a predator can possibly communicate includes only the agent's sensory information and its next move. This assumption is justified in our simulations because the agents do not have any internal state information that needs to be communicated (our predators do not maintain any internal state), and the agents' plan of action applies only to the current time step. Accounting for the environmental information observable for each agent and the four actions (N,S,E,W) that an agent can take, we get the following equation for the upper bound on the number of useful meanings:

$$\gamma_{\mathrm{U}} = 4N_{\mathrm{range}}N_{\mathrm{bearing}} = 128,$$

where $N_{\mathrm{range}} = 4$ is the number of discrete ranges from the prey, and $N_{\mathrm{bearing}} = 8$ is the number of discrete bearings to the prey. γ_{U} represents the maximum number of unique meanings that a predator agent can possibly communicate regarding its sensory information and its next action.

Assuming that the agents use all the words available to them, an upper bound on the semantic density of the evolved languages in our simulations is simply

$$\delta_{\mathrm{U}} = \frac{\gamma_{\mathrm{U}}}{\kappa} = \frac{128}{2^{l}} = 2^{(7-l)},$$

where l is the length of the binary communication string. Effectively, δ_{U} is the maximum average number of meanings that need to be assigned to each word to allow for an optimal multi-agent strategy that has access to all available local information.

Table 5. The theoretical upper bound δ_U on the meaning density and the average observed upper bound δ^*_U for the best predators. N^* is the average number of states in the evolved multi-agent Mealy machines, shown here again for convenience.

Predator	δ_U	δ^*_U against linear prey	δ^*_U against random prey	N^*
GaPredator(0)	128			1
GaPredator($0 \to 1$)	64	38	10	16
GaPredator(1)	64	38.5	16	16
GaPredator($1 \to 2$)	32	19.75	10	87
GaPredator(2)	32	20	20	252

However, a tighter bound can be obtained by observing traces of the sensory input and movements of all the predators during actual runs. Basically, we observe that in all runs the number of words used is still $\kappa = 2^l$; however, the number of possible meanings γ_U is less than the limit $4N_{\text{range}}N_{\text{bearing}}$ because not all combinations of sensory input and actions are experienced by the agents. In other words, the observed upper bound on the density δ^*_U appears to be much less than the theoretical upper bound δ_U. This is illustrated in Table 5, which shows the theoretical upper bound density δ_U and the average observed δ^*_U for the best predators at each communication string length. The interpretation of δ^*_U is slightly different from the interpretation of δ_U: whereas δ_U is an upper bound that allows for an optimal strategy using all available local information, δ^*_U is an upper bound that allows for the *best evolved strategy* observed, which may or may not be the optimal strategy.

Table 5 indicates that the theoretical upper bound on semantic density is rather large, and it is perhaps unrealistic to expect that a word can have so many meanings in an evolved language. Determining the actual semantic density is a difficult data-mining problem and will not be presented in this article. One would need first to mine for semantics from the data consisting of the sensory logs of each agent and their actions recorded during all runs. Instead, we make the following observations in support of the notion that a relatively high semantic density may in fact be realistic:

- First, it should be noted that the observed upper bounds on semantic density are much less than the theoretical upper bounds, as shown in Table 5.
- There does appear to be heavy re-use of symbols (or words) in the evolved languages. A symbol is used differently depending on the state of the message board. For example, the symbol "1" is used differently when the state of the message board is 1001 versus 0111 in Table 4. Thus the evolved languages are compact and are able to represent more concepts than the 2^l possible symbols available to each agent.
- This re-use of words is also observed in natural languages. It is analogous to contexts. For example, in the English language the word "drive" can mean a compulsion to do something, or a device for storing information, or to guide or control (e.g., "drive a vehicle"), depending on the context. In fact, the word "drive" can be a noun or a verb, and according to an on-line dictionary [23], the verb form can have at least 12 meanings.
- In the communication framework studied in this article, the content of the message board, or equivalently the state of the Mealy machine, determines the context for the spoken symbols. Therefore, the maximum number of contexts per word is equivalent to the number of states in the evolved Mealy machine, and this places a structural upper bound on the semantic density that can be represented by the

multi-agent system. Table 5 shows that for most cases the evolved Mealy machines can more than accommodate the upper bounds on semantic density because the average number of states in the Mealy machines is greater than the semantic density upper bounds. In fact, the cases in which the observed upper bound on semantic density δ^*_{U} is greater than the number of states are exactly the cases in which a larger language improved performance in our simulations. For example, δ^*_{U} against the linear prey with communication strings of length 1 is greater than the number of possible states, and in our simulations increasing the communication length to 2 improved capture performance.

- Thus, one pessimistic estimate for the minimum communication string length l is the following rule:

  Increase l until $N_{\mathrm{states}} \geq \delta_{\mathrm{U}}$,

  where $N_{\mathrm{states}} = 2^{ml}$ is the number of possible states (semantic contexts) in the Mealy machine that represents the multi-agent strategy, and m is the number of communicating agents. The value of δ_{U} will be different for each problem, and indeed it may be difficult to estimate in problems in which one does not know the space of local information available to each participating agent or when the agents maintain internal state information. The number N_{states} can easily be rewritten as a function of the number of words W in the language. For example, in our experiments $W = 2^l$, and N can be expressed as $N_{\mathrm{states}} = W^m$. Thus, our simple rule can be rewritten as

  Increase the number of words in the language until $N \geq \delta_{\mathrm{U}}$.

6 Conclusions

A multi-agent system in which all the agents communicate simultaneously is equivalent to a Mealy machine whose states are determined by the concatenation of the strings in the agents' communication language. Thus, evolving a language for this type of communicating multi-agent system is equivalent to evolving a finite state machine to solve the problem tackled by the multi-agent system. The simulations show that a genetic algorithm can evolve communicating predators that outperform the best evolved non-communicating predators, and that increasing the language size improves performance. A method is introduced for incrementally increasing the language size that results in a coarse-to-fine search that significantly reduces the time required to find a solution. Furthermore, a simple rule is derived for estimating the minimum language size that should be used for any multi-agent problem.

Future work could focus on the semantics of the evolved languages. In addition, more elaborate ways to generate an adaptive language can be explored. Finally, it would be an important step to extend the analysis introduced here to other forms of multi-agent communication structures, such as a system of agents that communicate asynchronously, or only to their nearest neighbors.

Acknowledgments

We would like to acknowledge useful discussions with G. Flake, S. Lawrence, and D. Pennock and suggestions from the referees.

References

1. Ackley, D. H., & Littman, M. L. (1994). Altruism in the evolution of communication. In R. A.

Brooks & P. Maes (Eds.), *Artificial life IV: Proceedings of the International Workshop on the Synthesis and Simulation of Living Systems* (pp. 40–49). Cambridge, MA: MIT Press.

2. Balch, T., & Arkin, R. C. (1994). Communication in reactive multiagent robotic systems. *Autonomous Robots, 1*(1), 27–52.

3. Benda, M., Jagannathan, V., & Dodhiawalla, R. (1985). *On optimal cooperation of knowledge sources.* (Tech. Rep. BCS-G2010-28). Boeing AI Center, Boeing Computer Services, Bellevue, WA.

4. Brooks, R. A. (1991). Challenges for complete creature architectures. In J. A. Meyer & S. W. Wilson (Eds.), *From animals to animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior* (pp. 434–443). Cambridge, MA: MIT Press.

5. De Jong, K. A., & Spears, W. M. (1992). A formal analysis of the role of multi-point crossover in genetic algorithms. *Annals of Mathematics and Artificial Intelligence Journal, 5*(1), 1–26.

6. Gmytrasiewicz, P. J., & Durfee, E. H. (1995). A rigorous, operational formalization of recursive modeling. In V. Lesser (Ed.), *Proceedings of the First International Conference on Multi–Agent Systems (ICMAS)* (pp. 125–132). Cambridge, MA: MIT Press.

7. Gmytrasiewicz, P. J., Durfee, E. H., & Rosenschein, J. (1995). Toward rational communicative behavior. In *AAAI Fall Symposium on Embodied Language* (Tech. Rep. FS-95-05) (pp. 35–43). AAAI Press.

8. Goldberg, D. E., & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. In G. J. E. Rawlins (Ed.), *Foundations of genetic algorithms* (pp. 69–93). San Mateo, CA: Morgan Kaufmann.

9. Hasida, K., Nagao, K., & Miyata, T. (1995). A game-theoretic account of collaboration in communication. In V. Lesser (Ed.), *Proceedings of the First International Conference on Multi–Agent Systems (ICMAS)* (pp. 140–147). Cambridge, MA: MIT Press.

10. Haynes, T., & Sen, S. (1996). Evolving behavioral strategies in predator and prey. In G. Weir & S. Sen (Eds.), *Adaptation and learning in multiagent systems* (pp. 113–126). Berlin: Springer.

11. Korf, R. E. (1992). A simple solution to pursuit games. In *Working Papers of the 11th International Workshop on Distributed Artificial Intelligence* (pp. 183–194).

12. Labrou, Y., & Finin, T. (1994). A semantics approach for kqml—A general purpose communication language for software agents. In *Proceedings of the International Conference on Information and Knowledge Management* (pp. 447–455). ACM Press.

13. MacLennan, B. J., & Burghardt, G. M. (1993). Synthetic ethology and the evolution of cooperative communication. *Adaptive Behavior, 2*(2), 161–188.

14. Nishimura, S. I., & Ikegami, T. (1997). Emergence of collective strategies in a prey-predator game model. *Artificial Life, 3*(4), 243–260.

15. Nolfi, S., & Floreano, D. (1998). Coevolving predator and prey robots: Do “arms races” arise in artificial evolution? *Artificial Life, 4*(4), 337–357.

16. Saunders, G. M., & Pollack, J. B. (1996). The evolution of communication schemes over continuous channels. In P. Maes, M. Mataric, J. A. Meyer, & J. B. Pollack (Eds.), *From animals to animats 4: Proceedings of the 4th International Conference on Simulation of Adaptive Behavior* (pp. 580–589). Cambridge, MA: MIT Press.

17. Savill, N. J., & Hogeweg, P. (1997). Evolutionary stagnation due to pattern-pattern interactions in a coevolutionary predator-prey model. *Artificial Life, 3*(2), 81–100.

18. Searle, J. R. (1969). *Speech acts: An essay in the philosophy of language.* Cambridge, UK: Cambridge University Press.

19. Steels, L. (1996). Self-organizing vocabularies. In C. Langton (Ed.), *Proceedings of Alife V.*

20. Stephens, L. M., & Merx, M. B. (1990). The effect of agent control strategy on the performance of a dai pursuit problem. In *Proceedings of the 10th International Workshop on DAI.*

21. Tan, M. (1993). Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the 10th ICML* (pp. 330–337).

22. Walker, A., & Wooldridge, M. (1995). Understanding the emergence of conventions in multi-agent systems. In V. Lesser & L. Grasser (Eds.), *Proceedings of the First International Conference on Multi-Agent Systems* (pp. 384–389). Menlo Park, CA: AAAI Press.

23. www.dictionary.com.