Coevolution of Role-Based Cooperation in Multi-Agent Systems

Chern Han Yong and Risto Miikkulainen

Abstract—**In tasks such as pursuit and evasion, multiple agents need to coordinate their behavior to achieve a common goal. An interesting question is, how can such behavior be best evolved? A powerful approach is to control the agents with neural networks, coevolve them in separate subpopulations, and test them together in the common task. In this paper, such a method, called Multi-Agent ESP (Enforced SubPopulations), is proposed and demonstrated in a prey-capture task. First, the approach is shown to be more efficient than evolving a single central controller for all agents. Second, cooperation is found to be most efficient through stigmergy, i.e. through role-based responses to the environment, rather than communication between the agents. Together these results suggest that role-based cooperation is an effective strategy in certain multi-agent tasks.**

Index Terms—**Cooperation, Coevolution, Multi-Agent Systems, Heterogeneous Teams, Neuroevolution, Prey-Capture Task, Stigmergy, Communication.**

I. Introduction

In cooperative multi-agent problem solving, several agents work together to achieve a common goal [29, 64, 65]. Due to their parallel and distributed nature, multi-agent systems can be more efficient, more robust, and more flexible than single-agent problem solvers. A central issue with such systems is how cooperation can be best established. First, should the agents be implemented as a diverse set of autonomous actors, or should they be coordinated by a central controller? Second, if the agents are autonomous, is communication is necessary for them to cooperate effectively in the task?

In this paper, these issues are addressed from the machine learning perspective: A team of neural networks is evolved using genetic algorithms to solve the cooperative problem of prey capture. More specifically, the Enforced SubPopulations method of neuroevolution (ESP [15, 16]), which has proven effective in single-agent reinforcement learning tasks, is first extended to multi-agent evolution, in a method named Multi-Agent ESP. This method is then evaluated in a task where a team of several predators must cooperate to capture a fast-moving prey. The main contribution is to show how different ways of encoding, evolving, and coordinating a team of agents affects performance in the task.

Two hypotheses are tested. The first one is that a coevolutionary approach (using Multi-Agent ESP), where autonomous neural networks are evolved cooperatively to each control a single predator of the team, outperforms a central-controller approach, where a single neural-network is evolved (using ESP) to control the entire team. It turns out that niching in coevolution, which is especially strong in ESP [15, 16], extends naturally to multi-agent evolution, making Multi-Agent ESP a powerful approach. Instead of searching the entire space of solutions, coevolution makes it possible to identify a set of simpler subtasks, and optimizing each team member separately and in parallel for one such subtask.

The second hypothesis is that (when allowed by the task) the most efficient way to establish coordination in such teams is through stigmergy [19, 26]. That is, the agents do not communicate with each other directly, but instead observe each others' effects on the environment, such as the changes in the prey's location caused by the teammates' movement. In fact, it turns out that even when a primitive form of communication is available (where each team member broadcasts its location to its teammates), communicating teams consistently perform worse than teams that do not! Each agent has evolved to perform its role reliably, and the task is solved through the stigmergic coordination of these roles. Communication is unnecessary and only complicates the task.

The paper will begin with a brief review of prior work in cooperative coevolution, ESP, agent communication, and the prey-capture task. The Multi-Agent ESP method and its implementation in the prey-capture task is then described, followed by an experimental evaluation of the hypotheses and an experimental analysis of the results. A discussion of future prospects of this approach concludes the paper.

II. Background and Related Work

In this section, the approach to testing the two main hypotheses is motivated by prior work. The cooperative coevolution technique in general, and the ESP method in particular, matches the goal of producing effective cooperative behavior. Coordination based on stigmergy is a potentially powerful alternative to coordination based on communication, and the prey-capture tasks is a suitable platform for studying it.

A. Cooperative Coevolution

Coevolution in Evolutionary Computation means maintaining and evolving multiple individuals, either in a single population or in multiple populations, so that their fitness evaluations interact. The interactions may be competitive or cooperative or both. In competitive coevolution, the individuals have adversarial roles in that one agent's loss is another one's

Chern Han Yong is with the Computational Biology Laboratory, School of Computing, National University of Singapore, Singapore 119077, email cherny@nus.edu.sg.

Risto Miikkulainen is with the Department of Computer Sciences, The University of Texas at Austin, Austin, TX 78712 USA, Email: risto@cs.utexas.edu.

Manuscript received October 24, 2009.

gain [34, 45]. In cooperative coevolution the agents share the rewards and penalties of successes and failures [41, 63]. Cooperative coevolution is most effective when the solution can be naturally modularized into components that interact, or cooperate, to solve the problem. Each component can then be evolved in its own population, and each population contributes its best individual to the solution.

For example, Haynes and Sen [22, 23, 24] explored various ways of encoding, controlling, and evolving predators that behave cooperatively in the prey-capture task. In the first of these studies [22], Genetic Programming was used to evolve a population of strategies, where each individual was a program that represented the strategies of all predators in the team. The predators were thus said to be homogeneous, since they all shared the same behavioral strategy. In follow-up studies [23, 24], they developed heterogeneous predators: Each chromosome in the population was composed of k different programs, each one representing the behavioral strategy of one of the k predators in the team. Haynes and Sen reported that the heterogeneous predators were able to perform better than the homogeneous ones.

In contrast to the advantage found for heterogeneous teams in the above studies, Luke [30] observed that heterogeneous teams could not outperform homogeneous teams evolved using Genetic Programming in the soccer softbot domain. However, he conjectured that, given sufficient time, the heterogeneous approach would have evolved better strategies. Such an extension was not practical in his domain, where each evaluation cycle took between 20 seconds and one minute. Quinn et al. [42] and Baldassarre et al. [3] studied such teams further by evolving neural networks to control teams of homogeneous robots. They found that role allocations would emerge in collective behaviors such as formation movement and flocking. Bryant and Miikkulainen [7] further showed that homogeneous teams can reorganize their role allocations dynamically as task requirements change. However, when all agents must be capable of all behaviors, it is difficult to generate sufficiently diverse behaviors. Heterogeneous teams therefore promise to be more general, scaling up to a wider variety of behaviors.

Balch [2] demonstrated an important principle in learning diverse behaviors in heterogeneous teams, by studying the diversity of robot teams developed through reinforcement learning. He found that when the reinforcement was local, i.e. applied separately to each agent, the agents within the team learned identical behaviors. Global reinforcement shared by all agents, on the other hand, produced teams with heterogeneous behavior. This result provides a useful guideline for evolving cooperating agents: Rewarding the whole team for good behavior privileges cooperation even when some agents do not contribute as much as others, whereas rewarding individuals induces more competitive behaviors because each individual tries to maximize its own reward at the expense of the good of the entire team.

This principle has been utilized effectively in cooperative coevolution of neural networks. Instead of a population of full networks that could potentially solve the task, a population of partial solutions, i.e. partial networks or neurons or connections, are evolved [17, 18, 36, 38, 41, 44]. Such methods are powerful because they break the problem of finding a solution network into smaller subproblems, making search faster, more focused, and diverse. One useful such approach is Enforced SubPopulations, as will be described next.

B. Enforced SubPopulations (ESP)

The Enforced SubPopulations neuroevolution method (ESP[1]; [15, 16]) is an extension of Symbiotic, Adaptive NeuroEvolution (SANE; [35, 36, 37]). SANE is a method of neuroevolution that evolves a population of neurons instead of complete neural networks. In other words, in SANE each chromosome represents the connections of a single neuron instead of the structure and weights of an entire network (analogous to the "Michigan" method of evolving rule-based systems, where each chromosome represents a single rule [25], versus the entire rule set as in the "Pitt" method [51]). Neurons are selected from the population to form the hidden layer of a neural network, which is then evaluated on the problem. Each neuron's fitness is the average fitness of all networks in which it participated. ESP extends SANE by allocating a separate population for each hidden-layer neuron of the network; a number of neuron populations are thus evolved simultaneously (Figure 1). ESP is thus a cooperative coevolution method: Each neuron population tends to converge to a role that results in the highest fitness when the neural network is evaluated. In this way, ESP decomposes the problem of finding a successful network into several smaller subproblems, resulting in more efficient evolution [15, 16].

The ESP idea can be further extended to evolution of individual connections [18]. In several robot control benchmark tasks, ESP and its connection-level extension, CoSyNE, were compared to other neuroevolution methods such as SANE, GENITOR [60], Cellular Encoding [20, 61], Evolutionary Programming [47], Evolutionary Strategies [21], and NEAT, [52] as well as to other reinforcement learning methods such as Adaptive Heuristic Critic [1, 4], Policy-Gradient Reinforcement Learning [55], Q-learning [40, 58], Sarsa(λ) [46], and VAPS [32]. Because of its robust neural network representation and efficient search decomposition, ESP-based methods turned out to be consistently the most powerful, solving problems faster, and solving harder problems [14, 16, 17, 18].

These results lead to an intriguing hypothesis about constructing multi-agent systems: If a neural network can be evolved effectively as a set of cooperating neurons and connections, perhaps a team of agents can be evolved the same way. In particular, perhaps the ESP method can be extended to multi-agent evolution, where not only the neurons but also the networks formed from them are required to cooperate in a team in order to receive a high fitness. Such a method, called Multi-Agent ESP, is developed and tested in this paper.

An important question immediately arises: How strongly should the agents in multi-agent evolution be coupled? Neurons in a network are tightly connected; does cooperation in a multi-agent task require that the agents communicate extensively?

[1] ESP neuroevolution software is available at http://nn.cs.utexas.edu/soft-list.php

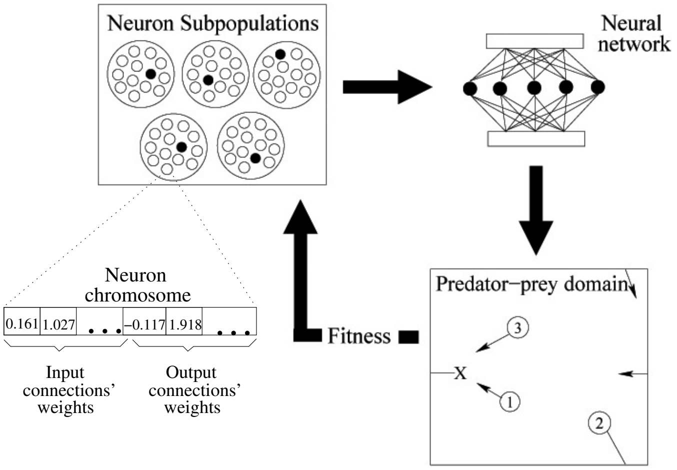

Fig. 1. **The ESP Method.** Each subpopulation of neurons contributes one neuron (with its input and output connections) to form the hidden layer of the neural network, which is then evaluated in the task. The fitness is passed back to the participating neurons. This scheme is used to evolve the central-controller neural network (Figure 4) that controls all three predators simultaneously. The extension to multiple controllers (the Multi-Agent ESP method) is depicted in Figure 3.

C. Agent Communication

Communication is a diverse field of research, and can be defined in various ways depending on the focus of inquiry. For the practical purpose of this paper, communication can be defined as the process where an agent transmits information to one or more other agents on purpose. This definition captures an important class of information transfer processes, differentiating them from e.g. sensing, which does not require that the agent makes the information available on purpose, and from various forms of emitting information that do not require a receiving agent.

The role of communication in cooperative behavior has been studied in several artificial life experiments [5, 8, 10, 11, 27, 48, 54, 59]. These studies showed that communication can be highly beneficial, even crucial, in solving certain tasks. For instance, complementing the results in this paper, Floreano et al. [11] demonstrated conditions under which communication emerges and is useful in teams of homogeneous agents. However, the cost of communication—such as the energy expenditure in signaling, or the danger of attracting predators, or the complexity of the apparatus required—was not taken into account in most of these studies. In fact, even in domains where communication does contribute toward solving a task, communicative traits may still not evolve if they involve a significant cost. Other kinds of cooperative strategies may evolve instead, depending on the nature of the task, how dense the population is, and whether resources are available [57].

One particularly interesting form of cooperation without communication is stigmergy, a concept proposed by Grassé [19] to describe the coordinated behaviors of social insects. Grassé observed that worker termites were stimulated to perform certain activities by a particular construction of their nest, transforming it into a new construction, which would in turn stimulate other activities. The word stigmergy was coined to describe this process: "The stimulation of the workers by the very performances they have achieved is a significant one inducing accurate and adaptable response, and has been named *stigmergy*" [19] (translated by Holland and Melhuish [26]).

Holland and Melhuish [26] examined stigmergy and self-organization in a group of robots that clustered and sorted frisbees, and found that the task was solvable using stigmergy-based coordination without any communication between robots. Franklin [13] proposed that stigmergic coordination may be an advantage over communication, because communication and the associated explicit planning between agents requires additional architecture, intelligence, and resources. An interesting hypothesis therefore arises for constructing multi-agent systems: In tasks where stigmergic coordination is possible, the task may be accomplished more effectively without communication.

This hypothesis is evaluated in the experiments presented in this paper in three ways: (1) By showing how stigmergy can emerge in cooperative coevolution when communication between teammates is not available; (2) by comparing the evolution performance of communicating teams and non-communicating teams; and (3) by comparing the emergent behaviors of communicating teams and non-communicating teams. To facilitate discussion, non-communicating cooperation based on stigmergic coordination is termed *role-based cooperation*. In certain tasks, this cooperative strategy is easier to evolve and more powerful than a strategy based on communication.

In order to make evaluation transparent, a most elementary form of communication is employed in these experiments. Whereas each non-communicating agent is completely unaware of its teammates, each communicating agent continuously broadcasts its location to its teammates. Although elementary, such exchange of information constitutes useful communication: In many real-world applications involving software agents or physically situated robots, it is not possible to sense the teammates' locations directly, because they may be far away or obscured or the appropriate sensors may not be available. A communicative apparatus (such as a radio system) is required to obtain this information. On the other hand, while other, more complex forms of communication are possible, the broadcasting of locations is sufficient to demonstrate differences between teams that communicate and those that do not. It therefore forms a suitable communicative process for studying cooperation with and without communication.

D. Prey-Capture Task

The experimental platform for studying cooperation and communication in this paper is the prey-capture task. It is one example of the general class of pursuit-evasion tasks [33]. Such tasks consist of an environment with one or more prey and one or more predators. The predators move around the environment trying to catch the prey, and the prey try to evade the predators. Pursuit-evasion tasks are interesting because they are ubiquitous in the natural world, offer a clear objective that allows measuring success accurately, and allows analyzing and visualizing the strategies that evolve.

Pursuit-evasion tasks generally cannot be solved with standard supervised learning techniques like backpropagation. The correct or optimal decisions at each point in time are usually not known, and the performance can be measured only after several decisions have been made. More complex algorithms are required that can learn sequences of decisions based on sparse reinforcement. Pursuit-evasion tasks are challenging for even the best learning systems because they require coordination with respect to the environment, other agents with compatible goals, and adversarial agents [15].

The prey-capture task focuses on the predators' behavior. It has been widely used to test multi-agent coordination and communication. As was described in previous subsections, for example Benda et al. [6] and Haynes and Sen [23, 24] used this task to assess the performance of different coordination systems, and Jim and Giles [27] studied the evolution of language and its effect on performance. In the variant of the task used in these studies, the predators are required to surround the prey in specific positions to catch it, and the main difficulty is in coordinating the predators to occupy the proper capture positions simultaneously. On the other hand, the prey moves either randomly or at a slower speed than the predators, thus allowing the predators to catch up with it easily.

In contrast, in the experiments described in this paper it is enough for one predator to move onto the prey to capture it. However, the prey moves as fast as the predators, and always away from the nearest predator, and therefore there is no way to catch the prey simply by chasing it. The main challenge is in coordinating the chase: The agents have to approach the prey from different directions so that it has nowhere to go in the end. This behavior requires developing a long-term cooperative strategy, instead of coordinating the timing of a few actions accurately, and therefore makes it possible to identify a class of tasks where efficient cooperation emerges without communication.

III. Methods

In this section, the implementation of the prey-capture task and the ESP and Multi-Agent ESP methods are described, followed by the details of the experiments.

A. Prey-Capture Implementation

The prey-capture task in this paper consists of one prey and three predators in a discrete toroidal environment (Figure 2). The prey is controlled by a rule-based algorithm; the predators are controlled by neural networks. The goal is to evolve the neural networks to form a team for catching the prey. The different approaches and techniques are compared based on how long it takes for the team to evolve to catch the prey consistently, and what kind of strategies they use.

The environment is a 100×100 toroid without obstacles or barriers (the 100×100 area is also referred to as the "world" below). All agents can move in four directions: N, S, E, or W. The prey moves as fast as the predators, and always directly away from the nearest predator. It starts at a random location of the world, and the predators start in a horizontal row at the bottom left corner (Figure 2a). All the agents make their moves simultaneously, and an agent can move into a position occupied by another agent. The team catches the prey when a predator moves onto the position occupied by the prey. If the predators have not caught the prey in 150 moves, the trial is terminated and counted as a failure.

Constrained in this way, it is impossible to consistently catch the prey without cooperation. First, since the predators always start at the bottom left corner, behaving greedily would mean that they chase the prey as a pack in the same direction. The prey will then avoid capture by running away in the same direction: Because it is as fast as the predators, and the environment is toroidal, the predators will never catch it (Figure 18 demonstrates this scenario). On the other hand, should the predators behave randomly, there is little chance for them to approach, circle, and run into the prey. The 150 steps limit is chosen so that the predators can travel from one corner of the world to the other, i.e. they have enough time to move to surround the prey, but it is not possible for them to just mill around and eventually capture the prey by accident.

B. Neuroevolution Implementation

Three approaches to evolving and controlling agents will be tested: the central controller approach, the autonomous communicating approach, and the autonomous non-communicating approach. In the central controller approach, all three predators are controlled by a single feedforward neural network, implemented with the usual ESP method (Figure 1).

For the communicating and non-communicating autonomous controllers, Multi-Agent ESP will be used. This method extends the subpopulation idea to the level of networks (Figure 3). Each predator is controlled by its own feedforward network, evolved simultaneously in separate populations. Each network is formed using the usual ESP method. These three networks are then evaluated together in the task as a team, and the resulting fitness for the team is distributed among the neurons that constitute the three networks.

Before running the comparisons, an appropriate number of hidden units was determined for each of the three approaches. Since small networks typically generalize better and are faster to train [28, 43], the smallest number of hidden units that allowed solving the task reliably was found. More specifically, for each of the three approaches, ten evolution runs were performed on the prey-capture task, initially with two hidden units. When any of the ten runs failed to solve the task completely, the number of hidden units was increased by one and another ten runs were tried. A run was deemed a failure if it stagnated ten consecutive times, that is, if its fitness did not improve despite ten burst mutations in 250 generations. Through this procedure, an appropriate number of hidden units was determined to be nine for the central controller, eight for each of the autonomous communicating controllers, and three for each of the autonomous non-communicating controllers (Figures 4–6). The comparisons were run with these architectures.

In all three approaches, the agents are evolved in a series of incrementally more challenging tasks. Such an incremental approach is motivated by natural evolution and shaping of

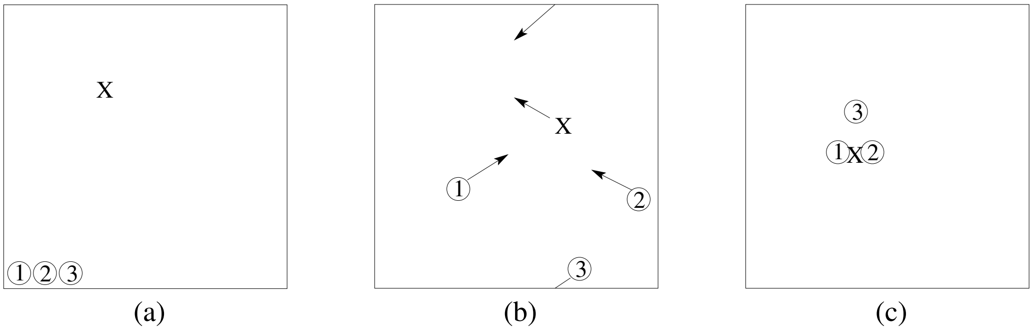

Fig. 2. **The Prey-Capture Task.** The environment is a 100×100 toroidal grid, with one prey (denoted by "X") and three predators (denoted by "1", "2" and "3"). Figure (a) illustrates a starting scenario: The predators start in a row at the bottom left corner, and the prey starts in a random location. Figure (b) illustrates a scene later during a trial. The arrows indicate a general direction of movement: Since each agent may only move in the four cardinal directions, a movement arrow pointing 45 degrees northwest means the agent is moving north and west on alternate time steps. Figure (c) shows the positions of the predators one time step before a successful capture. The prey always moves directly away from the nearest predator; even though it is as fast as the predators, if the predators approach it consistently from different directions, eventually the prey has nowhere to run.

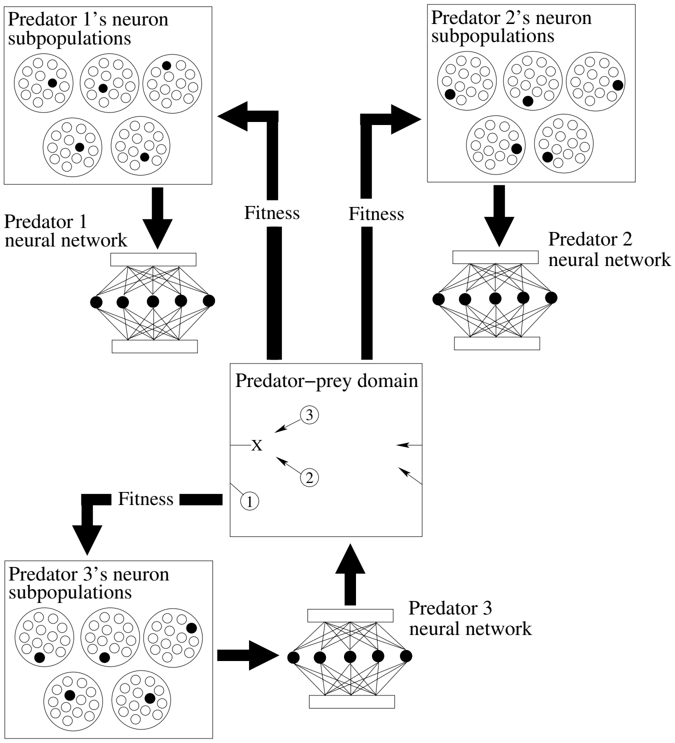
Predator 1's neuron subpopulations
Predator 2's neuron subpopulations
Fitness
Fitness
Predator 1 neural network
Predator 2 neural network
Predator−prey domain
X
1
2
3
Fitness
Predator 3's neuron subpopulations
Predator 3 neural network

Fig. 3. **The Multi-Agent ESP Method.** Each predator is controlled by its own neural network, formed from its own subpopulations of neurons. The three neural networks are evaluated in the task at the same time as a team, and the fitness for the team is passed back to all participating neurons.

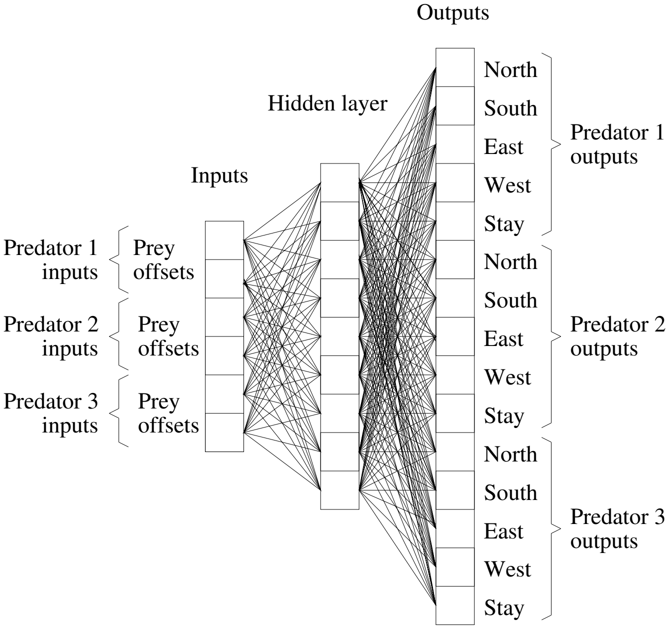

Fig. 4. **Central controller network for a team of three predators.** This network receives the relative x and y offsets (i.e. relative distance) of the prey from the perspective (i.e. location) of all three predators, and outputs the movement decisions for all three predators. This way it acts as the central controller for the whole team. There are nine hidden units, and the chromosomes for each hidden layer unit consist of 21 real-valued numbers (six inputs and 15 outputs).

animal behavior, where complex behaviors gradually emerge through a series of incremental challenges [9, 49]. It also facilitates computational evolution in complex tasks, where direct evolution in the goal task might otherwise take a long time, or result in inadequate, mechanical strategies such as running around in circles [15, 56]. One way to establish incremental evolution would be to coevolve the prey with the predators [53]. However, in order to make evolutionary progress transparent and the results comparable, a fixed set of tasks is used in this paper. Evolution proceeds through six stages: In the easiest task the prey is stationary, and in each subsequent task it moves at a faster speed and with a greater probability of heading away from the nearest predator, until in the final task it moves as fast as the predators, and always away from the nearest predator. Table I gives the speeds and evasive probabilities of the prey for each of these tasks; small variations to this schedule lead to similar results. When a team manages to solve the current task consistently, the next harder task is introduced. The team can thus utilize what it has already discovered in the easier task to help guide its evolution

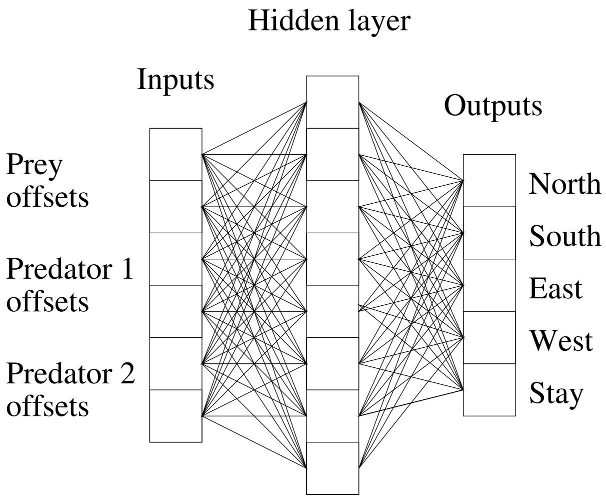

Fig. 5. **Controller for each autonomous communicating predator.** This network autonomously controls one of the predators; three such networks are simultaneously evolved in the task. The locations of this predator's teammates are obtained, and their relative x and y offsets are calculated and given to this network as information obtained through communication. It also receives the x and y offsets of the prey. There are eight hidden units, and the chromosomes for each hidden layer unit consist of 11 real-valued numbers (six inputs and five outputs).

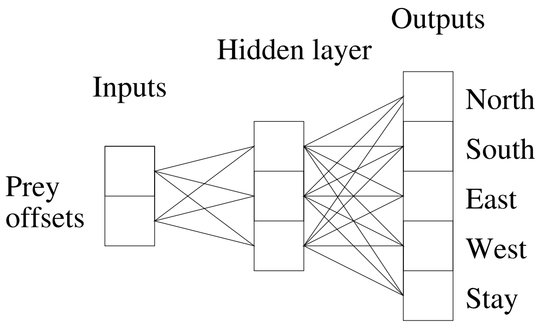

Fig. 6. **Controller for each autonomous non-communicating predator.** This network receives the prey's x and y offsets as its inputs. Therefore, it controls a single predator without knowing where the other two predators are (i.e. there is no communication between them). There are three hidden units, and the chromosomes for each hidden layer unit consist of seven real-valued numbers (two inputs and five outputs).

in the new, harder task. In the prey-capture task, incremental evolution is particularly useful because it allows the predators to discover early on how to catch the prey at close proximity. Placing the predators into the final task right from the start fails because they do not get close to the prey often enough to discover how to catch it. The incremental approach is therefore used to give evolution more experience with the necessary skills that would otherwise be hard to develop.

The fitness function consists of two components, depending on whether the prey was captured or not:

$$f = \begin{cases} \frac{d_0 - d_e}{10} & \text{if the prey was not caught} \\ \frac{200 - d_e}{10} & \text{if the prey was caught,} \end{cases}$$

where d_0 is the average initial distance of the predators from the prey, and d_e is the average final distance. This fitness function was chosen to satisfy four criteria:

1) If the prey is caught, the starting scenario (i.e. the initial distance from the prey) should not bias the fitness. Instead, teams should be rewarded if their ending positions are good—that is, if all predators are near the prey.
2) If the prey is not caught, teams that covered more distance should receive a higher reward.
3) Since a successful strategy has to involve sandwiching the prey between two or more predators, at least one predator must travel the long distance of the world so that two predators can be on the opposite sides of the prey. Thus the time taken for each capture (within the 150 step limit) tends to be about the same, and should not be a factor in the fitness function.
4) The fitness function should have the same form throughout the different stages of incremental evolution, making it simple and convenient to track progress.

The neuron chromosomes are concatenations of the real-valued weights on the input and output connections of the neuron (Figure 1). As is usual in ESP, burst mutation through delta-coding [62] on these weights is used as needed to avoid premature convergence: If progress in evolution stagnates (i.e. the best solution 25 generations earlier outperforms the current best solution), the populations are re-initialized according to a Cauchy distribution around the current best solution. Burst mutation typically takes place in prolonged evolution in difficult tasks [15, 16].

C. Experimental Setup

In each experiment in this paper, the different approaches were each run ten times with different random initial populations. Each run consisted of several generations, until success or maximum number of generations was achieved. The results were averaged across the runs and compared statistically (through Student's paired two-tailed t-test with $\mathrm{df} = 9$).

The following parameter settings were used for ESP and its multi-agent extension. Each subpopulation of neurons consisted of 100 neurons; each neuron (or chromosome) was a concatenation of real-valued numbers representing full input and output connections of one hidden unit. During each evolutionary generation, 1000 trials were run wherein the neurons were randomly chosen (with replacement) from their subpopulations to form the neural networks. In each trial, the team was evaluated nine times (to match the number of evaluations in the test benchmark suite to be described shortly). Unlike in the benchmark suite, the prey started in a random location in each of the nine evaluations. The predators always started in the bottom-left corner (Figure 2a), giving the different trials a common structure that makes it easier to analyze and compare results (Section VI-C shows that similar results are obtained when predators start at random locations). The fitnesses over the nine evaluations were averaged, and assigned to all the neurons that constituted the network. Thus, each neuron was evaluated as part of ten different networks on average, and each generation consisted of 9000 total network evaluations. After the trials, the top 25% of neurons in each subpopulation were recombined using one-point crossover.

Task	Prey Speed (Probability of Moving)	Prey Probability of Moving Away from Nearest Predator
1	0	0
2	0.45	0.45
3	0.63	0.63
4	0.77	0.77
5	0.89	0.89
6	1	1

TABLE I
THE SEQUENCE OF INCREMENTALLY MORE DIFFICULT TASKS.

The offspring replaced the bottom 50% of the neurons, and they were then mutated with a probability of 0.4 on one randomly-chosen weight on each chromosome, by adding a Cauchy-distributed random value to it. These parameter values were found effective in preliminary simulations, but the results are not highly sensitive to them.

The environment is stochastic only in the prey's starting location, and this location is the only factor that determines the course of action taken by the predators. In order to test these team strategies comprehensively, a suite of benchmark problems was implemented. The lower left 99×99 part of the world was divided into nine 33×33 subsquares. In each trial, each team was tested nine times, with the prey starting at the center of each of these subsquares in turn. Such an arrangement provides a sampling of the different situations, and allows estimating how effective each team is in general. A team that manages to catch the prey in all nine benchmark cases is considered to have completely solved the task, and indeed such a team usually has a 100% success rate in random, general scenarios.

Communication between predators was modeled by giving the locations of the other predators as input to each predator's neural network. The idea is that this transmission of information takes place on purpose, i.e. each predator broadcasts its location in order for the other predators to pick it up. In contrast, even though the predators also receive the location of the prey as their input, the prey does not make it available on purpose. Instead, the idea is that the predators use sensors to obtain this information. (To make this scenario concrete, imagine a situation where the pursuit occurs in a dark environment where the predators can move silently but the prey moves with a sound. The prey can be located based on that sound, and the predators can then use vocalizations to transmit their location at will.) According to the definition adopted in section II-C, the predators' locations are communicated, whereas the prey's location is not. Therefore, the controllers that receive predators' locations are called communicating controllers, and those that do not (even though they obtain the same information through stigmergy) are called non-communicating controllers in this paper.

IV. EVOLUTION OF COOPERATIVE BEHAVIOR

In this section, two baseline experiments are presented, testing the two main hypotheses of this paper: First, that cooperative coevolution of autonomous controllers is more effective than evolving a central controller in this task (Section IV-A), and second, that the agents controlled by autonomous neural networks can evolve to cooperate effectively without communication using stigmergy (Section IV-B). These behaviors and conditions under which they arise are then analyzed in more detail in Sections V and VI.

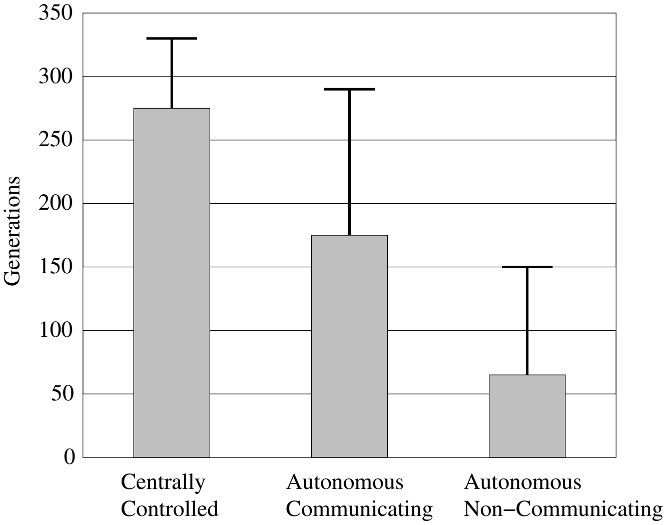

Fig. 7. **Evolution performance for each approach.** The average number of generations, with standard deviation, required to solve the task is shown for each approach. The centrally controlled team took 50% longer than the autonomously controlled communicating team, which in turn took over twice as long as the autonomously controlled non-communicating team, to evolve a successful solution. All differences are statistically significant ($p < 0.05$).

A. Standard Evolution of a Central Controller vs. Cooperative Coevolution of Autonomous Controllers

This section tests the first hypothesis, i.e. that it is easier to coevolve three autonomous communicating neural networks, each controlling a single predator (Figure 5), than it is to evolve a single neural network that controls the entire team (Figure 4). The number of evolutionary generations needed to solve the task, that is, to evolve a team able to catch the prey in all nine benchmark cases, are compared for the two approaches.

Figure 7 shows a clear result: On average, the three autonomous controllers were evolved almost twice as fast as the centralized controller. The conclusion is that the cooperative coevolution approach is more powerful than the centralized approach in this task. (The results in Section VI-A further confirm that this result does not simply follow because the networks have different sizes.)

Figure 8 shows how long it took each approach to solve each incrementally more difficult task during evolution. While the

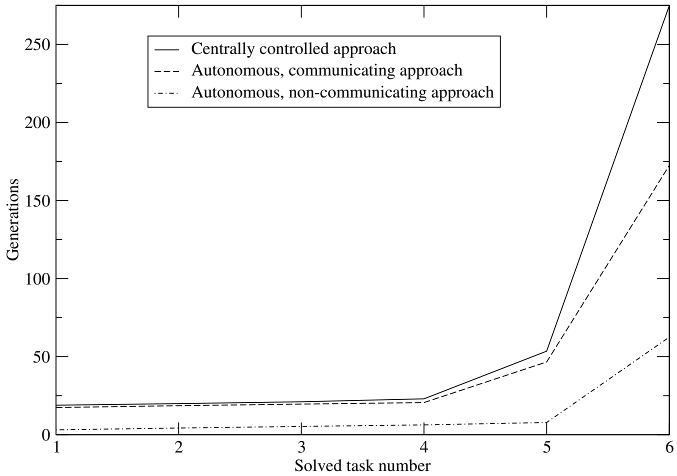

Fig. 8. **Progress of evolution through incrementally more difficult tasks.** Number of generations required for each approach to solve each task in the sequence is shown. As shown in Table 1, in Task 1 the prey is stationary, whereas in Task 6 it moves at the same speed as the predators, and always away from the nearest predator. The centrally controlled teams did not take statistically significantly longer than the autonomously controlled communicating teams to solve the easier tasks. With more difficult tasks (i.e. task 6), the differences became statistically significant. On the other hand, the autonomously controlled non-communicating teams solved all the tasks significantly faster than either of the two other approaches.

centrally controlled teams require just slightly more time to solve the easier tasks than the autonomous controllers, as the tasks become more difficult, the differences in performance grow significantly. This result suggests that the autonomous controller approach is most useful with harder tasks.

B. *Cooperative Coevolution With vs. Without Communication*

The conclusion from the first comparison is that separating the control of each agent into disjoint autonomous networks allows for faster evolution. The controllers no longer receive direct information about what the other agents see; however, the domain is still completely represented in each predator's inputs, which include the relative locations of the teammates and the prey. In this section the available information is reduced further by preventing the predators from knowing each other's locations. This way the agents will have to act independently, relying on stigmergy for coordination. The objective is to test the second hypothesis, i.e. that cooperation based on stigmergy may evolve more efficiently than cooperation based on communication.

The network architecture for such non-communicating controllers is shown in Figure 6. The predator no longer receives the relative x and y offsets of the other predators, only the offsets of the prey. These networks were evolved with the same coevolutionary Multi-Agent ESP method as the communicating networks of Figure 5.

The non-communicating teams solved the entire task more than twice as fast as the communicating teams (Figure 7). Furthermore, the non-communicating teams solved each incrementally more difficult task significantly faster than the communicating teams (Figure 8).

These results show that communication between teammates is not always necessary: Cooperative behavior can emerge even when teammates do not receive each other's locations as input. In fact, since communication is not necessary, it is more efficient to do away with it entirely. In the next section, examples of evolved behaviors will be analyzed to gain insight into why this is the case, concluding that the agents rely on stigmergy. In Section VI, a series of further simulations will be presented to demonstrate that this result is robust against variations in the architecture and the task.

V. Analysis of Evolved Behaviors

In this section, the behaviors evolved using the three approaches are analyzed in two ways: First, the degree to which each predator's actions depend on those of the other predators is measured quantitatively. Second, examples of evolved behaviors are analyzed qualitatively. This analysis leads to the characterization of team behaviors in two extremes: Those based on role-based cooperation, where interaction between different roles takes place through stigmergy, and those based on communication.

A. *Measuring Dependency*

A predator's actions are independent of its teammates if the predator always performs the same action for the same prey position, regardless of where its teammates are. Conversely, the predator's strategy is dependent on its teammates if its actions are determined by both the prey's and the teammates' positions.

The team of autonomous, non-communicating predators must act independently: Since the neural network for each predator cannot see the locations of its teammates, their positions cannot affect its actions. A more interesting question is whether dependent actions evolve in the communicating team. The agents could evolve to simply ignore their communication inputs, or they could evolve to use communication to develop a coordinated strategy. Also, it is interesting to observe whether the actions of a centrally controlled team are more dependent than those of an autonomous communicating team, since its control architecture is more tightly coupled between teammates.

Action dependence can be measured in the following way: For each predator in the team, a sample is taken of possible relative prey positions. For each of these sampled positions, possible configurations of teammates' locations are then sampled. Each such case is presented to the predator's control network, and the predator's resulting action observed. The percentage of actions that differ for the same prey position (but different teammate positions) is a measure of how dependent this predator's actions are on its teammates. The team's dependence is obtained as the average of those of its three predators.

The dependencies of the centrally controlled, communicating, and non-communicating teams are compared in Figure 9. As expected, the non-communicating predators act independently. In contrast, over 90% of the actions of the centrally

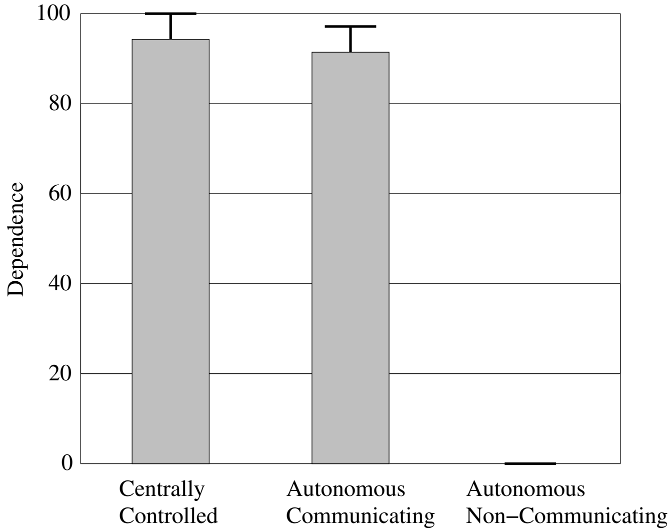

Fig. 9. **Action dependence in teams evolved using each approach.** The centrally controlled and autonomous communicating teams both evolved behaviors in which each agent's actions were highly dependent on its teammates positions (the difference is statistically insignificant, with $p = 0.16$). The non-communicating team's evolved behaviors were independent (the difference between the dependence of the non-communicating team and those of the other two approaches is statistically significant, $p < 0.01$).

controlled and communicating team members depend on the other predators. On average, the centrally controlled teams were slightly more dependent than the communicating teams; however, this difference is not statistically significant.

These results demonstrate that indeed the communicating teams evolve a distinctly different behavior from the non-communicating teams. Even though the weights on the communication inputs could evolve to zero, they do not. Apparently, given a starting point where all weights are random, it is easier for evolution to discover an adequate communicating strategy than to un-evolve communication altogether. An interesting issue is whether a different starting point would bias evolution to discovering non-communicating solutions instead. This question is addressed in Section VI-D.

B. Characterizing Sample Behaviors

Dependence measurements demonstrate that the behaviors differ, but to understand how, actual examples need to be analyzed. In this section, example behaviors for each approach are described and compared in detail. The main results are that without communication, evolution produces specific roles for each team member, and these roles interact only indirectly through stigmergy, that is, by causing changes to the environment that affect the other teammates' roles; furthermore, these teams utilize a single effective strategy in all cases. On the other hand, evolution with communication produces agents with more varied (although less effective) behaviors, able to employ two or more different strategies at different times.

A typical successful strategy for the non-communicating team is illustrated in Figure 10. This strategy is composed of two stages, the *setup* stage and the *chase* stage. In the setup stage, illustrated in the first two frames, the predators maneuver the prey into an appropriate configuration for a chase: In frame one, predators 2 and 3 move eastward, causing the prey to flee in the same direction, while predator 1 moves westward. When the prey detects that predators have closed in to its south, it starts fleeing northward (frame two). In frame three, the predators detect that the prey is directly to their north, and the chase stage of the strategy begins. This stage involves two different roles, *chasers* and *blockers*. Predator 1, the blocker, moves only in the horizontal direction, staying on the same vertical axis as the prey, while predators 2 and 3, the chasers, pursue the prey northward (frames three and four). Eventually, the prey is trapped between the blocker and the chasers, who move in for the capture. Notice that this strategy requires no communication between predators. As long as the predators get the prey into a chase configuration, and the blockers and chasers execute their roles, the prey will always be caught. Moreover, it is reliable for all starting locations of the prey, even when the predators do not all switch from setup to chase modes at the same time.

From the above description, it is clear that some form of coordination is taking place, even without communication. When the prey is not directly above (or below) a predator, the predator is in setup mode, moving either east (predators 2 and 3), or west (predator 1). This action causes a change in the environment, i.e. a specific reaction on the part of the prey: to flee east or west. Because of this reaction, each of the predators will eventually find the prey directly above or below it, triggering a second activity: They either chase the prey north (predators 2 and 3), or remain on the prey's vertical axis (predator 1). This activity in turn causes a change in the prey's behavior, to flee north, until it is eventually trapped between chasers and blockers. This sequence of events is a clear example of stigmergy: Each agent's action causes a change in the environment, that is, a reaction on the part of the prey; this reaction in turn causes a change in the agent's and its teammates' behaviors.

Sample behavior of a communicating team is illustrated in Figure 11. Two different strategies are shown because this team actually displays both of them, and also their combinations and variations, depending on the relative locations of the prey and the other predators at each timestep. The first strategy, in Figure 11a, illustrates behavior similar to the chaser-blocker strategy. The first frame is a snapshot of the starting position. Predators 1 and 2 are the chasers, and they start pursuing the prey upward. Predator 3 is the blocker, and it moves left onto the prey's vertical axis. At this point, however, it starts chasing the prey downward, in Frame 2, until the prey is trapped between all three predators in Frame 3. Already this strategy is more varied than those of the non-communicating teams, as a combination of blocking and opposite chasers.

Another strategy employed by the same team in a different situation is shown in Figure 11b. In the first frame, predators 1 and 3 start moving toward the prey diagonally upward and downward, while predator 2 moves upward until it is horizontal with the prey. By the second frame, predators 1 and 3 are chasing the prey horizontally, until it is trapped between them and predator 2 in frame 3. This strategy is again similar to the chaser-blocker strategy, except this time

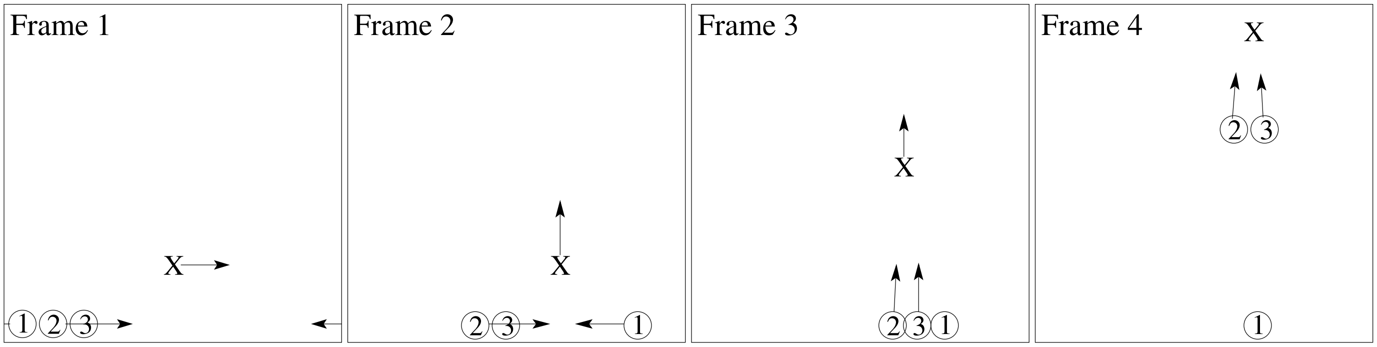

Fig. 10. **A sample strategy of a non-communicating team.** In frames 1 and 2, the predators are in setup mode, maneuvering into an appropriate chase configuration. In frame 3, they switch to chase mode: Predators 2 and 3 chase the prey toward predator 1, which acts as a blocker. This strategy is effective and does not require communication. Animated demos of this strategy, and others discussed in this paper, are available at http://nn.cs.utexas.edu/?multiagent-esp.

the prey is chased horizontally instead of vertically, and the chase includes diagonal movement as well.

Although each strategy is similar to those of non-communicating teams, in this case they are employed by one and the same team. This team occasionally also utilizes combinations of these strategies, for example by starting with one and finishing with the other. Thus, each predator does not have a specific, fixed role, but modifies its behavior depending on the situation. Through communication each predator is aware of its teammates' relative locations, and its behavior depends not only on the prey's relative position, but also directly on what the other predators are doing. In this way, the communicating strategies are more varied. On the other hand, they are less efficient to evolve (Section IV-A), and less robust (Section VI-B). In a sense, the non-communicating teams resemble players in a well-trained soccer team, where each player knows what to expect from the others in each play, whereas the behavior of the communicating teams is similar to a pickup soccer team where each player has to constantly monitor the others to determine what to do. Such players can perhaps play with many other kinds of players, but not as efficiently.

The centrally controlled teams exhibit behavior nearly identical to those of the communicating autonomous teams. In particular, the more tightly coupled architecture does not translate to behavior that is visibly more coordinated in this task.

Of course we have to be careful not to attribute undue intelligence to neural networks that simply manage to adapt to each other's behavior. However, the differences in behavior are striking: The non-communicating teams employ a single, efficient, fail-proof strategy in which each team member has a specific role, while the communicating and centrally controlled teams employ variations and combinations of two or more strategies. These two forms of cooperative behavior can be distinguished as *role-based cooperation* and *communication-based cooperation*. Role-based cooperation consists of two ingredients: Cooperation is achieved through the combination of the various roles performed by its team members, and these roles are coordinated indirectly through stigmergy. On the other hand, communication-based cooperation may or may not involve roles performed by its team members, but the team members are able to coordinate through communication with each other.

In the following section, these cooperative strategies will be characterized further by testing how robustly they evolve and how robustly the evolved agents perform under various extreme conditions.

VI. Robustness of Strategy Evolution and Performance

Section IV showed that when minimal neural-network architectures are used for each approach, the coevolutionary approach evolves faster than the centralized approach, and the team without communication evolves faster than the team with communication. This section presents a number of control simulations to verify that these results are robust. First, it is important to verify that they hold when equivalent neural-network architectures are used across all approaches. Also, since the evolved non-communicating strategies include fixed roles, it is necessary to demonstrate that they are robust against changes in the prey's behavior. It is also important to show that evolution can discover non-communicating strategies robustly even when the predators are initially placed randomly. Another interesting issue is whether communicating teams can be biased by design to evolve role-based, rather than communication-based cooperative behavior. Finally, it is important to demonstrate that coevolution of heterogeneous agents is indeed necessary to solve the task. Each of these issues is studied in a separate experiment in this section.

A. Do the results hold across equivalent network architectures?

When the network architectures were optimized separately for each approach, the coevolutionary approach solved the task faster than the centralized approach (Section IV-A), and teams without communication faster than teams with communication (Section IV-B). However, it is unclear how much of this result is due to the different search-space sizes (i.e. different number of weights that need to be optimized), and how much is due to the centralized vs. distributed control strategy itself, or to the availability of communication.

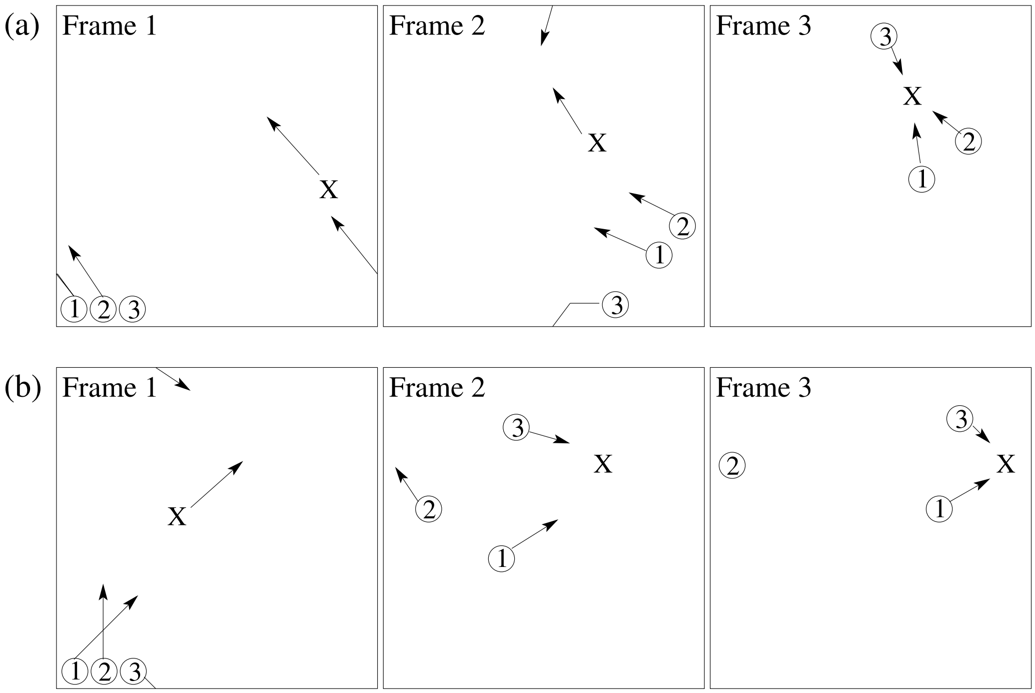

Fig. 11. **Two sample strategies of the same communicating team.** This team employs the two strategies shown above, as well as their variations and combinations. In the first, (a), the chase starts with two chasers and a blocker, but ends with opposite chasers. In the second, (b), there is a blocker and two chasers throughout, but the movement is horizontal. In this manner, the same team utilizes different strategies, depending on the starting position of the prey.

In the centralized approach, the neural network had nine hidden units and 21 connections per unit for a total of 189 weights (Figure 4), whereas the network used in the coevolutionary approach (with communication) had eight hidden units and 11 connections per unit for a total of 88 weights (Figure 5). Such a smaller search space may allow evolution to progress significantly faster. The same issue arises in comparing communicating vs. non-communicating networks: The latter approach has three hidden units and seven connections per unit for a total of 21 weights (Figure 6).

Of course, such small search spaces are possible precisely because the approaches are different, but it is still interesting to verify that the results are not completely due to search complexity. To this end, the experiments in Sections IV-A and IV-B were repeated with identical network architectures across all approaches. Specifically, all approaches use the architecture of the centralized approach, with nine hidden units, six inputs, and 15 outputs. When some inputs are not part of the approach (such as teammate locations in the non-communicating approach), random noise values are used for them. Outputs that are not used are simply ignored (Figures 12). In this manner, each approach receives and generates the same information as before, but now within a uniform architecture. The evolution performance of the coevolutionary approaches with such architectures is compared against that of the original centralized approach, providing a comparison of the three different approaches given a uniform search space.

Figure 13 presents the results for these experiments. Using larger neural networks did not significantly change the number of required generations to solve the task for either of the coevolutionary approaches (i.e. with or without communication). As a result, the relative differences in evolution time between the three approaches are preserved. Therefore, the better performance of the coevolutionary over the centralized approach, and that of the coevolutionary non-communicating over the coevolutionary communicating approach, are indeed due to the approaches themselves, and not simply a consequence of being able to search in a smaller space.

B. Are the strategies robust against novel prey behavior?

Although the non-communicating networks work together like a well-trained soccer team, soccer (like most interesting real-world tasks) is unpredictable. For example, a player from the other team may intercept a pass, in which case the team members will have to adapt their strategy quickly to cope with the new situation. To determine how the non-communicating team can deal with such unpredictability, three further experiments were conducted where the non-communicating teams were pitted against a prey that behaved differently from those encountered during evolution. For comparison, the same tests were also run for the communicating teams. Since the non-communicating teams' predators act according to fixed roles, they might not be able to adapt as well as the communicating teams' apparently more flexible agents.

The first experiment presented the agents with a more challenging version of the original behavior: The prey moved three times faster than usual (by moving three steps each time) and in a random direction 20% of the time. The second test was similar but harder: The prey moved three times faster than usual and in a random direction 50% of the time. In the third

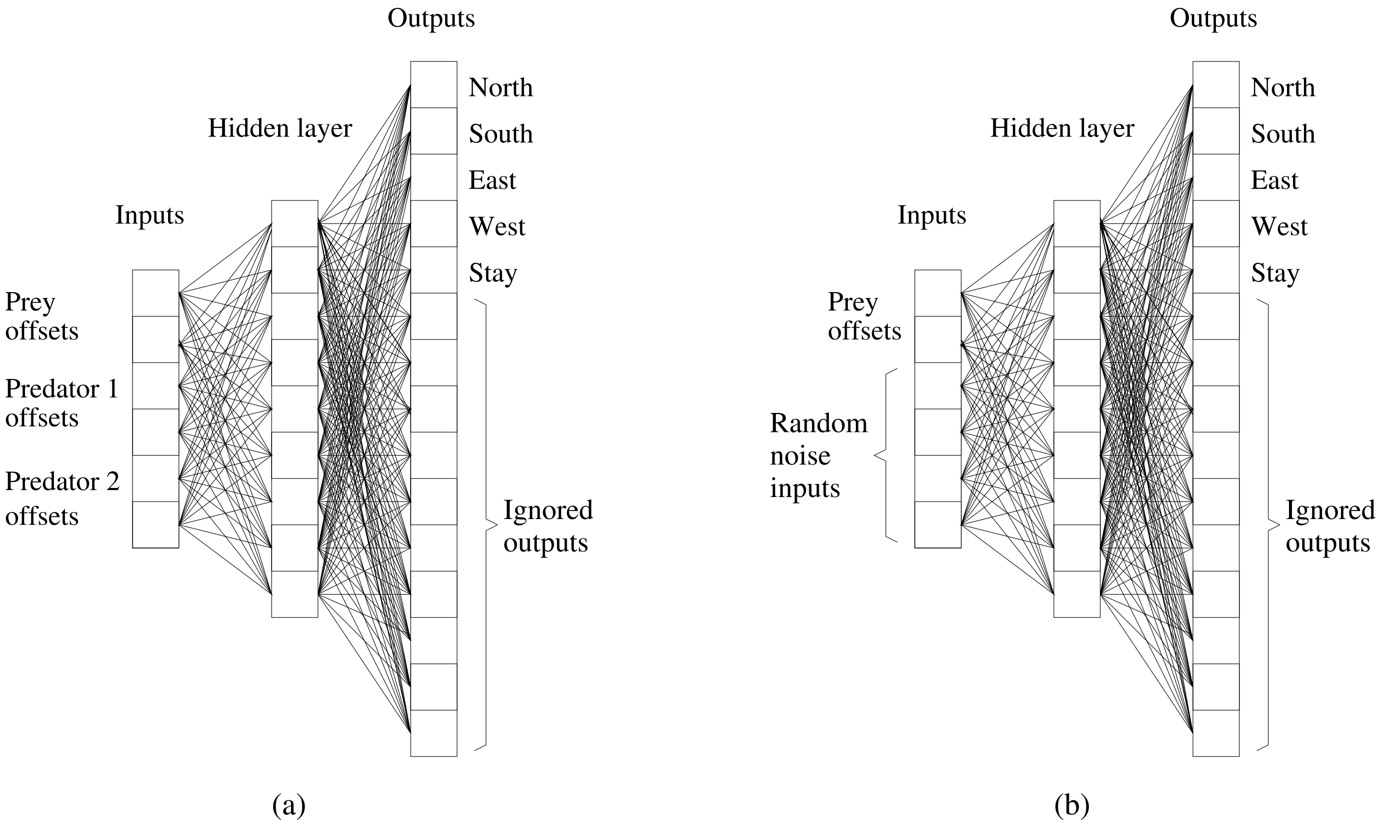

Fig. 12. **The large-network versions of the communicating and non-communicating controllers.** (a) The network used in the autonomous communicating approach; ten of the outputs are ignored. (b) The network used in the autonomous non-communicating approach; ten of the outputs are ignored, and four inputs are fed random noise. In both cases, there are nine hidden units, and the chromosomes for each hidden layer unit consist of 21 real-valued numbers (six inputs and 15 outputs), which is the same as in the central-controller approach.

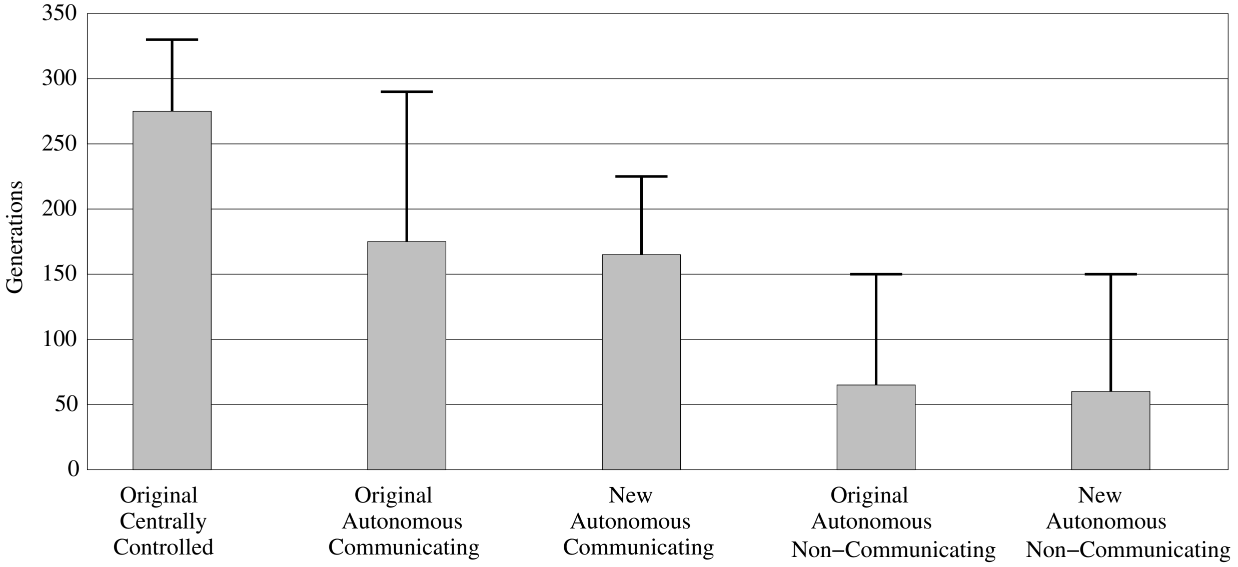

Fig. 13. **Evolution performance for each approach with equivalent network architectures.** The average number of generations, with standard deviation, required to solve the task is shown for each approach. The performance is about the same as with minimal architectures, and the relative performance between approaches is preserved. The differences between the three approaches are statistically significant ($p < 0.05$).

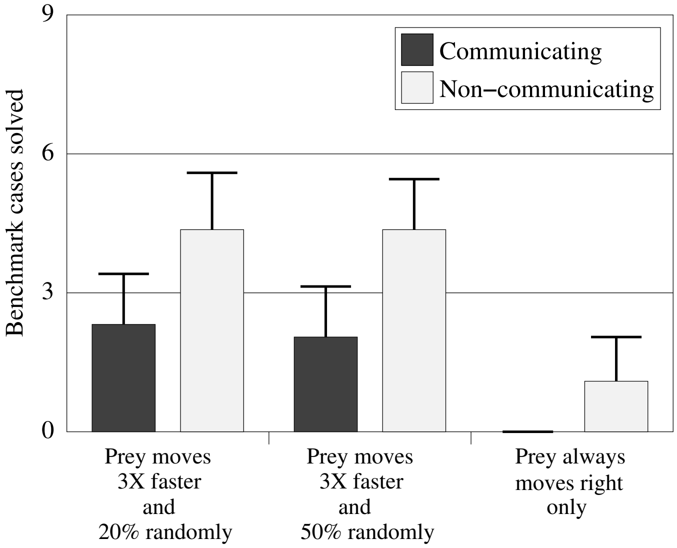

Fig. 14. **Robustness of communicating and non-communicating teams against novel prey behavior.** In the first test, the prey moved three steps each time and in a random direction 20% of the time; in the second, three steps and randomly 50% of the time; and in the third, always right. Surprisingly, the non-communicating teams performed significantly better than the communicating teams ($p < 0.01$ for all three tests), even though the communicating strategies are generally more varied. The non-communicating teams tolerate the occasional novel behavior well as long as their basic strategy is valid. However, even they cannot cope if the prey employs a consistently different strategy.

experiment, the prey exhibited a distinctly different strategy from the original behavior: It moved at the same speed but always to the right.

The results, summarized in Figure 14, are surprising: The non-communicating teams are more robust against unpredictable prey than the communicating teams. Apparently, the first two prey behaviors are still familiar enough so that the fixed roles are effective: The teams still catch the prey about 50% of the time. The agents only have to track the occasional erratic movement, otherwise their strategy is effective as is, even when the prey is substantially faster. The communicating teams, however, have a narrower range of adaptable situations, particularly because their agents tend to switch strategies and roles based on the current state of the world, and thus get easily confused by the unexpected prey actions. In the third case, where the prey always moves right, neither team is able to track it well. This behavior is consistently novel, and the agents are evolved not to expect it.

In sum, teams that have delegated specific and fixed roles to their members are more tolerant of noisy or unusual situations, as long as their basic strategy is still valid.

C. Are the strategies robust with random deployment?

In all experiments so far, the predators always started at the bottom left corner of the world, and only the prey's initial position was varied. Such a limitation makes it possible to analyze the resulting behaviors systematically. It is important to demonstrate that similar behavior evolves also in the more general case where the predators' positions are initially random. Coordination is more difficult then because the agents first need move into proper positions for the chase. The issue is, will the non-communicating networks be able to establish effective roles even then?

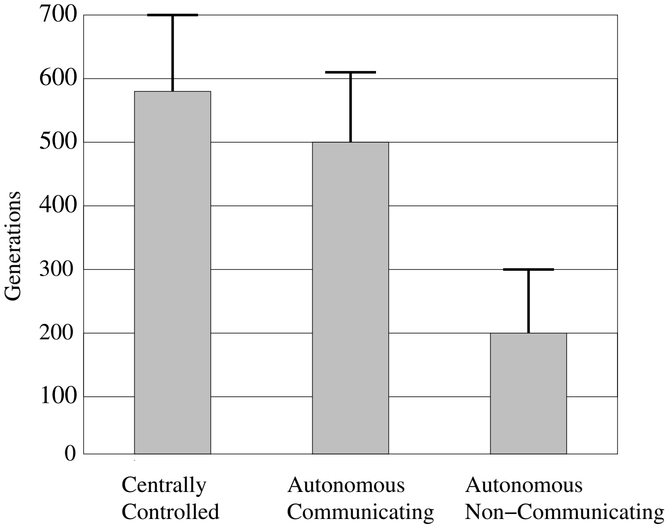

Fig. 15. **Evolution performance of the three approaches when the predators start at random locations.** In this harder task, evolution took longer than with fixed initial placement of predators in all cases. Moreover, as before, the non-communicating teams were significantly easier to evolve than communicating teams ($p < 0.01$), which were slightly easier than central controllers ($p < 0.1$). The team behaviors were also similar to those evolved earlier.

The experiment was set up as described in Section III-C, except that predators were placed randomly in the world during both evolution and benchmark tests. Furthermore, the number of evaluations per trial during evolution was increased from nine to 45, in order to obtain a sufficient sample of the different starting positions (45 was found to be sufficient experimentally). The number of evaluations during benchmark tests was increased from nine to 900 for the same reason (whereas relatively sparse sampling is sufficient to guide evolution, the benchmarks need to measure performance accurately). A network was deemed successful if it caught the prey in 750 of the 900 tests.

The minimum effective network sizes were determined as in Section III-C. In this more challenging task, they were found to be 13 hidden units for the centrally controlled and communicating autonomous networks, and five hidden units for the non-communicating networks.

Figure 15 shows how long it took each approach to solve the task on average. Although this task was much harder, the conclusions are the same as before: The communicating autonomous teams were easier to evolve than central controllers, and the non-communicating networks easier than the communicating networks. The strategies evolved were also similar to those with fixed initial placement of predators.

D. Can evolution be biased toward role-based cooperation?

As discussed in Section V, the communicating networks could in principle evolve non-communicating behavior by setting weights to zero on the connections from those input units that specify the other predators' positions. It does not happen, and the reason may be that it is difficult for evolution to turn off all such connections simultaneously. It is easier to discover a competent communicating solution instead, utilizing those inputs and weights as well.

An interesting question is whether evolution would discover role-based cooperation if it was biased to do so from the beginning. Such an experiment was run by setting all the communicating weights to zero in the initial population; the other weights were initialized randomly as usual. Evolution only had to keep the communicating weights at zero while developing the rest of the network to solve the task. As a comparison, networks with all weights initially at zero were also evolved, and the results compared to normal evolution of communicating networks.

The action dependencies that evolved are shown in Figure 16. The results show that such an initial bias does have an effect: Evolution discovers behaviors where the actions depend significantly less often on the positions of other predators. With communication weights initially at zero, 14% of the actions depend on them, and with all weights initially zero, 48%. In contrast, the earlier evolved networks with all randomly-initialized weights exhibit 91% dependency. Qualitatively the behaviors of the initially biased networks consist mostly of role-based cooperation, with occasional switches or deviations in the roles of the predators.

The performance in terms of number of generations needed to solve the task is shown in Figure 17. Teams with communication weights initialized to zero evolve significantly faster: The number of generations required, compared to the normal communicating teams, dropped by a factor of three, making it as fast as the non-communicating teams. When all weights are initialized to zero, the improvement in evolution time was found to be insignificant.

These results show that evolution can indeed discover role-based cooperation, and the number of generations required can be comparable to that needed by an architecture that forces role-based behavior (i.e. the non-communicating networks). However, the initial state needs to be biased the right way, and the role-based cooperation discovered may not be perfect. In other words, it is still important for the designer of a multi-agent system to recognize whether role-based cooperation could work in the task, and utilize the appropriate architecture to evolve it.

E. Is Coevolution Necessary?

Although the performance of cooperative coevolution looks convincing, it does not necessarily mean that coevolution is essential for the task. Perhaps it is possible to evolve good predators individually, and just put them together to solve the task? This subsection demonstrates experimentally that such an approach is not sufficient: They do not evolve cooperative behavior like they do with coevolution.

A single predator without communication inputs (as shown in Figure 6) was evolved alone incrementally in the prey-capture task, using the standard ESP method as described

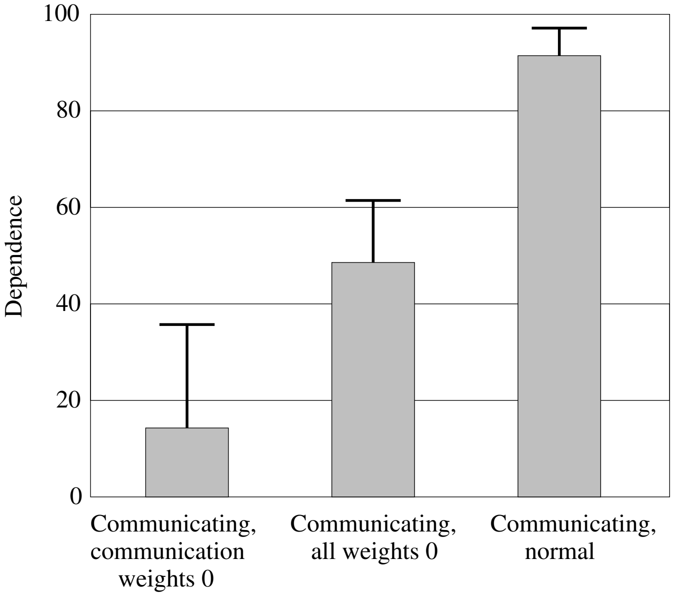

Fig. 16. **Degrees of dependence of communicating teams biased to evolve role-based cooperation.** The communicating teams with communication weights initialized to zero evolved behaviors that were 14% dependent on the positions of teammates; with all weights initialized to zero, the evolved behaviors were 48% dependent. In both cases, the evolved behaviors were significantly less dependent on teammates compared to those of the normal (randomly-initialized) communicating teams, which was 91% dependent on teammates (all differences are statistically significant, $p < 0.01$).

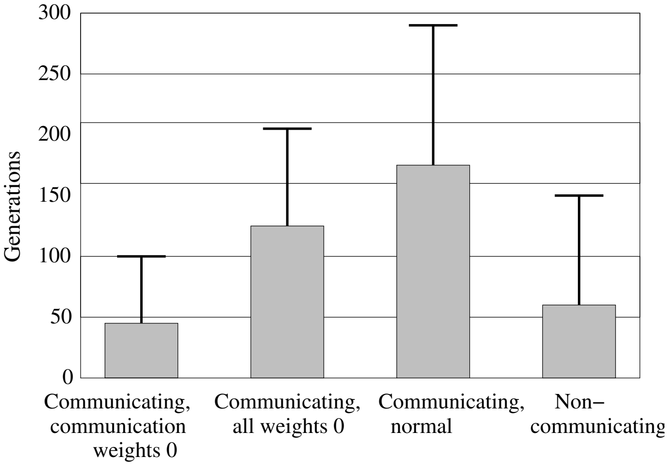

Fig. 17. **Evolution performance of communicating teams biased to evolve role-based cooperation.** The communicating teams with communication weights initialized to zero solved the task significantly faster than the normal communicating teams ($p < 0.01$), taking about the same number of generations as the non-communicating teams ($p = 0.32$). On the other hand, with all weights initialized to zero, evolution was as slow as with normal communicating teams ($p = 0.11$), and slightly slower than with communicating weights initialized to zero and non-communicating teams ($p < 0.10$).

in Section II-B and Figure 1. The predator was allowed to evolve until it could no longer improve its fitness. This process was repeated three times, each time with a new predator, to produce three independent but competent predators. These three predators were then put into the same environment and evaluated in the prey-capture task.

The results support the hypothesis that coevolution is necessary. When a predator evolves alone, it is never able to catch the prey, since the prey moves at the same speed as the predator. It evolves to chase the prey but is never able to reduce the distance between them, and is only able to prevent the prey from increasing it. When the three individually evolved predators are put together against the prey, they all chase the prey in the nearest direction, and are unable to catch it at all—the prey keeps running and maintains the distance (Figure 18). In other words, coevolution is essential in this task to evolve successful cooperative behavior.

F. Are Heterogeneous Agents Necessary?

Even though a team of individually evolved predators cannot catch the prey, it is possible that a homogeneous team, i.e. one where all predators are controlled by identical networks, could. In the current experimental setup, such a team can be successful only when the agents communicate. Otherwise, because they start out close together, they will all employ the same behavior, and fail like the team in the previous section.

In a separate experiment, communicating networks were evolved through the standard ESP method. Each network was evaluated by making three copies of it, each controlling one of the three predators in a team. The fitness of such a homogeneous team was then taken as the network's fitness. Networks of different sizes were evolved, from six hidden units up to 16, with ten runs at each size.

The success rate was extremely low: The homogeneous teams were able to solve the task only three times out of the 100 total runs. Apparently, it was difficult for evolution to discover a way to establish the roles initially in the beginning of the chase. In the heterogeneous communicating teams all agents are different and their behaviors diverge in the beginning, and the agents can immediately adopt a strategy that fits the situation. In the homogeneous team, all predators initially perform the same actions, and it is difficult to break the symmetry. A more effective form of communication might be to signal roles rather than simply sensing teammates' locations. Each predator in turn would decide on a distinct role (such as chase northward, or block horizontally), and communicate that to its teammates. Alternatively, evolving heterogeneous networks allows symmetry breaking to occur naturally, resulting in effective heterogeneous behaviors without a negotiation phase.

VII. Discussion

In Section IV-A, a central controller was found to take over 50% longer to evolve than autonomous cooperating controllers to solve the prey-capture task. Cooperative coevolution is able to decompose the task into simpler roles, thereby making it easier to discover solutions.

Such decomposition is a special case of speciation in evolutionary systems. Speciation has been widely used to maintain diversity in evolution. Using techniques such as islands and fitness sharing [31, 39, 50], separated populations are encouraged to diverge, resulting in more efficient search of the solution space. If these separated populations are further evaluated jointly and rewarded with a global fitness, they tend to converge to heterogeneous policies that work well together. This is the driving mechanism behind cooperative coevolution and also behind ESP and its multi-agent extension. ESP preserves diversity across populations of neurons and networks, because these populations are disjoint by design. Although diversity is gradually lost within each subpopulation as evolution focuses on a solution, the subpopulations become distinctly different from each other, and diverse and complementary roles result. Thus, the observed cooperation between predators emerges from cooperation between populations during evolution.

Section IV-B presented the result that predators without knowledge of teammates' relative locations evolve to cooperate and solve the task more than twice as fast than predators with such knowledge. This result is interesting because such knowledge from communication allows each predator to make decisions based directly on where the other predators are, as well as where the prey is. On the other hand, non-communicating predators do not have such knowledge, and have to coordinate using stigmergy. At first glance, this result seems attributable to three factors. First, allowing communication usually requires more structure (the minimal communicating autonomous controllers have eight neurons with 11 weights each, while the minimal non-communicating autonomous controllers have three neurons with seven weights each), which translates to a larger search space. However, this difference turns out to be unimportant. As discussed in Section VI-A, when the same architecture is used across all approaches, the non-communicating approach still outperforms the communicating approach substantially.

The second potential factor is that communication presents more information for evolution to understand and organize, which might take more evolutionary generations. While it is theoretically possible for evolution to simply discard the unnecessary information, Section VI-D demonstrated that evolution instead attempts to make use of it. On the other hand, Section VI-A showed that evolution takes little time to discard random noise fed into the extra inputs of the non-communicating approach: The number of generations taken to solve the task is the same as without noisy inputs. The conclusion is that with evolutionary methods such as ESP it is easy to discard noisy, random activations given as inputs, but difficult to discard inputs such as teammates' relative positions that "make sense," even if they are marginally useful at best.

The third potential factor, and actually the most important, is the different behaviors evolved in each approach and their relationship to the decompositional property of ESP. As was discussed in Section V, the non-communicating team always employed a single strategy where each agent had a specific role. During evolution, each non-communicating subpopulation converges toward optimizing specific functions such that the team solves the task successfully even though the team

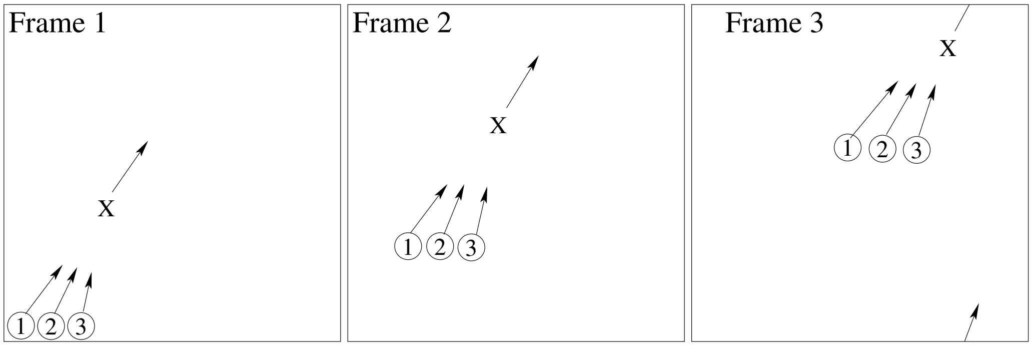

Fig. 18. **A strategy of three individually evolved predators placed on the same environment.** The predators chase the prey together in the nearest direction but are unable to catch it. Coevolution is thus essential in this task to evolve successful cooperative behavior.

members are independent of one another. Evolution without communication thus places strong pressure on each predator to perform its assigned role well. These roles are assigned through simultaneous adaptive niching: As one agent begins to converge to a particular behavior, the other agents that behave complementarily are rewarded, and themselves begin to niche into such roles. This adaptation in turn yields a higher fitness, and all predators begin to converge into cooperative roles. In this way, a stigmergic form of coordination between team members quickly emerges.

In contrast, the communicating team is continuously getting into different situations during evolution, depending on how ESP combines neurons and agents into teams. In order to achieve high fitness, an agent must perform well in a number of different situations. There is pressure to develop a rich set of behaviors that cover many situations and can change dynamically in reaction to the predators' actions as well as the prey's. As a result, the communicating agents utilize variations and combinations of two or more strategies, instead of clear-cut roles. It takes time to build an effective set of such behaviors, which is partly why communicating evolution takes longer.

Franklin [13] pointed out that communication abilities involve non-trivial costs, and suggested that in some cases coordination without communication may be an advantage over communication-based coordination. Our findings suggest that this is indeed true: First, communication presents extra information that ESP needs to organize and utilize, even if such information is not crucial in solving the task. Second, communication raises the structural requirements of the communicating controller in terms of number of hidden units and weights. Even if such complexity does not increase the number of generations needed to solve the task, it increases the computational resources needed to store, evaluate and operate upon the larger structures. On the other hand, stigmergy is faster and simpler for ESP to discover and represent, due to the powerful decompositional properties of the algorithm, and is utilized more beneficially in the task to find an efficient, fixed-role solution.

In general, the results in this paper suggest that it is better to use the most parsimonious architecture possible to solve a task, rather than an architecture with full capabilities in the hope that evolution will ignore the unnecessary capabilities (such as extra inputs and outputs). First, it is usually faster to evolve simpler network architectures. Efficiency becomes important especially with difficult real-world problems, which tend to be noisy and unpredictable. For example, Section VI-C showed that when predators start in random initial positions, the number of generations required to solve the task increased significantly, and each generation took longer because more sampling was required. Furthermore, Section VI-D demonstrated that evolution may not discard extraneous capabilities, but rather to use them in less efficient and less robust solutions. In tasks that are completely role-based, a particularly efficient way to simplify a system's architecture is to discard communication between agents. The system will then solve the task faster, and evolve more efficient and robust solutions.

Of course, not all multi-agent tasks may be as efficiently solved through role-based cooperation. For example in the prey-capture task where a capture configuration is necessary (as used by Benda [6] and Haynes and Sen [23, 24]) it would be very difficult for the agents to guess the exact locations of the other agents and timing of their actions to achieve successful capture. In contrast, if the agents can let other agents know where they are at any given time, they can coordinate their actions. Just as team behaviors were classified as either role-based or communication-based in Section V-B, it may be possible to classify multi-agent tasks likewise: Role-based cooperative tasks can be solved more efficiently by teams employing role-based cooperation without communication, while communication-based cooperative tasks require some degree of communication between teammates to be solved efficiently. The prey-capture task in this paper is entirely role-based, in that communication is not necessary at all.

Such a classification is useful because it makes it possible to identify the most effective approach for each task. As discussed above, employing a communicating team in a role-based task may not give the most efficient or most robust solution, while a non-communicating team may not even be able to solve a task that is communication-based. While it may not be possible to identify the type of task conclusively in every case, there are a few observations that are likely to be helpful. First, in a task that requires synchronized actions, communication between teammates is likely to be essential, and the task is therefore communication based. An example is a prey-capture task where all predators have to move to occupy a capture configuration simultaneously. (In contrast, in the current task, as long as the predators approach the prey from different directions, they will catch the prey; only the appropriate spatial organization is necessary to succeed.) Second, in a task where an agent may not be able to complete its role-based subtask, communication is likely to be required so that agents can seek assistance from teammates, or to notify them of the problem. For example, if the prey-capture task included irregularly-placed obstacles, stigmergic coordination might be insufficient to solve the task.

On the other hand, there are likely to be several real-world tasks where role-based cooperation is effective. These include tasks that lend themselves to territorial solutions [12]. For example, in controlling a bank of elevators in a building, each agent's role could be to serve a specific range of floors; in coordinating agents to search the web for information, each agent could cover a separate web region. They also apply to tasks where the roles are based on expertise, and where the agents react to the changing environment. For example, in deploying robots to cover a physical space (e.g. clean a surface, mow the lawn, rake the leaves, etc.) each one can react to what others have already done; in coordinating agents to construct an information object such as a Wikipedia article, each agent can provide a different expert perspective, building on what is already there in the article. In such tasks, communication may turn out to be a source of noise that diverts teams from the most effective solution. Systematically identifying such tasks and applying role-based cooperative coevolution to them constitutes a most interesting direction for future work.

VIII. Conclusion

The experiments reported in this paper showed that coevolving several autonomous, cooperating neural networks to control a team of agents is more efficient and robust than evolving a single centralized controller. They also showed that Multi-Agent ESP is an efficient and natural method for implementing such multi-agent cooperative coevolution. Furthermore, a class of tasks was identified, called role-based cooperative tasks, where communication is not necessary for success, and may actually make evolution less effective. Instead, a team of non-communicating agents can be evolved to utilize stigmergic coordination to solve the task more efficiently and robustly. This class is likely to include interesting real-world tasks. Recognizing such tasks and applying the cooperative coevolution approach to them, as well as studying the limits of stigmergic coordination in dealing with novel and changing environments, are the main directions of future work in this area.

Acknowledgments

This research was supported in part by the National Science Foundation under grants IIS-0083776, EIA-0303609, IIS-0757479, and IIS-0915038, and the Texas Higher Education Coordinating Board under grants ARP-003658-476-2001 and ARP-003658-0036-2007. We thank Bobby Bryant for insightful suggestions on how to test cooperation and communication, and Tino Gomez for suggestions on adapting ESP to multi-agent systems, as well as the anonymous reviewers for suggestions on stigmergy and network size experiments.

References

[1] Anderson, C. W. (1989). Learning to control an inverted pendulum using neural networks. *IEEE Control Systems Magazine*, 9:31–37.

[2] Balch, T. (1997). Learning roles: Behavioral diversity in robot teams. In Sen, S., editor, *Multiagent Learning: Papers from the AAAI Workshop. Technical Report WS-97-03*, 7–12. Menlo Park, CA: American Association for Artificial Intelligence.

[3] Baldassarre, G., Nolfi, S., and Parisi, D. (2003). Evolving mobile robots able to display collective behaviors. *Artificial Life*, 9:255–267.

[4] Barto, A. G., Sutton, R. S., and Anderson, C. W. (1983). Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13:834–846.

[5] Batali, J. (1994). Innate biases and critical periods: Combining evolution and learning in the acquisition of syntax. In Brooks, R. A., and Maes, P., editors, *Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems (Artificial Life IV)*, 160–171. Cambridge, MA: MIT Press.

[6] Benda, M., Jagannathan, V., and Dodhiawala, R. (1986). On optimal cooperation of knowledge sources—An empirical investigation. Technical Report BCS-G2010-28, Boeing Advanced Technology Center.

[7] Bryant, B. D., and Miikkulainen, R. (2003). Neuroevolution for adaptive teams. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC 2003)*, vol. 3, 2194–2201. Piscataway, NJ: IEEE.

[8] di Paolo, E. (2000). Behavioral coordination, structural congruence and entrainment in a simulation of acoustically coupled agents. *Adaptive Behavior*, 8:25–46.

[9] Elman, J. L. (1991). Incremental learning, or The importance of starting small. In Hammond, K. J., and Gentner, D., editors, *Proceedings of the 13th Annual Conference of the Cognitive Science Society*, 443–448. Hillsdale, NJ: Erlbaum.

[10] Ficici, S. G., and Pollack, J. B. (1998). Coevolving communicative behavior in a linear pursuer-evader game. In Pfeifer, R., Blumberg, B., Meyer, J.-A., and Wilson, S. W., editors, *Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior*, 505–514. MIT Press.

[11] Floreano, D., Mitri, S., Magnenat, S., and Keller, L. (2007). Evolutionary conditions for the emergence of communication in robots. *Curent Biology*, 17:514–519.

[12] Fontán, M. S., and Matarić, M. J. (1998). Territorial multi-robot task division. *IEEE Transactions of Robotics and Automation*, 14:815–822.

[13] Franklin, S. (1996). Coordination without communication. Technical report, Institute for Intelligent Systems, Department of Mathematical Sciences, University of Memphis.

[14] Gomez, F. (2003). *Robust Non-Linear Control Through Neuroevolution*. PhD thesis, Department of Computer Sciences, The University of Texas at Austin.

[15] Gomez, F., and Miikkulainen, R. (1997). Incremental evolution of complex general behavior. *Adaptive Behavior*, 5:317–342.

[16] Gomez, F., and Miikkulainen, R. (1999). Solving non-Markovian control tasks with neuroevolution. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, 1356–1361. San Francisco: Morgan Kaufmann.

[17] Gomez, F., Schmidhuber, J., and Miikkulainen, R. (2006). Efficient non-linear control through neuroevolution. In *Proceedings of the European Conference on Machine Learning*, 654–662. Berlin: Springer.

[18] Gomez, F., Schmidhuber, J., and Miikkulainen, R. (2008). Accelerated neural evolution through cooperatively coevolved synapses. *Journal of Machine Learning Research*, 9:937–965.

[19] Grassé, P.-P. (1959). La reconstruction du nid et les coordinations inter-individuelles chez *Bellicositermes natalensis* et *Cubitermes sp.* la théorie de la stigmergie: Essai d'interpretation des termites constructeurs. *Insectes Sociaux*, 6:41–83.

[20] Gruau, F., Whitley, D., and Pyeatt, L. (1996). A comparison between cellular encoding and direct encoding for genetic neural networks. In Koza, J. R., Goldberg, D. E., Fogel, D. B., and Riolo, R. L., editors, *Genetic Programming 1996: Proceedings of the First Annual Conference*, 81–89. Cambridge, MA: MIT Press.

[21] Hansen, N., and Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9:159–195.

[22] Haynes, T., and Sen, S. (1996). Evolving behavioral strategies in predators and prey. In Weib, G., and Sen, S., editors, *Adaptation and Learning in Multi-Agent Systems*, Lecture Notes in Computer Science, 113–126. Berlin: Springer.

[23] Haynes, T., and Sen, S. (1997). Crossover operators for evolving a team. In Koza, J. R., Deb, K., Dorigo, M., Fogel, D. B., Garzon, M., Iba, H., and Riolo, R. L., editors, *Genetic Programming 1997: Proceedings of the Second Annual Conference*, 162–167. San Francisco: Morgan Kaufmann.

[24] Haynes, T. D., and Sen, S. (1997). Co-adaptation in a team. *International Journal of Computational Intelligence and Organizations*, 1:231–233.

[25] Holland, J. H. (1986). Escaping brittleness: The possibilities of general-purpose learning algorithms applied to parallel rule-based systems. In Michalski, R. S., Carbonell, J. G., and Mitchell, T. M., editors, *Machine Learning: An Artificial Intelligence Approach*, vol. 2, 275–304. San Francisco: Morgan Kaufmann.

[26] Holland, O., and Melhuish, C. (1999). Stigmergy, self-organization, and sorting in collective robotics. *Artificial Life*, 5:173–202.

[27] Jim, K.-C., and Giles, C. L. (2000). Talking helps: Evolving communicating agents for the predator-prey pursuit problem. *Artificial Life*, 6(3):237–254.

[28] le Cun, Y., Denker, J. S., and Solla, S. A. (1990). Optimal brain damage. In Touretzky, D. S., editor, *Advances in Neural Information Processing Systems 2*, 598–605. San Francisco: Morgan Kaufmann.

[29] Lesser, V. R. (1999). Cooperative multiagent systems: A personal view of the state of the art. *IEEE Transactions on Knowledge and Data Engineering*, 11:122–142.

[30] Luke, S. (1998). Genetic programming produced competitive soccer softbot teams for Robocup97. In Koza, J. R., Banzhaf, W., Chellapilla, K., Deb, K., Dorigo, M., Fogel, D. B., Garzon, M. H., Goldberg, D. E., Iba, H., and Riolo, R., editors, *Genetic Programming 1998: Proceedings of the Third Annual Conference*, 214–222. Morgan Kaufmann.

[31] Mahfoud, S. W. (1995). *Niching Methods for Genetic Algorithms*. PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL.

[32] Meuleau, N., Peshkin, L., Kim, K.-E., and Kaelbling, L. P. (1999). Learning finite-state controllers for partially observable environments. In *Proceedings of the Fifteenth International Conference on Uncertainty in Artificial Intelligence*, 427–436. San Francisco: Morgan Kaufmann.

[33] Miller, G., and Cliff, D. (1994). Co-evolution of pursuit and evasion I: Biological and game-theoretic foundations. Technical Report CSRP311, School of Cognitive and Computing Sciences, University of Sussex, Brighton, UK.

[34] Mitchell, M. (2006). Coevolutionary learning with spatially distributed populations. In Yen, G. Y., and Fogel, D. B., editors, *Computational Intelligence: Principles and Practice*. Piscataway, NJ: IEEE Computational Intelligence Society.

[35] Moriarty, D. E. (1997). *Symbiotic Evolution of Neural Networks in Sequential Decision Tasks*. PhD thesis, Department of Computer Sciences, The University of Texas at Austin. Technical Report UT-AI97-257.

[36] Moriarty, D. E., and Miikkulainen, R. (1996). Efficient reinforcement learning through symbiotic evolution. *Machine Learning*, 22:11–32.

[37] Moriarty, D. E., and Miikkulainen, R. (1997). Forming neural networks through efficient and adaptive coevolution. *Evolutionary Computation*, 5:373–399.

[38] Moriarty, D. E., Schultz, A. C., and Grefenstette, J. J. (1999). Evolutionary algorithms for reinforcement learning. *Journal of Artificial Intelligence Research*, 11:199–229.

[39] Mühlenbein, H., Schomisch, M., and Born, J. (1991). The parallel genetic algorithm as function optimizer. *Parallel Computing*, 17:619–632.

[40] Pendrith, M. (1994). On reinforcement learning of control actions in noisy and non-Markovian domains. Technical Report UNSW-CSE-TR-9410, School of Computer Science and Engineering, The University of New South Wales, Sydney, Australia.

[41] Potter, M. A., and Jong, K. A. D. (2000). Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation*, 8:1–29.

[42] Quinn, M., Smith, L., Mayley, G., and Husbands, P. (2002). Evolving teamwork and role-allocation with real robots. In *Proceedings of the Eighth International Conference on Artificial Life*, 302–311. Cambridge, MA: MIT Press.

[43] Ramachandran, S., and Pratt, L. Y. (1992). Information measure based skeletonisation. In Moody, J. E., Hanson, S. J., and Lippmann, R. P., editors, *Advances in Neural Information Processing Systems 4*, 1080–1087. San Francisco: Morgan Kaufmann.

[44] Reisinger, J., Stanley, K. O., and Miikkulainen, R. (2004). Evolving reusable neural modules. In *Proceedings of the Genetic and Evolutionary Computation Conference*.

[45] Rosin, C. D., and Belew, R. K. (1997). New methods for competitive evolution. *Evolutionary Computation*, 5.

[46] Santamaria, J., Sutton, R., and Ram, A. (1998). Experiments with reinforcement learning in problems with continuous state and action spaces. *Adaptive Behavior*, 6(2).

[47] Saravanan, N., and Fogel, D. B. (1995). Evolving neural control systems. *IEEE Expert*, 23–27.

[48] Saunders, G. M., and Pollack, J. B. (1996). The evolution of communication schemes over continuous channels. In Maes, P., Mataric, M. J., Meyer, J.-A., Pollack, J., and Wilson, S. W., editors, *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, 580–589. MIT Press.

[49] Savage, T. (1998). Shaping: The link between rats and robots. *Connection Science*, 10:321–340.

[50] Smith, R. E., Forrest, S., and Perelson, A. S. (1993). Searching for diverse, cooperative populations with genetic algorithms. *Evolutionary Computation*, 1:127–149.

[51] Smith, S. F. (1980). *A Learning System Based on Genetic Adaptive Algorithms*. PhD thesis, Department of Computer Science, University of Pittsburgh.

[52] Stanley, K. O., and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10:99–127.

[53] Stanley, K. O., and Miikkulainen, R. (2004). Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research*, 21:63–100.

[54] Steels, L. (1996). Self-organising vocabularies. In Langton, C. G., and Shimohara, K., editors, *Proceedings of the 5th International Workshop on Artificial Life: Synthesis and Simulation of Living Systems (ALIFE-96)*, 179–184. Cambridge, MA: MIT Press.

[55] Sutton, R. S., McAllester, D., Sing, S., and Mansour, Y. (2001). Policy gradient methods for reinforcement learning with function approximation. In Leen, T. K., Dietterich, T. G., and Tresp, V., editors, *Advances in Neural Information Processing Systems 13*, 1057–1063. Cambridge, MA: MIT Press.

[56] Urzelai, J., Floreano, D., Dorigo, M., and Colombetti, M. (1998). Incremental robot shaping. *Connection Science*, 10:341–360.

[57] Wagner, K. (2000). Cooperative strategies and the evolution of communication. *Artificial Life*, 6:149–179.

[58] Watkins, C. J. C. H., and Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3):279–292.

[59] Werner, G. M., and Dyer, M. G. (1991). Evolution of communication in artificial organisms. In Langton, C. G., Taylor, C., Farmer, J. D., and Rasmussen, S., editors, *Proceedings of the Workshop on Artificial Life (ALIFE '90)*, 659–687. Reading, MA: Addison-Wesley.

[60] Whitley, D., Dominic, S., Das, R., and Anderson, C. W. (1993). Genetic reinforcement learning for neurocontrol problems. *Machine Learning*, 13:259–284.

[61] Whitley, D., Gruau, F., and Pyeatt, L. (1995). Cellular encoding applied to neurocontrol. In Eshelman, L. J., editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, 460–469. San Francisco: Morgan Kaufmann.

[62] Whitley, D., Mathias, K., and Fitzhorn, P. (1991). Delta-Coding: An iterative search strategy for genetic algorithms. In Belew, R. K., and Booker, L. B., editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, 77–84. San Francisco: Morgan Kaufmann.

[63] Wiegand, R. P. (2003). *An Analysis of Cooperative Coevolutionary Algorithms*. PhD thesis, George Mason University.

[64] Wooldridge, M. (2002). *An Introduction to Multiagent Systems*. Hoboken, NJ: Wiley.

[65] Wooldridge, M. J., and Jennings, N. R. (1999). Cooperative problem solving. *Journal of Logic and Computation*, 9:563–592.