An Experiment in Mixing Evolving and Preprogrammed Robots

Sancho Oliveira[1], Luis Nunes[1] and Anders Lyhne Christensen[1]

[1]Instituto de Telecomunicações & Instituto Universitário de Lisboa (ISCTE-IUL), Lisboa, Portugal
sancho.oliveira@iscte.pt

Abstract

Artificial evolution of robot behavior is commonly conducted in environments containing a single robot or multiple robots that are all controlled by evolving behavioral logic. In this paper, we take a novel approach and study how the presence of preprogrammed robots affects the evolutionary process and the solutions evolved. We evolve behavioral control that enables robots to forage. The robots are situated in an environment that contains a nest and a number of prey. The robots must either push or carry the prey to the nest. We analyze the behaviors evolved in mixed setups in which one or more preprogrammed robots are present. We compare these behaviors to behaviors evolved in non-mixed setup in which no preprogrammed robots are present. The results show that although the evolved robots do not use their capacity to communicate, they do collaborate with the preprogrammed robots. We find that the performance of some of the solutions evolved in the mixed setup is higher than the performance of homogeneous groups of robots.

Introduction

In this paper, we take a novel approach to the evolution of behavioral control for robots. We report on experiments in which we evolve behaviors for robots that share the environment with preprogrammed robots. The preprogrammed robots are (aside from their behavior) indistinguishable from the evolving robots. Mixing evolving robots with preprogrammed robots is interesting for several reasons: from an engineering perspective, artificial evolution may be used to fill in the gaps between partially known (easily preprogrammable) solutions to complex tasks and/or to optimize the performance of a robot collective. From an evolutionary perspective, it is interesting to evaluate how the presence of robots programmed with a solution influences the evolutionary process and the solutions evolved – such as determining whether the evolving robots adopt the preprogrammed solution and/or whether they learn to communicate with the preprogrammed robots.

We use a multirobot foraging task for our experiments. A robot can push prey or it can pick up and carry a prey. If a prey-carrying robot collides with another robot, it loses the prey. Thus, the robots must avoid collisions when carrying prey. The preprogrammed robots have the same sensory and actuation capabilities as the evolving robots. Each robot can control the color of its body. Whenever carrying prey, a preprogrammed robot sets its body color to red. When not carrying a prey, a preprogrammed robot sets its body color to green. Thus, nearby robots can see when a preprogrammed robot is carrying a prey or not and give way in order to avoid collisions. Since evolving robots have control over their body color too, they have the potential to communicate to nearby teammates in the same way as the preprogrammed robots do.

In this study, we analyze and discuss the fitness trajectories and the solutions obtained in evolutionary runs where one preprogrammed robot and two evolving robots are present. We discuss if and how the robots collaborate and communicate. We setup an experiment in which we take an incremental approach to evolution in order to increase the rate of solutions with a high average fitness. Finally, we report on experiments in which three preprogrammed robots and six evolving robots are present during evolution.

The contribution of this paper is three-fold: i) We demonstrate that evolving robots can learn to collaborate with preprogrammed robots. ii) We demonstrate how a basic incremental approach to evolution can increase the rate at which collaborative solutions are evolved when preprogrammed robots are present. iii) We show that heterogeneous groups of preprogrammed robots and evolved robots can achieve a better performance than homogeneous groups of preprogrammed robots.

Related work

Interest in evolutionary robotics started in the early 90s (Cliff et al., 1993; Nolfi and Floreano, 2000). Initially, focus was on evolving a controller for a single robot to perform relatively simple tasks such as obstacle avoidance, exploration, and navigation (see for instance Nolfi et al. (1994)). Recently, there have been several studies on the evolution of controllers for multirobot systems—particularly those systems in which control is decentralized and in which individual robots have limited sensory capabilities. In swarm robotics research (Şahin, 2005), it has been demonstrated how the application of evolutionary robotics can overcome the fundamental design problem of deriving microscopic rules for the individuals such that the desired macroscopic behavior emerges. When artificial evolution is applied to swarms of robots, the designer can specify a fitness function that scores the collective behavior and let evolution search the space of individual behaviors. Using this approach, Dorigo et al. (2004) demonstrated how a group of homogeneous robots could be evolved to aggregate and to display coordinated-motion when physically connected to each other. In another study, Trianni et al. (2006) demonstrated how a group of evolved homogeneous robots could cooperatively avoid holes.

Evolutionary robotics has been applied to heterogeneous multirobot systems: Tuci et al. (2008) evolved homogeneous controllers for heterogeneous robots. Nolfi and Floreano (1998) co-evolved a predator agent and a prey agent. The fitnesses of the two types of agents were co-dependent although each had a different genome.

It has also been demonstrated that heterogeneity can arise in a homogeneous system (identical agents with identical neuro-controllers). Quinn et al. (2003) evolved controllers for a team of three robots with minimal sensory capabilities. The robots' task was to aggregate and then travel a distance of one meter as a group. Interestingly, the team members dynamically adopted roles and moved in a line formation. The robot that would adopt the role as the leader, moved backward in order to perceive the middle robot. The middle and rear robot, on the other hand, moved forward. Ampatzis et al. (2009) evolved homogeneous controllers for two real robots that allowed them to *self-assemble*, that is, physically connect to one another. However, the robots first had to allocate roles so that one would be the gripping robot, while the other would be the gripped robot. The roles were allocated during what can be described as a dance: the robots would circle each other while performing oscillatory movements until one would approach the other to perform the grip.

In this study, we use a novel evolutionary setup. We explore the effect of the presence of preprogrammed robots on the evolved behaviors. We find that the heterogeneity in the group composition leads to role allocation and collaboration.

Robot Model and The Task

Below, we start by presenting the robot model that we use. We go on to describe the foraging task and the environment. Finally, we briefly discuss the software simulator in which we conduct our experiments.

The Robot Model

We use a differential drive, cylindrical robot model. Each robot has a diameter of 10 cm. The set of actuators is composed of two wheels, a prey carry mechanism and a changeable body color. The two wheels can be controlled independently allowing a robot to move and to turn. Gaussian noise with standard deviation of 5% is added independently to the left wheel speed and to the right wheel speed set by the robot controller in order to simulate issues such as slippage, slightly uneven ground and so forth. The prey carry mechanism enables a robot to pick up a prey within a distance of 5 cm. The body color actuator has three possible settings: green, red, and black. Whenever green or red, a robot can be detected by other nearby robots, while when black, the robot is invisible to other robots.

The robots are equipped with several sensors that allow them to perceive i) whether they are currently carrying a prey or not (prey-carried sensor), ii) whether they are inside the nest or not (in-nest sensor), and iii) the presence of nearby objects: eight nest sensors, eight prey sensors, eight red robot sensors, and eight green robot sensors.

Aside from the prey-carried sensor and the in-nest sensor, all the sensors operate in a similar way, but register different types of objects. The nest sensors only register the nest. The prey sensors only register prey. The green robot sensors only register green robots. The red robot sensors only register red robots. The sensors are distributed evenly around the robot's body.

A sensor only registers objects within a certain distance and angle with respect to its orientation. All sensors have an opening angle of $135\,^\circ$ and a range of 1 meter, except for the nest sensors which have a range of 10 meters. If there are no sources within sensor's range and opening angle, its reading is 0. Otherwise, the reading is based on distance to the closest source (c) according to the following equation:

$$s = \frac{range - d_c}{range}, \tag{1}$$

where $range$ is the sensor's detection range and d_c is the distance between the closest source c and the sensor.

The Foraging Task

Our experiments are conducted in the arena shown in Figure 1. The robots must search for prey and transport them to the circular nest area with a diameter of 0.50 m centered in the arena. The nest can be perceived by the robots using their nest sensors. The prey are scattered in the foraging area around the nest. The foraging area is circular and has a diameter of 4 meters. Whenever a prey is dropped in the nest, it is immediately redeployed to a random location in the foraging area. 13 prey are present in the environment which results in a prey density of $1\ \text{prey}/\text{m}^2$. When a prey-carrying robot collides with another robot, it loses the prey that it was carrying. The lost prey is randomly redeployed in the environment in order to keep the prey density constant.

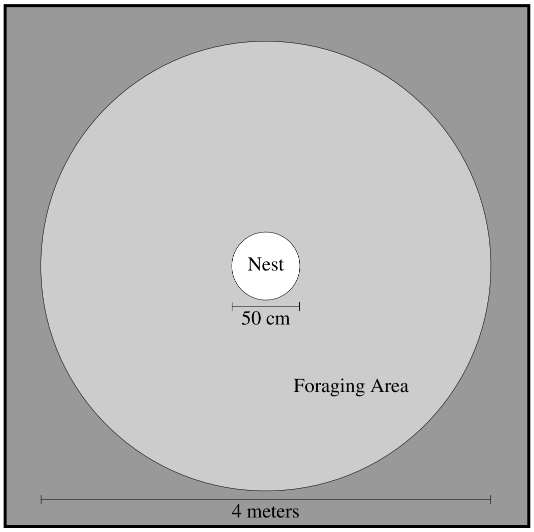


Figure 1: Foraging arena.

Simulation Environment

We have implemented the robot model and constructed the environment discussed above in JBotEvolver (see http://sourceforge.net/projects/jbotevolver). We have implemented our own neuro-evolution framework that allows for distributed, fault tolerant fitness evaluation.

Controller Architecture

Below, we present the control logic for the preprogrammed robot and the artificial neural network used for the evolving robots.

Preprogrammed Robots

A finite state machine representation of the control program for the preprogrammed robots is shown in Figure 2. A preprogrammed robot starts of in the "Search" state in which it locates and moves towards the nearest prey. If the preprogrammed robot detects the presence of a red robot in its way, it assumes that the red robot is carrying a prey and therefore turns around ($180\,^{\circ}$) and moves out of the way (state "Make way"). When a prey is encountered, the preprogrammed robot attempts to pick it up (state "Pick up"). If the prey is picked up successfully, the preprogrammed robot becomes red and starts moving towards the nest (state "Transport"). When the nest is reached, the preprogrammed robot drops the prey (state "Drop") and returns to the "Search" state. If the preprogrammed fails to pick up a prey or if it loses the prey (due to a collision), the preprogrammed robot returns to the "Search" state.

In the finite state machine in Figure 2, we have colored the states with the color that a preprogrammed robot has when in the respective states. Whenever a prey is carried, the preprogrammed robot is red. Otherwise, the preprogrammed robot is green. Preprogrammed robots are never black.

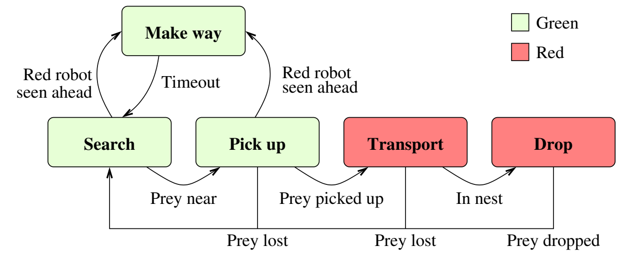


Figure 2: Preprogrammed controller.

The Evolving Robots

The evolving robots are controlled by a continuous time recurrent neural network (Beer and Gallagher, 1992). The network consists of three layers of neurons: an input layer with 34 neurons, a hidden layer with 5 neurons, and an output layer with 4 neurons. The input neurons I_i are reactive. The prey-inputs (I_1 to I_8), the nest-inputs (I_9 to I_{16}), the green-inputs (I_{17} to I_{24}), and the red-inputs (I_{25} to I_{32}) are all set based on sensor readings from the respective sensors. The prey-carried-input (I_{33}) is 1 if a prey is currently carried and 0 otherwise. The in-nest-input (I_{34}) is 1 if the robot is in the nest and 0 otherwise. The neurons in the hidden layer are fully connected and governed by the following equation:

$$\tau_i \frac{dH_i}{dt} = -H_i + \sum_{j=1}^{34} \omega_{ji} I_i + \sum_{k=1}^{5} \omega_{ki} Z(H_k + \beta_k), \quad (2)$$

where τ_i is the decay constant, H_i is the neuron's state, ω_{ji} the strength of the synaptic connection from neuron j to neuron i, β the bias terms, and $Z(x) = (1 + e^{-x})^{-1}$ is the sigmoid function. τ, β, and ω_{ji} are genetically controlled network parameters. The possible ranges of these parameters are: $\tau \in [0.1, 32]$, $\beta \in [-10, 10]$ and $\omega_{ji} \in [-10, 10]$. Cell potentials are set to 0 when the network is initialized and circuits are integrated using the forward Euler method with an integration step-size of 0.2.

The output layer is fully connected to the neurons in the hidden layer. The activation of the output neurons is given by the following equation:

$$O_i = \sum_{j=1}^{4} \omega_{ji} Z(y_j + \beta_j); \quad (3)$$

The first two outputs O_1 and O_2 control the speed of the left and the right wheel, respectively. Their output is linearly mapped to speeds in the range $[-50$ cm/s, 50 cm/s$]$. The third output O_3 is mapped to the prey carrying mechanism: if $O_3 > 0.5$, the robot attempts to pick up the closest prey or to hold a prey if one is already carried. If $O_3 \leq 0.5$, any prey carried will be dropped. The fourth output O_4, controls the color of the robot. For values in the range [0, 0.33] the robot becomes invisible to other robots, for values in the range $]0.33, 0.66[$, the robot becomes green, and for values in the range $[0.66, 1.00]$, the robot becomes red.

Evolutionary Algorithm

We use a simple generational evolutionary algorithm (Schwefel, 1995; Goldberg, 1989). Each generation consists of 100 genomes. Each genome consists of a vector of 228 real valued numbers. These values encode the weights of the synaptic connections between neurons, the bias terms and the decay constants for a neural network with the topology described in the previous section. After sampling the fitness of each genome in a generation, the 5 best genomes are retained and the rest are discarded. These 5 genomes are the parents of the subsequent generation. From each parent an equal number of children (19) are created and the parents are copied to the new generation. The genotype for a child is obtained adding a random Gaussian offset to each real-valued gene with a probability of 15%.

We compute the fitness at the group-level. Thus, in the experiments where a preprogrammed robot is present, its behavior and its performance contribute to the fitness of the group in the same way as the behavior of the evolving robots. The fitness function $F(i)$ is given below:

$$F(i) = P_i + \sum_{s=1}^{\text{time-steps}} f_{i,s} \tag{4}$$

where i is the genome being evaluated, P_i is the number of prey foraged and $f_{i,s}$ is computed at every time-step, s. The term $f_{i,s}$ is computed in the following way:

$$f_{i,s} = 10^{-3}c_s + 10^{-4}d_s \tag{5}$$

where c_s is the number of robots carrying a prey at time-step s and d_s is a prey distance reward that depends on the distance between each prey and the nest at time-step s. The prey distance reward is computed using the formula:

$$d_s = \frac{1}{\text{n° prey}} \sum_{j=1}^{\text{n° prey}} \frac{1.75\ \text{m} - dist(p_j, \text{nest})}{1.75m}. \tag{6}$$

We sample the fitness of each genome five times and selection is based on the average fitness obtained.

Results and Discussion

We initially experimented with two different evolutionary setups: a *mixed* setup in which two evolving robots and one preprogrammed robot were present, and a *non-mixed* setup in which three evolving robots were present. In each setup, we performed 30 evolutionary runs with different initial random seeds for 2000 generations each. Each generation consisted of 100 genomes. The fitness of each genome was sampled in five trials of five minutes of virtual time (3000 control steps) each.

Below, we provide an overview of the results obtained. We then describe the different types of behaviors evolved in the *non-mixed* setup and the *mixed*, respectively. We go on to discuss cooperation and communication. We then experiment with incremental evolution in order to speed up evolutionary learning. Finally, we experiment with setups in which nine robots are present.

Fitness Trajectories

The plot in Figure 3 summarizes the results of the evolutions runs conducted in the *mixed* setup and in the *non-mixed* setup, respectively. The figure shows the average fitness of the best genome in each generation in all the 30 runs conducted in the *mixed* setup and in the *non-mixed* setup, respectively. We have included the fitness trajectory for the single highest scoring *mixed* run and for the single highest scoring *non-mixed* run. The horizontal line at $y = 119.9$, shows the average fitness obtained by three preprogrammed robots alone in the environment.

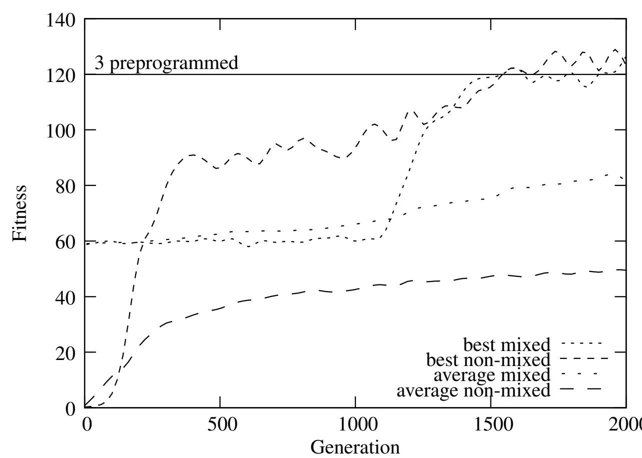


Figure 3: The fitness scores of the best and the average the best genomes in all the runs in the *mixed* setup and in the *non-mixed* setup. The horizontal line at $y = 119.9$ indicates the average performance of a team of three preprogrammed robots.

The results in Figure 3 show that the fitness of the best genome in the *mixed* setup is on average higher than the best genomes in the *non-mixed* setup. The higher fitness in the beginning of an evolutionary run in the *mixed* setup is explained by the presence of the preprogrammed robot. The preprogrammed robot finds and transports prey to the nest from the onset of an evolutionary run whereas the evolving robots first have to learn to forage.

When a preprogrammed robot is alone in the environment, it obtains an average fitness of 60.0. When three preprogrammed robots are present in the environment, they interfere with one another. Furthermore, as a trial progresses, prey tend to be distributed further from the nest since more preprogrammed robots tend to forage the prey close to the nest faster. Interference and the increased prey distance both have negative impacts on the fitness score. Three preprogrammed robots therefore obtain a fitness (119.9) that is less than three times what a single preprogrammed robot obtains on average (60.0). In the beginning of an evolution run when the evolving robots are not yet foraging, the preprogrammed robot can often forage undisturbed in the *mixed* setup. The average fitness in the beginning of an evolutionary run in the *mixed* setup is therefore close to the fitness obtained by a single preprogrammed robot operating alone.

Behavioral Analysis

In this section, we analyze the evolved robots' behaviors. A summary of the post evaluation scores for the 30 evolutionary runs conducted in the *non-mixed* setup and the 30 evolutionary runs conducted in the *mixed* setup can be seen in Figure 4. In the plot, we have grouped the evolutionary runs according to their foraging behaviors and fitness.

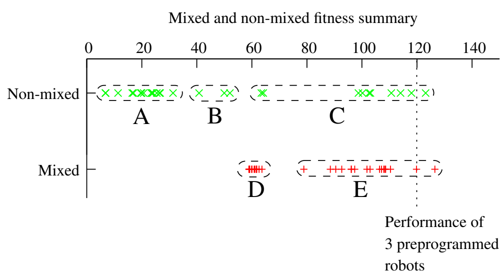


Figure 4: Summary of the post evaluation of the best behavior evolved in each evolutionary run in the *non-mixed* setup and in the *mixed* setup. We have divided the evolved solutions into groups A to E based on fitness and behavior.

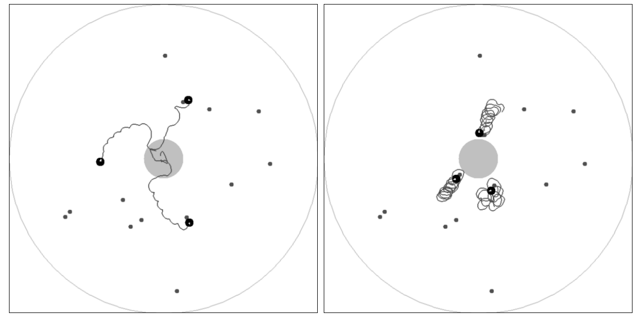


Figure 5: An example of a behavior in group A evolved in the *non-mixed* setup (two screenshots from the same experiment). The robots forage by pushing prey towards the nest. As can be seen on the figure, this behavior results in a lot of small circular movements and is thus not very effective.

In the *non-mixed* evolutionary runs, we observed behaviors that can be divided into three groups: A, B and C. All the solutions in all groups successfully forage prey, however, they forage in different ways. The behaviors group A all rely on pushing prey towards the nest. An example of the pushing behavior can be seen in Figure 5. The pushing behavior requires the robots to move in small circular patterns to constantly get behind the prey and the behavior is thus not very efficient.

The behaviors in group B rely on continually picking up and dropping prey. When a prey is picked up, it is often dropped after a single or a few control cycles, only to be picked up again immediately. One of the behaviors in group B is particularly interesting: often the robots transport two or more prey at a time by repeatedly picking up, dropping different prey. An example can be seen in Figure 6. Transporting multiple prey, however, comes at a cost: since a robot can only carry one prey at a time, it has to constantly make small circular movements to pick up the prey left behind. This means that the average fitness of the behavior in group B is lower than the average fitness of the behaviors in the last group of behaviors evolved in the *non-mixed* setup, group C.

In group C, the robots pickup prey and transport the prey back to the nest. The differences in fitness between the different solutions are due to a number of factors: how the robots search for prey, how efficient they are in moving to a prey once they have located the prey, and if and how much they interfere with one another. Some robots move away from the nest in a straight line to search for prey, some robots circle away from the nest, while in other cases, the robots move in more irregular patterns. Most of the robots move only forward or only backward, however, for some behaviors, the robots change direction once a prey is picked up. Changing direction is especially efficient for those robots that move directly from the nest to a prey: when a prey is picked up, they change direction (without having to turn around) to transport the prey back to the nest. Examples of some of the behaviors in group C can be seen in Figure 7.

We have divided the behaviors evolved in the *mixed* setup into two groups: D and E (see Figure 4). Group D contains the lowest scoring behaviors evolved in the *mixed* setup. The evolving robots in this group do not contribute to the foraging, but instead move away from the nest in order to let the preprogrammed robot forage undisturbed. In some cases, the evolving robots move beyond the foraging area, in some cases the evolved robots remain in the foraging area, and sometimes they even pickup prey.[1] However, in none of the cases do the evolved robots attempt to move prey closer to

[1] Carrying prey is rewarded in the fitness function (see the "Evolutionary Algorithm" section).

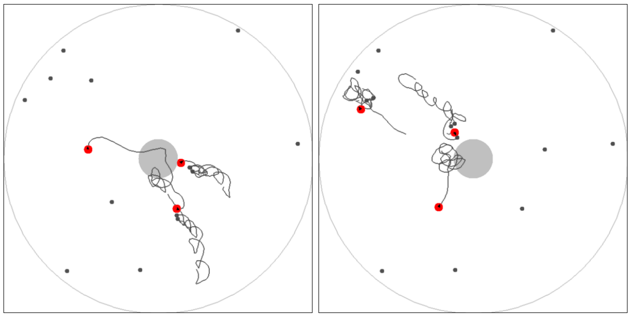


Figure 6: An example of the behavior in group B evolved in the *non-mixed* setup (two screenshots from the same experiment). By continually picking up and dropping prey, the robots are able to transport multiple prey towards the nest at the same time.

the nest.

The evolved solutions in group E all obtained an average post evaluation fitness of more than 70. In all of these solutions, the evolved robots actively forage. The difference in performance is due to the way in which the evolved robots search for prey: some of the evolved robots move directly towards prey close to the nest while others circle the foraging area and forage mainly prey located far away from the nest (thereby leaving the prey close to the nest for the preprogrammed robot to pickup). This type of behavior indicates that the evolved robots collaborate with the preprogrammed robot.

Collaboration To examine the level of collaboration (if any) between the evolving and preprogrammed robots, we analyzed if there is some evidence of division of labor: we recorded the number of prey foraged by evolved robots and the number of prey foraged by the preprogrammed robot in the *mixed* setup. We ran 100 trials with each of the highest scoring genomes from the 30 evolutions conducted in the *mixed* setup. For 16 of the 30 genomes, the preprogrammed robot forages significantly more prey when the evolved robots are present compared to when it is the only robot in the environment (Mann-Whitney, $p < 0.05$).

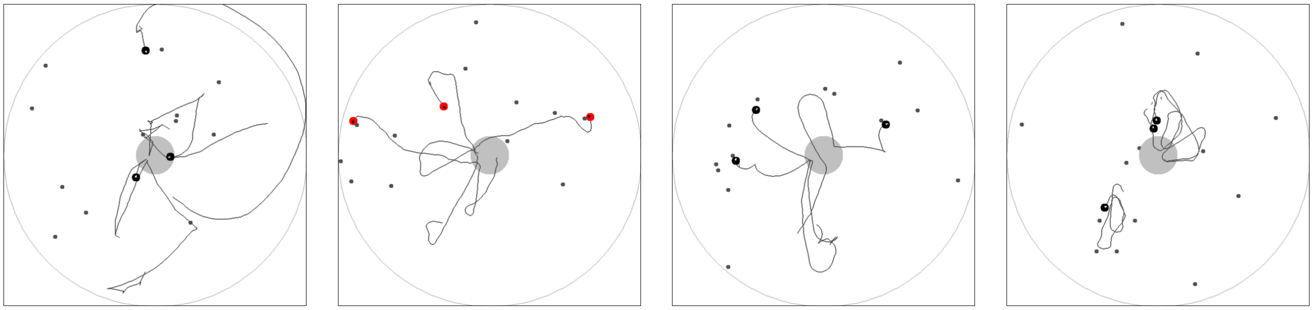
When the preprogrammed robot is alone, it forages 57.9 prey on average during a five minute trial, while when three preprogrammed robots are present, each forages on average 38.7 prey. When evolved robots are present, the preprogrammed robot forages an average of 75.3 prey per trial for the best solution in the *mixed* setup. These results indicate that the evolved robots have learned to collaborate with the preprogrammed robot. For the best solution in the *mixed* setup, the average distance (over 100 five minute trials) of the preprogrammed robot from the center of the nest was 0.54 m, while the average distance of the each of the two evolved robots from the center of the nest was 1.06 m. The evolving robots forage prey that are located far from the nest and leave the prey close to (but not always in) the nest. The preprogrammed robot (which prioritizes prey located close to the nest) then transports the prey left by the evolving robots the rest of the way to the nest. The division of labor is efficient because the evolved robots in general operate far from the nest, while the preprogrammed robot operates close to and in the nest – collisions are therefore avoided.

Communication The robots in both the *non-mixed* and the *mixed* setups have the capacity to change their body color and to detect the body color of nearby teammates. This capacity potentially allows the robots to communicate. However, in 22 out of 30 evolutionary runs in the *non-mixed* setup, the evolved robots remain mainly black (invisible to one another) during experimental trials. In the remaining 8 runs, the robots either remain mainly red (5) or constantly change color (3) during a trial.

In order to determine if communication plays a major role in the evolved solutions, we ran three sets of experiments in the *non-mixed* setup, where we fixed the body color of all the robots to black, red and green in 100 trials each. The differences in terms of performance when the body color is fixed and when the neural network has the control over the body color were minimal. The average performance difference was only 0.5%, with the largest drop being 3.7% and the largest increase in performance being 5.4%.

In a similar set of experiments in the *mixed* setup, we fixed the color of the preprogrammed robot and the two evolving robots. Fixing the body color to red results in an average performance drop of 22.6%. This drop is explained by the fact that the preprogrammed robot attempts to make way each time it encounters a red robot. The average difference in performance when the body color is fixed to either black or green and when the controllers have control over the body color was 0.6% with the largest difference being 3.2%. This indicates that the performance of the evolved robots does not depend on their capacity to change their body color.

It is surprising that the robots did not evolve to exploit their capacity to change color in the *mixed* setup to communicate with the preprogrammed robot (which already communicates its internal state by changing color depending on whether it is carrying a prey or not). A probable explanation for the lack of communication is that the robots can forage efficiently in the *mixed* setup without communicating. As discussed in the previous section, the evolved robots do in most cases learn to collaborate with preprogrammed robot by transporting prey located far from the nest closer to the nest for the preprogrammed transport the rest of the way to the nest. The robots operate in different regions of the environment and they do therefore not need to communicate in order coordinate their actions or to avoid collisions.



(a) (b) (c) (d)

Figure 7: Examples of the behaviors from group C evolved in the *non-mixed* setup (screenshots from different experiments with different controllers). (a) the robots move directly to prey or start circling the foraging area in case no prey is found. (b) the robots have set their body color to red and turn around once a prey is picked up. (c) the robots move to prey in arcs. (d) robots pick up and carry prey, but they often interfere with one another.

Incremental Evolution in the *mixed* setup Of the 30 evolutionary runs conducted in the *mixed* setup, the 12 runs in group D did not evolve foraging behaviors, but instead, the evolved robots move away from the nest in order to avoid interfering with the preprogrammed robot. This solution is a local maximum in the fitness landscape because the preprogrammed robot is an efficient forager from the onset of the evolutionary process and any interference – a lost prey due to a collision for instance – would result in a lower collective fitness. We set up a series of experiment in which we tried to increase evolutionary pressure towards solutions in which the evolving robots participate in the foraging by initially reducing the speed of the preprogrammed robot. When the preprogrammed robot moves at a reduced speed, it forages less than when moving at full speed. Evolutionary pressure towards solutions in which the evolving robots actively forage is thus increased because any contribution made by the evolving robots proportionally is higher with respect to the fitness obtained by the team than when the preprogrammed robot is moving at full speed. In a new *incremental mixed* setup the preprogrammed robot initially moved at 50% of the full speed. Once a collective fitness of 50 was reached by the highest scoring individual in a generation, the speed of the preprogrammed robot was increased to full speed.

We performed 30 evolutionary runs in the *incremental mixed* setup. Out of the 30 evolutionary runs, only 6 produced non-foraging behaviors compared to 12 in the normal (non-incremental) *mixed* setup. The average of the post evaluation fitness of the best genome from each run in the *incremental mixed* was 90.0 compared to 84.7 in the *mixed* setup. For 24 of the 30 genomes, the preprogrammed robot forages significantly more prey when the evolved robots are present compared to when it is the only robot in the environment (Mann-Whitney, $p < 0.05$). Hence, in the *incremental mixed* setup, the evolving robots learn more frequently to collaborate with the preprogrammed robot than in the non-incremental *mixed* setup. Visual inspection of the successful solutions evolved in the *incremental mixed* setup confirmed that they are similar to the successful solutions evolved in the non-incremental *mixed* setup (that is, the behaviors in group E in Figure 4).

Performance in larger *mixed* groups In order to determine if and how the mixture of preprogrammed and evolved could benefit larger groups of robots, we conducted experiments in which nine robots were present in the environment: three preprogrammed robots and six evolving robots. We conducted the evolution in the same environment and with the same fitness function as used above. We used an incremental setup with four increments:

1st increment: Only the six evolving robots were present [Fitness limit: 20].

2nd increment: The three preprogrammed robots were introduced but moving at 25% of full speed [Fitness limit: 100].

3rd increment: The speed of the three preprogrammed robots was increased to 50% of full speed [Fitness limit: 200].

4th increment: The speed of the three preprogrammed robots was increased to full speed.

We conducted 30 evolutionary runs till the 2000th generation. The average fitness obtained in a post evaluation (100 samples) of the best chromosome from each run was 358. The average fitness score obtained in 100 samples with a homogeneous group of nine preprogrammed robots was 363. The average post evaluation fitness obtained by the larger mixed groups was thus slightly lower than the fitness obtained by nine preprogrammed robots. However, 12 out of the 30 evolutionary runs produced solutions for mixed groups that obtained a higher post evaluation fitness than nine preprogrammed robots (Mann-Whitney, $p < 0.02$).

The average post evaluation fitness of the best mixed group was 403, thus well above the score obtained by a homogeneous group of nine preprogrammed robots.

We also observed collaborated between the six evolved robots and the three preprogrammed robots just like in our previous experiments. For the best solution evolved, the average distance from the center of the nest to each of the preprogrammed robots was 0.58 m, whereas the average distance to each of the evolved robots was 1.23 m.

Conclusions

In this paper, we evaluated how the presence of preprogrammed robots affects the evolutionary process and the behaviors evolved in a multirobot foraging task. We conducted evolutions in which a preprogrammed robot was present and evolutions in which it was absent. Without the preprogrammed robot, three different kinds of foraging behaviors were evolved: one in which robots push prey to the nest, one in which robots continually pickup and drop prey, and one (much more efficient) in which robots pickup and carry prey to the nest.

In the setup in which the preprogrammed robot was present, we only observed the pickup and carry behavior. To increase the rate at which foraging solutions are evolved, we conducted a series of incremental evolution experiments in which the preprogrammed robot initially moved at a lower speed and only after the evolved robots had learned to forage did the preprogrammed robot start to move at normal speed. We applied a similar incremental approach for a mixed group of nine robots. We found that when preprogrammed robots were present, the highest performing evolving robots had learned to collaborate with them: the evolving robots targeted prey far from the nest and dropped them close to the nest for the preprogrammed robots to pickup and deploy in the nest. As a result, the robots occupied different regions of the environment and avoided collisions.

The results demonstrate that robots can be evolved to collaborate with preprogrammed robots. The evolving robots did not adopt neither the preprogrammed solution nor the preprogrammed communication protocol, but instead assumed different roles and collaborated with the preprogrammed robots.

In this study, the preprogrammed robots had a complete solution: they were able to forage on their own. In ongoing work, we are evolving robots to fill in the behavioral gaps between robots preprogrammed with different partial solutions to complex tasks.

Acknowledgements: This study was supported by the FCT grant PTDC/EEACRO/104658/2008.

References

Ampatzis, C., Tuci, E., Trianni, V., Christensen, A. L., and Dorigo, M. (2009). Evolving self-assembly in autonomous homogeneous robots: experiments with two physical robots. *Artificial Life*, 15(4):465–484.

Beer, R. D. and Gallagher, J. C. (1992). Evolving dynamical neural networks for adaptive behavior. *Adaptive Behavior*, 1:91–122.

Cliff, D., Husbands, P., and Harvey, I. (1993). Explorations in evolutionary robotics. *Adaptive Behavior*, 2(1):73.

Şahin, E. (2005). Swarm robotics: From sources of inspiration to domains of application. In *Swarm Robotics Workshop: State-of-the-art Survey*, pages 10–20. Springer-Verlag, Heidelberg, Germany.

Dorigo, M., Trianni, V., Şahin, E., Groß, R., Labella, T., Baldassarre, G., Nolfi, S., Deneubourg, J., Mondada, F., Floreano, D., and Gambardella, L. M. (2004). Evolving self-organizing behaviors for a swarm-bot. *Autonomous Robots*, 17(2):223–245.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA.

Nolfi, S. and Floreano, D. (1998). Coevolving predator and prey robots: Do "arms races" arise in artificial evolution? *Artificial Life*, 4(4):311–335.

Nolfi, S. and Floreano, D. (2000). *Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines*. MIT Press, Cambridge, MA.

Nolfi, S., Floreano, D., Miglino, O., and Mondada, F. (1994). How to evolve autonomous robots: Different approaches in evolutionary robotics. In *Proceedings of the 4th International Workshop on Artificial Life*, pages 190–197. MIT Press, Cambridge, MA.

Quinn, M., Smith, L., Mayley, G., and Husbands, P. (2003). Evolving controllers for a homogeneous system of physical robots: Structured cooperation with minimal sensors. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 361(1811):2321.

Schwefel, H.-P. (1995). *Evolution and Optimum Seeking*. Wiley & Sons, New York.

Trianni, V., Nolfi, S., and Dorigo, M. (2006). Cooperative hole avoidance in a swarm-bot. *Robotics and Autonomous Systems*, 54(2):97–103.

Tuci, E., Ampatzis, C., Vicentini, F., and Dorigo, M. (2008). Evolving homogeneous neurocontrollers for a group of heterogeneous robots: Coordinated motion, cooperation, and acoustic communication. *Artificial Life*, 14(2):157–178.