# Physically Embedded Genetic Algorithm Learning in Multi-Robot Scenarios: The PEGA algorithm

**Ulrich Nehmzow**
Department of Computer Science
University of Essex
GB - Colchester CO4 3SQ
udfn@essex.ac.uk

## Abstract

We present experiments in which a group of autonomous mobile robots learn to perform fundamental sensor-motor tasks through a collaborative learning process. Behavioural strategies, i.e. motor responses to sensory stimuli, are encoded by means of genetic strings[1] stored on the individual robots, and adapted through a genetic algorithm (Mitchell, 1998) executed by the entire robot collective: robots communicate their own strings and corresponding fitness[2] to each other, and then execute a genetic algorithm to improve their individual behavioural strategy.

The robots acquired three different sensor-motor competences, as well as the ability to select one of two, or one of three behaviours depending on context ("behaviour management"). Results show that fitness indeed increases with increasing learning time, and the analysis of the acquired behavioural strategies demonstrates that they are effective in accomplishing the desired task.

## 1. Introduction

### *1.1 Motivation*

For certain applications of autonomous mobile robots — surveillance, cleaning or exploration come immediately to mind — it is attractive to employ multi-robot scenarios. Such tasks are easily divisible between independent robots, and using several robots simultaneously promises a speedup of task execution, as well as more reliable and robust performance.

To determine a suitable control strategy for a mobile robot operating in noisy and possibly dynamic environments requires — if learning mechanisms are to be used — the search through a very large state space. By parallelising this process through the use of several robots and collaborative learning, we aim to accelerate the search and therefore the learning process.

Fixed behavioural strategies, defined by the user, can be used to control robots in such multi-robot scenarios. However, we argue that such an approach will usually be brittle in practice, due to the noisy and partly unpredictable nature of the real world. Instead of using *fixed* behavioural strategies, we suggest that using *learning* controllers will result in more robust control strategies.

### *1.2 Related Work*

That individual (i.e. single robot) learning of sensor-motor competences is possible, has already been demonstrated (Daskalakis, 1991, Mahadevan and Connell, 1991, Nehmzow, 1992, Ramakers, 1993, Colombetti and Dorigo, 1993, Nehmzow, 2000). In a multi-robot scenario, however, it would be beneficial to exploit the fact that several robots are exploring the perception-action space simultaneously, using a *collaborative* learning strategy. Related work with this focus is described, for instance, in (Billard and Dautenhahn, 1999), where a teacher-learner scenario in a multi-robot scenario is presented. In contrast to Billard's and Dautenhahn's work, the experiments presented here do not use a hierarchical teacher-learner scenario, but a collaborative learning strategy in which all robots involved contribute equally to the learning process.

(Watson et al., 1999, Ficci et al., 1999) present a multi-robot learning scenario in which a genetic algorithm (strings are the weights of an artificial neural network controlling the robot) is used to acquire a single competence, phototaxis. Their implementation is similar to the one presented in this paper, in that physical robots perform the task, and modify their strings through communication. Contrary to

[1] Vectors encoding such a behavioural strategy — see figure 2 for an example.

[2] A numerical value indicating to what extent the robot's behaviour achieves pre-defined targets, i.e. a measure of success of the developed behavioural strategy.

our approach, however, only *strings* are exchanged, not fitnesses. Strings are broadcast to the entire population of 8 robots, and the individual robot makes a decision as to whether or not to replace a string with a received one, based on its own fitness alone. Consequently, the resulting convergence times are quite long. After more than 80 minutes the developed ability to move towards a light source matches that of a hand-coded program.

(Lee et al., 1998) present a *simulate-and-transfer* mechanism, in which a controller is developed using a genetic algorithm on a simulator, and the solution is afterwards transferred to a physical robot. The single acquired task is that of pushing a box towards a light source.

(Mataric, 1998) presents experiments with two legged robots that learn to push a box collaboratively towards a goal position, using radio communication in the process. Robots learned a reactive mapping between sensory perceptions and pre-installed fixed behaviours, by means of reinforcement learning. Communication was used to overcome sensory limitations (for instance caused by obstructed views of one robot). In contrast to this use of communication to overcome hidden state problems, in the experiments presented here communication was used to establish a "global" learning process across robots, in which the individual robot's exploration of sensor-motor space was used to increase the fitness of the entire robot group.

There are also software simulations of multi-robot scenarios that employ genetic algorithms to acquire behavioural policies. (Wilson, 1998) for instance developed a system for the RoboCup domain, in which simulated robots learn to move towards the ball by means of a genetic algorithm. Whether such algorithms are applicable to real world robot scenarios, however, remains an open question.

Learning in the experiments presented here is accomplished by means of a genetic algorithm. As this algorithm is physically embedded on real robots, rather than executed by software agents, we refer to it as a physically embedded genetic algorithm (PEGA).

## 2. Experimental Setup

### 2.1 Hardware

All experiments reported here were conducted using two small mobile robots, which were equipped with sonar, infrared (IR), ambient light and tactile sensors (see figure 1). Most importantly for the experiments discussed in this paper, the robots were also able to communicate with each other by means of infrared sensors. Communication speeds were low (not more than 200 Baud), and possible if the robots were within 1 metre of each other.

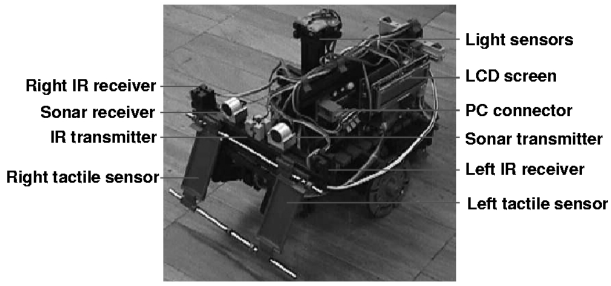


Figure 1: One of the mobile robots used in the experiments.

### 2.2 Software

For all initial experiments, described in section 3., the robots were equipped with the pre-installed competences of obstacle avoidance (using IR and tactile sensors), seeking the other robot, and communication via IR with the other robot. This aided the process of acquiring a further competence through the physically embedded genetic algorithm.

In the later experiments described in section 4., more competences were acquired through the PEGA, and fewer were implemented by the designer.

In addition to these pre-installed competences, each robot had an on-board implementation of a genetic algorithm. This included the fitness evaluation, mutation, crossover and executing the control strategy encoded by the string.

In software implementations of genetic algorithms, the population size is chosen large, each software agent executing one particular string. In a physical implementation this is not easily done: too many robots will eventually interfere with each other so much, that a principled exploration of the action search space is impossible. In our implementation, each robot carried two strings (i.e. behavioural strategies): the "active" one that was being evaluated by the GA, and the "best" one found so far. The latter was used as a fallback option if the GA did not produce a "better" solution than the "best" one so far. "Better" was defined here by a higher fitness value of the evaluated string.

### 2.3 Experimental Procedure

The experimental procedure adopted in all experiments was this: the two robots were left to execute the behavioural strategy encoded by the current string for a certain amount of time, evaluating the fitness while they were doing this. After the alloted time had expired, the robots initiated a search behaviour (based on infrared signal emissions) to locate the other robot. This ensured that robots were within communication distance of one another. Once found, robots faced each other and exchanged their current strings and corresponding fitnesses.

### 2.3.1 Implementation of the PEGA

**Crossover.** After the robots had exchanged the strings and their corresponding fitnesses, crossover and mutation were applied in the following manner. If the received remote string was fitter than the local one, one half of the local string (either the first or the second half, determined by a random process), was replaced by the corresponding part of the remote string.

If the local string had a higher fitness than the remote one, no crossover between local and remote string happened. However, if in this case the locally stored "best" string had a higher fitness than the currently used local string, there was a 30% probability that a randomly selected bit (binary digit) of the current string was replaced by the corresponding bit of the "best" string. In the remaining 70% of cases, mutation was applied to the current string.

**Mutation.** If invoked, the likelihood of mutation of a string was dependent upon a string's fitness. This probability $p_m$ of changing one randomly selected bit is given by $p_m = R\frac{100-F}{100}$, where $F$ stands for the fitness of the string, and $R$ is a constant chosen to be $R = 0.3$.

## 3. Experiments: Acquisition of Single Competences

### 3.1 Phototaxis

The first task that the robots were to learn, using the PEGA algorithm, was to move towards a light source placed at a random location in the experimental arena.

### 3.1.1 Phototaxis: Implementation of the Genetic Algorithm

The string used to represent the robot's behavioural repertoire is shown in figure 2. It consists of four elements, each containing a motor response for the case that the brightest light in the environment is detected at the front, rear, right or left of the robot respectively. Each of these fields contains one of three possible motor responses (forward, left or right).

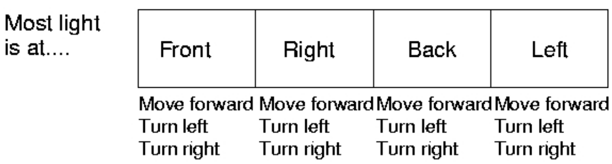


Figure 2: Learning phototaxis: String representation.

The fitness $F_P$ of a string was determined by $F_P = \frac{L}{T}100$, with $L$ being the number of program loops during which the robot was facing the brightest light, and $T$ being the total number of program loops processed.

### 3.1.2 Phototaxis: Experimental Results

Ten experiments were conducted, each over 30 generations (where one generation denotes one modification to a string). The change in fitness over those 30 generations in each of the 10 runs is shown in figure 3, a clear upward trend with time is visible. Figure 4 shows the average fitness of those ten runs against learning time. The convergence of solutions is visible from the decreasing standard deviations with increasing learning time.

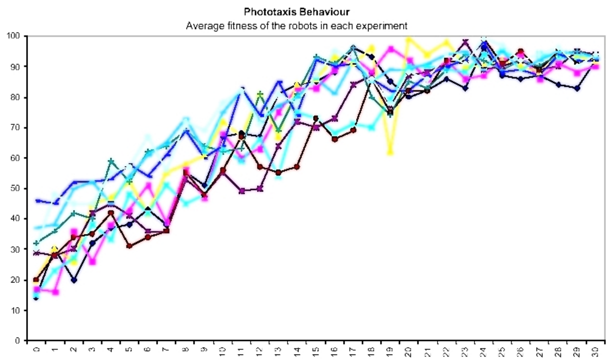


Figure 3: Learning phototaxis: Individual fitnesses (in %) of 10 separate runs against training time (generations). Total learning time: 30 generations.

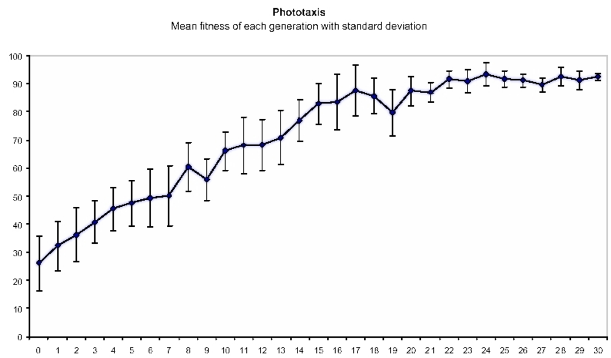


Figure 4: Learning phototaxis: Mean fitness (in %) and standard deviation of ten runs against learning time (generations). Total learning time: 30 generations.

Table 1 shows the frequencies of acquired sensor-motor pairs. It is clear from that table that a successful phototaxis competence was acquired.

### 3.2 Obstacle Avoidance

The second single competence to be acquired was the ability to steer away from obstacles by means of the robot's infrared sensors.

| Sensor Input | Forward | Left | Right |
|---|---|---|---|
| Strongest light ahead | 19 | 1 | 1 |
| Strongest light on left | 0 | 20 | 0 |
| Strongest light on right | 1 | 2 | 17 |
| Strongest light behind | 0 | 17 | 3 |

Table 1: Learning phototaxis: Frequency of acquired input-output associations.

### 3.2.1 *Obstacle Avoidance: Implementation of the Genetic Algorithm*

The string used to encode the behavioural strategy of the robot is given in figure 5. It consists of four fields. representing the motor response (forward/left/right) in case IR sensors were triggered to the left *and* right, left, right, or not at all.

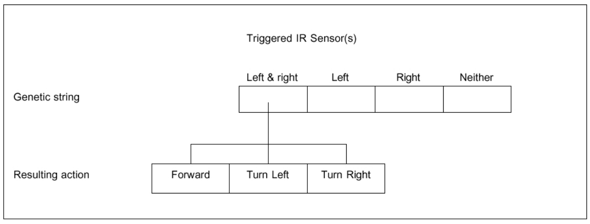


Figure 5: Structure of the string to achieve obstacle avoidance.

The fitness $F_A$ of a string was determined by $F_A = \frac{A}{T}100$, with $A$ being the number of program loops during which neither the robot's left IR nor its right IR detected an object nearer than a preset threshold, and $T$ being the total number of program loops processed.

### 3.2.2 *Obstacle avoidance: Experimental Results*

Again, 10 runs of 30 generations each were executed. The fitnesses of individual runs are given in figure 6. The noticeably low fitness of one of the 10 runs was due to a hardware fault of one IR sensor, which was only detected after the experiment was concluded.

The average fitness of all 10 runs against learning time is given in figure 7, the acquired sensor-motor pairings are shown in table 2. Again, clear upward trends in fitness with decreasing standard deviations are shown, as is a successful behavioural strategy.

## 3.3 *Robot seeking*

In order to communicate with one another, the robots need to face each other, and need to be less than 1 metre apart. The robots detect that this is the case, i.e. that communication is possible, by exchanging initialisation messages through their infrared sensors. Once communication is possible, a fixed, error correcting protocol is used to exchange information (Nix, 2000).

For the first two experiments just discussed — phototaxis and obstacle avoidance — a fixed robot-seeking behaviour was implemented, and the purpose of the following experiment was to establish a robot-seeking competence through the PEGA.

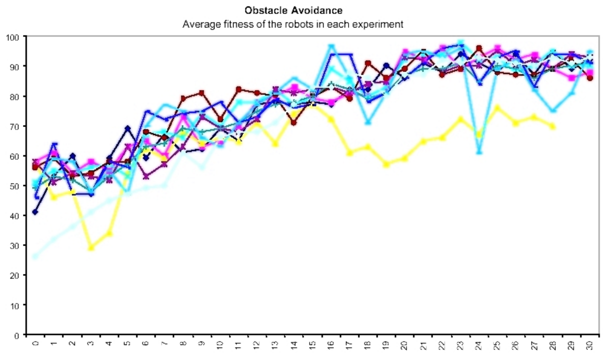


Figure 6: Individual fitnesses (in %) of 10 runs against number of generations learning time when learning obstacle avoidance.

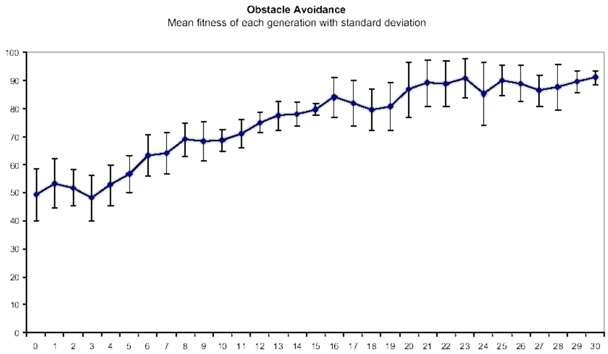


Figure 7: Mean fitness (in %) and standard deviation of ten runs against learning time (in generations) when learning obstacle avoidance, against learning time.

### 3.3.1 *Robot seeking: Implementation of the Genetic Algorithm*

The string used to acquire the robot-seeking competence is given in figure 8. This string is largely identical to the one used for obstacle avoidance (figure 5), with the difference that the case of "no IR reflection detected" is mapped onto a collection of four actions. When invoked, these four actions are executed cyclically in sequence, until an IR signal is detected.

The fitness $F_{RS}$ was determined by $F_{RS} = 100 - T$, with $T$ being the amount of time that it takes the robot to position itself correctly with respect to the other robot.

| Sensor Input | Forward | Left | Right |
|---|---|---|---|
| Left IR triggered | 0 | 1 | 19 |
| Right IR triggered | 2 | 18 | 0 |
| Left & right IR triggered | 1 | 8 | 11 |
| None triggered | 19 | 1 | 0 |

Table 2: Obstacle avoidance: frequencies of acquired sensor-motor mappings.

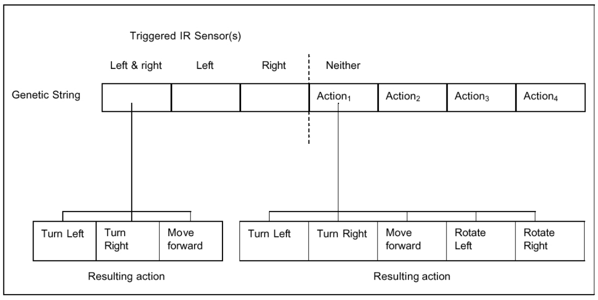


Figure 8: String used to acquire robot-seeking behaviour.

### 3.3.2 *Robot seeking: Experimental Results*

Figure 9 gives the fitness against time of individual runs, figure 10 the mean fitness. Again, there is a noticeable increase in fitness over time, comparable to that observed in the obstacle avoidance experiment (bearing in mind that only fitness change per time unit, not absolute values can be compared, due to the task-specific definition of fitnesses).

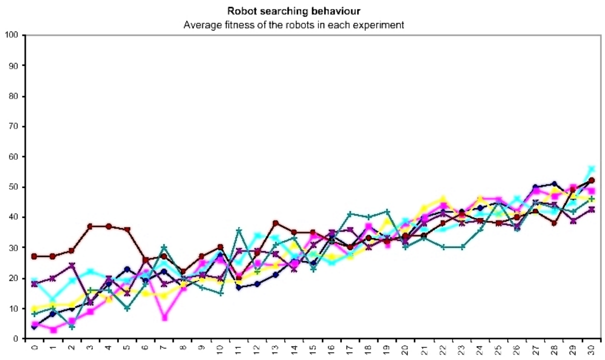


Figure 9: Learning to find other robots: Fitnesses (in %) of ten individual runs against learning time (in generations).

## 4. Experiments: Managing Multiple Competences

Given the evidence that acquisition of single competences is indeed possible, using a physically embedded genetic algorithm, we were interested to investigate whether *multiple* competences could be acquired by the same mechanism. In the first experiment on multi-competence learning, the robots' task was to learn when to select one of two preprogrammed behaviours: either obstacle avoidance, or phototaxis.

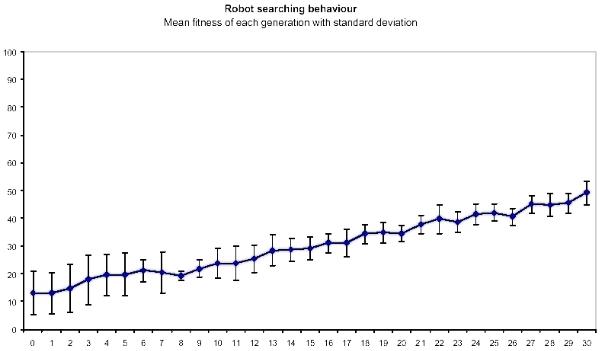


Figure 10: Learning to find other robots: Mean fitnesses (in %) and standard deviation of ten runs, against learning time (in generations).

### 4.1 *Action Selection: Phototaxis and Obstacle Avoidance*

#### 4.1.1 *Action Selection: Implementation of the Genetic Algorithm*

Figure 11 gives the string used to acquire this competence. The four sensory fields each encode whether in the particular sensory situation the robot should select the obstacle avoidance action, or the phototaxis action.

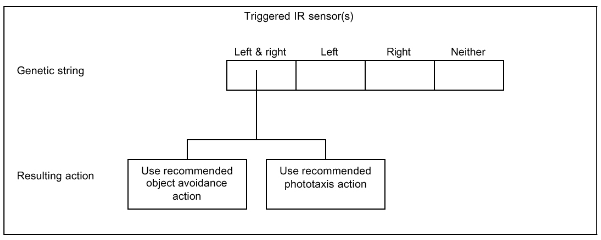


Figure 11: String used to select either phototaxis or obstacle avoidance behaviour.

#### 4.1.2 *Experimental Results*

Figure 12 shows the fitness of 10 individual runs against learning time, figure 13 shows the mean fitness.

Table 3 shows an interesting result: almost always the phototaxis strategy was selected, irrespective of sensory perception. The post-hoc explanation was that the experiment were carried out in an environment in which approaching an attractor automatically meant avoiding obstacles! Obvious future experiments are to conduct the same investigation in differently structured environments.

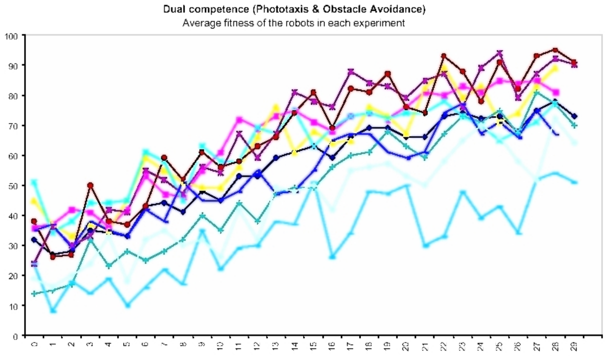


Figure 12: Selecting phototaxis or obstacle avoidance: Fitnesses (in %) of ten individual runs against learning time (in generations).

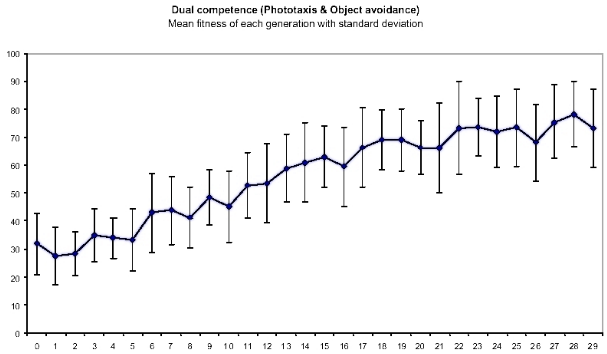


Figure 13: Selecting phototaxis or obstacle avoidance: Mean fitnesses (in %) and standard deviation of ten runs, against learning time (in generations).

## *4.2 Phototaxis, Obstacle Avoidance and Robot Seeking*

The final set of experiments, concerning the acquisition of a triple competence of phototaxis, obstacle avoidance *and* robot seeking, consisted of two separate experiments. In the first ("phase 1"), the robot had to acquire the ability to manage behaviours, and to select one of three behaviours, depending on context (these were the competences acquired in the experiments described in sections 3.1, 3.2 and 3.3). In the second ("phase 2"), the robots attempted to learn the three competences simultaneously, while using a pre-supplied management strategy.

Action selection was achieved by means of a behaviour managing module (see figure 14), which selected the competence to be used to drive the robot.

In phase 1, the three previously learned competences were pre-supplied, and the PEGA had to learn the management strategy. In phase 2, a fixed management strategy was supplied, the three competences were acquired through the PEGAs described in sections 3.1, 3.2 and 3.3.

| Sensor Input | Object avoid. | Phototaxis |
|---|---|---|
| Left IR triggered | 2 | 18 |
| Right IR triggered | 1 | 19 |
| Left & right IR trigg. | 7 | 13 |
| None triggered | 0 | 20 |

Table 3: Managing phototaxis and obstacle avoidance: Sensor-motor mappings acquired.

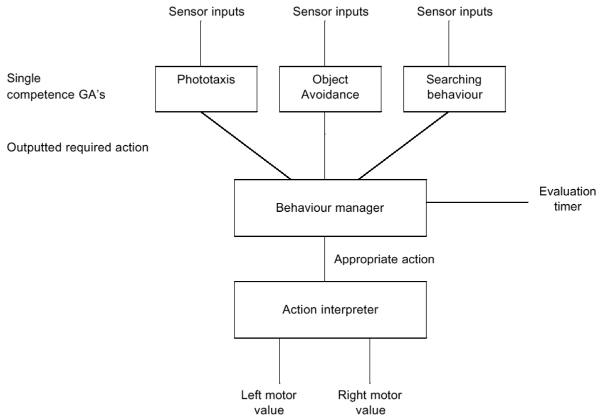


Figure 14: Behaviour manager for triple-competence acquisition.

### *4.2.1 Triple Competence: Implementation of the Genetic Algorithm*

Figure 15 shows the string used in both phases of the experiment. The leftmost four fields determine the robot's behaviour during the first 60 seconds of a generation, the rightmost four entries the strategy for the next 100 seconds. If after those 160 seconds no other robot is detected, a pre-supplied seek behaviour is initiated.

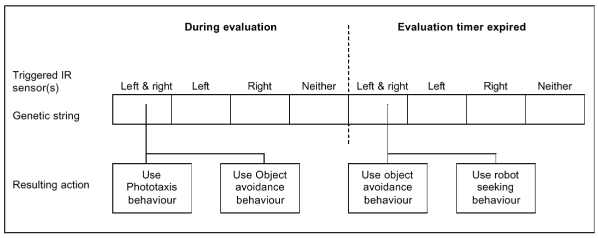


Figure 15: String used to manage phototaxis, obstacle avoidance and robot-seeking behaviour.

The fitness in this final set of experiments was determined as the average fitness of the three individual behaviours.

### *4.2.2 Experimental Results*

Individual fitnesses of 5 runs and the average fitness in phase 1 are shown in figures 16 and 17.

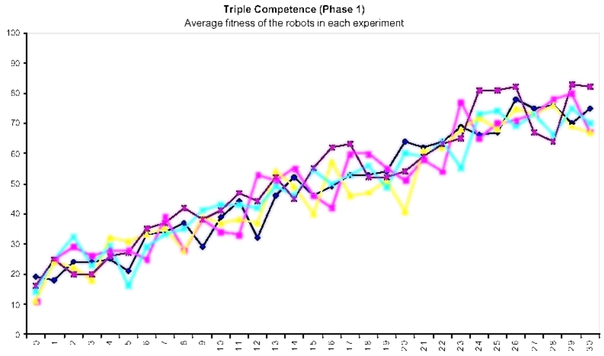


Figure 16: Selecting phototaxis, obstacle avoidance or robot-seeking (experiment 1): Fitnesses (in %) of ten individual runs against learning time (in generations).

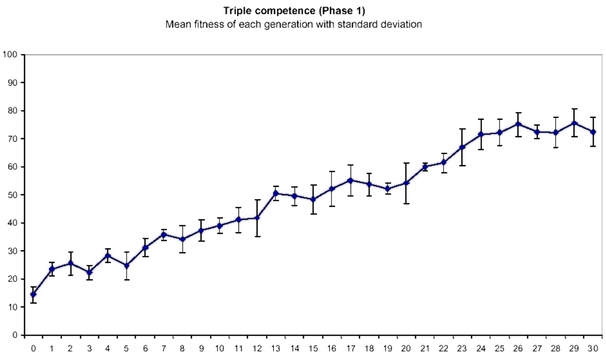


Figure 17: Selecting phototaxis, obstacle avoidance or robot-seeking (Experiment 1): Mean fitnesses (in %) and standard deviation of ten runs, against learning time (in generations).

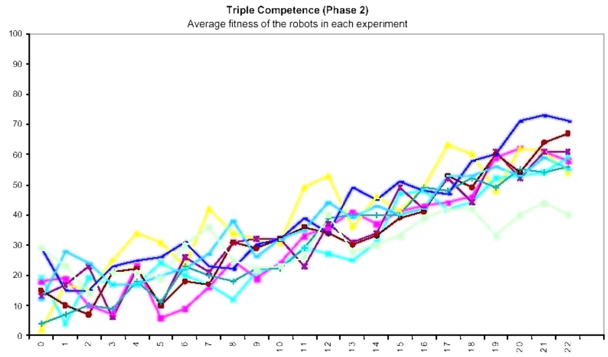


Figure 18: Learning phototaxis, obstacle avoidance and robot-seeking (experiment 2): Fitnesses (in %) of ten individual runs against learning time (in generations).

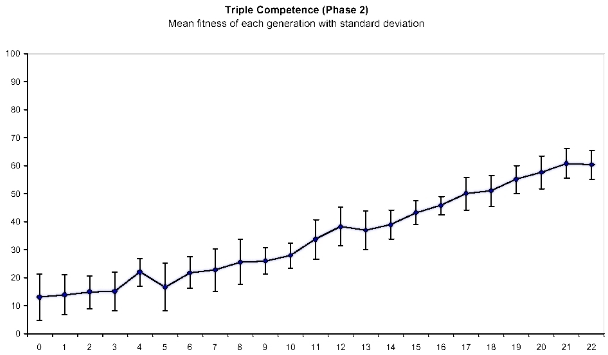


Figure 19: Learning phototaxis, obstacle avoidance and robot-seeking (experiment 2): Mean fitnesses (in %) and standard deviation of ten runs, against learning time (in generations).

Likewise, individual fitnesses of 10 runs and the average fitness in phase 2 are given in figures 18 and 19. In all cases, fitnesses increase with learning time, and the robots exhibited successful behaviour.

## 5. Summary and Conclusion

### *5.1 Motivation and Context of this Work*

For certain real world applications of mobile robots — especially those that are easily divisible into subtasks — it is beneficial to let groups of robots perform the task concurrently. Examples of such tasks are cleaning, surveillance, or exploration.

Due to the partial unpredictability of the real world and the noise inherent to it, it is also desirable to use *learning* controllers, which are (because of their ability to adapt to real world circumstances) less brittle and more robust than pre-coded, fixed control algorithms.

Learning in individual robots has previously been achieved (Daskalakis, 1991, Mahadevan and Connell, 1991, Nehmzow, 1992, Ramakers, 1993, Colombetti and Dorigo, 1993, Nehmzow, 2000), but learning in multi-robot scenarios has only recently received attention. Work carried out at Brandeis University (Watson et al., 1999, Ficci et al., 1999) is the closest to work presented here, in terms of experimental procedure — the main difference being the number of competences being learned (phototaxis only in (Watson et al., 1999, Ficci et al., 1999), five different competences in the experiments reported here), and the learning time needed (2hrs versus about 20 to 30 minutes in the work reported here). (Lee et al., 1998) present a simulate-and-transfer GA approach to box pushing.

(Billard and Dautenhahn, 1999) have presented multi-robot learning scenarios in which a teacher conveys knowledge to a learner robot. In a teacher-learner scenario the emphasis is on *propagating* knowledge. However, there is another interesting aspect to multi-robot situations, which we aimed to exploit in the experiments presented here. A robot interacting with its environment is in essence exploring a sensor-motor space, in which it tries to establish meaningful mappings between perception and action, with respect to a given task. Multiple robots can obviously explore this sensor-motor space quicker, and it would therefore be interesting to devise a learning scenario in which all robots contribute equally to the learning process, by pooling their knowledge. The way we aimed to achieve this goal was to implement a physically embedded genetic algorithm on each of a group of robots.

## *5.2 The Physically Embedded Genetic Algorithm (PEGA)*

### *5.2.1 Experimental setup*

For the experiments reported in this paper, two autonomous mobile robots were used (see figure 1). Besides sensory capability to perceive their environment, these robots were able to communicate with each other by means of infrared transmissions.

Each robot was controlled by a genetic string that encoded the robot's behavioural policy in response to certain sensory situations. Both robots executed their behavioural policy for a certain amount of time, at the same time determining the "fitness" (i.e. the quality of performance of the given task) of their current string. They then communicated their strings and corresponding fitnesses to each other, and used a genetic algorithm to modify their strings in an attempt to achieve a higher fitness in the next "generation" (i.e. the next round of the abovementioned cycle).

### *5.2.2 Differences between conventional GAs and the PEGA*

While the PEGA uses strings to encode behavioural policy, and crossover and mutation operators like a standard genetic algorithm (GA), there are some important differences between the PEGA and a GA. Typically, a GA uses a large number of agents (several hundreds), each carrying one string. For real world robot scenarios, it is not feasible to use hundreds of robots, because the physical interaction of robots with one another would eventually interfere with task-oriented behaviour (apart from more mundane reasons such as sensor crosstalk or cost). The search for better policies is therefore more difficult in the PEGA.

Although the task each robot performs is essentially a single-robot task, we believe — without being able to demonstrate this at this stage — that through the multi-robot learning scenario we achieve a faster exploration of the search space than with a single robot; and that therefore the individual robot benefits from being part of a society. The question of speedup in learning in multi-robot scenarios, in contrast to single-robot learning scenarios, is an interesting aspect of future research.

The way we overcame the difficulty of searching for better control policies in the experiments presented here, was to employ a mixture of fixed, pre-installed behaviours and an adaptive GA component. The fixed behaviours effectively reduced the search space and thus aided the search process for better policies.

Besides the use of pre-installed behaviours, each robot also carried two strings, rather than one as in standard GAs. The "current" string was the one actually being used, but a copy of the "best string so far" was carried as a fallback position. In case the GA resulted in a lower fitness, robots were able to resort to the "best so far" string, preventing a fall in fitness.

## *5.3 Results*

Single sensor-motor competences as well as behaviour-management competences were acquired by our robots, using the PEGA. In a first set of three experiments, the robots acquired the single competences of phototaxis, obstacle avoidance and robot-seeking. In a second set of experiments, the robots learned to select between phototaxis or obstacle avoidance behaviour, depending on context, and in a third set of two experiments the robots learned to manage the triple competence of phototaxis, obstacle avoidance and robot seeking.

In all experiments, a marked increase in fitness with increasing learning time was noted. In all cases, the robots also acquired a task-achieving competence.

There were differences between tasks, some fitness curves rising faster than others, which indicates i) that there are tasks that are better suited to the PEGA than others, and ii) that there might be tasks that cannot be accomplished by the PEGA at all. Future work would have to address this question, by scaling up the complexity of the tasks, the number of tasks acquired simultaneously, and the complexity of the environment the robots operate in.

## *5.4 Conclusion*

The main point that the experiments reported in this paper make is that physically embedded learning in robot colonies *is* possible, in real time, using simple hardware and a simple learning strategy — that of a physically embedded genetic algorithm.

Given the simplicity of the algorithm, robot programming was straightforward and required only a moderate amount of programming time. The advantages gained by using a learning controller, rather than a "hardwired" one, however, are numerous: adaptation to changing environments, robustness in the presence of sensor noise and contradictory sensor information, and the acquisition of behavioural strategies that would not have been obvious to a human designer, due to the fact that our view of the world differs from that of a robot. In addition to this, using a global learning algorithm that is distributed over a colony of robots further advantages: potentially faster exploration of state space, faster acquisition of competences, and robustness (e.g. the failure of one member of a colony will not result in any drastic loss of competence).

### *5.5 Future Work*

The experiments with the PEGA reported in this paper are just a starting point to further research. We believe that the following aspects are particularly interesting for future work.

We have used a "colony" of two robots. Clearly, experiments with a larger number of robots need to be conducted to investigate whether the expected benefits of this distributed learning algorithm will actually materialise in larger colonies. The question of optimal colony size (cooperation versus interference) also deserves attention.

In our experiments, the robots acquired either sensor-motor competences or the ability to select between different behaviours. An obvious extension to these experiments is to let the robots acquire *both* behaviours and management strategies.

Finally, it would be interesting to investigate the question of scaling these experiments up, by increasing the state space of the robots (more sensor modalities, higher resolution, more complex tasks and environments).

### *Acknowledgements*

This project was carried out in collaboration with my colleague Jonathan Shapiro of Manchester University, the experimental work reported in this paper was carried out by our project student Mark Johnson (Johnson, 2001), and I acknowledge their contribution with thanks.

The experiments were conducted at the University of Manchester, which provided the research facilities used in this research.

## References

Billard, A. and Dautenhahn, K. (1999). Experiments in learning by imitation - grounding and use of communication in robotic agents. *Adaptive Behavior*, 10(3/4).

Colombetti, M. and Dorigo, M. (1993). *Robot Shaping: Developing Situated Agents through Learning.* Technical Report Nr. 40, International Computer Science Institute, Berkeley CA.

Daskalakis, N. (1991). *Learning Sensor-Action Coupling in Lego Robots.* MSc thesis, Department of Artificial Intelligence, University of Edinburgh.

Ficci, S., Watson, R., and Pollack, J. (1999). Embodied evolution: A response to challenges in evolutionary robotics. In Wyatt and Demiris, (Eds.), *Proc. 8th European workshop on learning robots.*

Johnson, M. (2001). *Physically embedded genetic algorithms on mobile robots.* BSc Thesis, Department of Computer Science, University of Manchester.

Lee, W., Hallam, J., and Lund, H. (1998). Learning complex robot behaviours by evolutionary computing with task decomposition. In Wyatt and Birk, (Eds.), *Proc. 6th European workshop on robot learning.* Springer LNAI 1545.

Mahadevan, S. and Connell, J. (1991). Automatic programming of behavior-based robots using reinforcement learning. In *Proc. 9th National Conference on Artificial Intelligence.*

Mataric, M. (1998). Using communication to reduce locality in distributed multiagent learning. *JETAI*, 10(3):357–369.

Mitchell, M. (1998). *An Introduction to Genetic Algorithms.* MIT Press.

Nehmzow, U. (1992). *Experiments in Competence Acquisition for Autonomous Mobile Robots.* PhD thesis, Department of Artificial Intelligence, University of Edinburgh.

Nehmzow, U. (2000). *Mobile Robotics: A Practical Introduction.* Springer Verlag.

Nix, C. (2000). *A communication protocol for cooperating mobile robots.* BSc Thesis, Department of Computer Science, University of Manchester.

Ramakers, W. (1993). *Investigation of a Competence Acquiring Controller for a Mobile Robot.* Licentiate Thesis, Vrije Universiteit Brussel.

Watson, R., Ficci, S., and Pollack, J. (1999). Embodied evolution: Embodying an evolutionary algorithm in a population of robots. In Michalewicz, Schoenauer, Yao, and Zalzala, (Eds.), *Proc. Congress on Evolutionary Computation.*

Wilson, P. (1998). *Development of Genetic Programming Strategies for use in the RoboCup Domain.* Honours thesis, Department of Computer Science, Royal Melbourne Institute of Technology.