1 Introduction and Motivation

Motivation. For certain applications of autonomous mobile robots — surveillance, cleaning or exploration come immediately to mind — it is attractive to employ several robots simultaneously. Tasks such as the ones mentioned above 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.

For any robot operating in the real world, the question of how control is to be achieved is of prime importance. While fixed behavioural strategies, defined by the user, can indeed be used to control robots, they tend to be brittle in practice, due to the noisy and partly unpredictable nature of the real world. Therefore, instead of using fixed and pre-defined control procedures, learning is an attractive alternative.

To determine a suitable control strategy for a mobile robot operating in noisy and possibly dynamic environments through learning requires a search through a very large state space. By parallelising this process through the use of several robots and collaborative learning, this learning process can be accelerated.

A physically embedded GA (PEGA). In this paper, we present experiments conducted with two communicating mobile robots. Each robot’s control policy was encoded through a genetic string. By communicating genetic strings and fitnesses to one another at regular intervals, robots modified their individual control policy, using a genetic algorithm (GA). Contrary to common GA approaches, we did not use a simulate-and-transfer method, but implemented the GA directly on the robots.

We were able to show that the following competences can all be acquired using the PEGA approach:

1. Phototaxis
2. Obstacle avoidance
3. Robot seeking (for communication purposes)
4. Phototaxis and obstacle avoidance
5. Phototaxis, obstacle avoidance and robot seeking.

2 Experimental Setup

2.1 Mobile Robot Hardware

All experiments were conducted using two small mobile robots, equipped with sonar, infrared, ambient light and tactile sensors (see figure 1). Most importantly, the robots were also able to communicate with each other by means of infrared sensors.

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

2.2 PEGA Software

For the first three experiments mentioned in section 1, the robots were equipped with the following pre-installed competences: Obstacle avoidance, using IR and tactile sensors, seeking the other robot, and communication via IR with the other robot.

In the 4th and 5th experiments, more competences were acquired through the PEGA, and fewer were implemented by the designer.

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

In our PEGA, 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 string population is therefore two). The “best string so far” was used only 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.
Experimental procedure. 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 allotted time had expired, the robots initiated a search behaviour (based on infrared signal emissions) to locate the other robot. Once found, robots faced each other and exchanged their current strings and corresponding fitnesses through infrared communication.

2.2.1 Implementation of the PEGA

Crossover. After the exchange of strings and 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 (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 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 $R = 0.3$ is a constant.

3 Experimental Results

The robots acquired all five competences listed in section 1 completely, within a few tens of generations. Average fitness versus the number of generations are shown in figures 2 to 5; full details are given in [Nehmzow 2002].

4 References, background and details

Full details, including related work, detailed description of the algorithm, experimental procedure, results in detail and a discussion can be found in


Acknowledgements This project was carried out in collaboration with my colleague Jonathan Shapiro of Manchester University — I acknowledge his contribution with thanks. The experimental work reported in this paper was carried out by our project student Mark Johnson, and I acknowledge his contribution gratefully.