Distributed evolution of behaviour for a group of social autonomous agents

Fredrik Samuelsson
Chalmers University of Technology
d4sama@dtek.chalmers.se

Peter Nordin
Physical Resource Theory
Chalmers University of Technology
nordin@fy.chalmers.se

Abstract

In nature many tasks are performed by co-operation and mating between individuals. In an effort to mimic nature, experiments to analyse how behaviour evolve in a group of small social mobile robots have been made.

Real physical robots have been used for the experiments. Lego Mindstorms has been used as hardware platform.

Robots migrate genetic individuals between each other using infrared communication.

Robots can learn from each others' success or failure by migration of fit individuals.

Linear genetic programming has been used through out the experiments.

1 Introduction and Background

This paper is based on a Master Thesis [1] by Fredrik Samuelsson supervised by Peter Nordin.

A group of mobile robots exchange individuals between each other and thus can learn from each others' mistakes.

The hardware platform used is Lego Mindstorms v1.0 [2] and the software platform used is LegOS v0.2.4 [3].

Each robot is equipped with a single distance sensor in front, using modulated infrared light to measure the distance to objects up to 20 cm in front of the robot.

The hardware used here is very limited in processor speed and amount of memory compared to other robot platforms such as the Khepera, used in other GP experiments [4]. Even relatively simple problems tend to take long time for a single robot to learn. The idea is to compensate for this by parallel execution on several communicating robots. To see if relatively dumb robots can perform better when they work together.

In these experiments the robots only communicate with each other using migration of individuals.

2 Problem

The task to solve for these robots is of a relatively simple nature, they only need to move fast and straight and avoid getting to close to other objects such as walls or other robots. Since the only input to a robot is the distance to objects in front of it, the robots can not learn complex tasks at this point. However if fitted with more sensors they could probably evolve cooperative behaviour such as moving objects to heavy for a single robot, foraging [5] etc.

3 Method

3.1 Genetic Programming

Linear genomes for a register machine [6] have been used in these experiments. Each individual consists of a few lines of code for a register machine that is interpreted by a genetic engine on the robots. The distance to nearby objects in front of the robot measured by the distance sensor is used as input to the individual and the output of the individual is used to set the speed of the two motors. Tournament selection with tournament size of four individuals have been used.

The fitness function is constructed to promote individuals that move the robot straight and fast without hitting objects. The only way to get maximum fitness (1) is to move the robot straight forward with maximum speed. The lowest possible fitness is 0.

At certain intervals individuals migrate: a robot broadcasts the fittest individual in last tournament to all other robots. The imported individual replaces the least fit individual in last tournament on the receiving robot.

3.2 Basic motion

The robot is constructed with two wheels (left and right) and two supports (rear and front) allowing it to move in six basic ways: forward, backward, turn while moving forward, turn while moving backward, rotating and still. Each individual can learn to move the robot in one or more of these ways.

If an individual learns to control the robot in more than one of the basic motions, it has learned to use the sensor, since the sensor value is the only input to the individual that can change.

3.3 Statistics

Every robot send statistical information to a computer over IR-link at certain intervals. This include the fittest individual in last tournament together with it's fitness. Used to produce figures 3, 6 etc.

The entire population is exported to the computer and used to determine the behaviour of individuals as seen in figures 1, 4 etc. The number of behaviours per individual is also computed from these exported populations, as seen in figures 2, 5 etc.

4 Results

Experiments done with groups of robots that communicate show that learning generally is more stable and fast than with a single robot. Robots that get stuck in corners tend to evolve rotating behaviour while those on mid-floor tend to move fast and straight. For a group of robots these behaviours can be evolved at the same time, given that enough robots are in different positions.

4.1 Behaviour and migration rate

The evolved behaviours of individuals are more uniform with higher migration rate. Compare figure 7 where 25% of the population is migrated per generation, and figure 13 where 5% of the population is migrated every generation.

On the other hand, to much migration can slow down the evolution. Compare figure 9 and 15. To some part this can be explained by the extra time needed to migrate individuals over the IR-port, but also because unfit individuals migrate and can replace fitter ones.

Generally fitness is improved when migration is at “normal” rates. Compare figure 18 with figure 12. And figure 6 with figure 18.

4.2 Behaviour and execution time

When individuals are given short execution time, they tend to learn suboptimal movements such as turning while moving forward, see figure 16 in contrast to when given longer execution time, see figure 4 where individuals use straight forward motion.

This suboptimal behaviour can be helped with migration, as seen in figure 10.

4.3 Competitive and cooperative evolution

The normal evolution for a population in these experiments is competitive evolution where all individuals try to become as fit as possible by evolving several behaviours per individual.

With short execution time for each individual, the individuals tend to evolve in cooperative evolution. Each individual evolves only one behaviour and the robot moves reasonably well with the entire population. This does not happen very often, but one example is found in figures 1 to 3.

In figure 2 the individuals are only given 250ms to control the robot, which is just enough to make it move a few centimeters. The individuals evolve to use only one behaviour each, and thus have not learned to use the sensor input. The population represents how the robots environment is configured, as seen in figure 1 where the forward motion is in clear majority since the environment for this test was a relatively large open area surrounded with walls. Enough individuals have learned to rotate or turn the robot to make it come clear when stuck in corners. Since no individuals use the sensors, the robot hits the walls, and only by chance make a turn to come free.

4.4 Explanation of figures

4.4.1 Column

Each column show three figures for one experiment. Briefly explained below each figure.

Experiment 1 is a single robot where each individual is allowed to execute for 250 ms. This particular robot's population evolves in a cooperative behaviour as explained above.

4.4.2 Row

Each row show the same type of figure:

In the first row "Basic behaviours" there are six possible behaviours where a few have been marked clearly, those not marked should be possible to distinguish by exclusion. E.g. in figure 16 "turn while moving forward", "turn while moving backwards" and "still" have been marked. Thus showing that "forward" is not used by any individuals resulting in poor fitness as seen in figure 18. While in figure 7 only "rotating" have been marked and can be seen to be the only one to decrease to a minimum level. Thus, the rest of the six basic motions are used.

It is easy to evolve a behaviour where the robot is still for some input values, e.g. when changing from forward to backward motion, at a certain point the motion will be classified as "still". Thus the "still" behaviour is more common in these figures comparing to how the robot actually moves.

The second row of figures "Number of behaviours per individual" shows at what point the individuals learn to use the sensor to change the motion of the robot. E.g. in figure 5 a majority of the individuals have more than one behaviour after 15 generations and almost all individuals have more than one behaviour after 30 generations.

The third row of figures "Mean fitness" shows the mean fitness values for the individuals. The robots send their fittest individual in last tournament together with fitness value to a statistics engine over their IR-port at intervals, and the figures show the mean values over ten samples for these.

5 Conclusions

In these experiments we show that slow hardware can be compensated with parallelism. The experiments performed show that tasks are performed better and faster when robots communicate.

More complex and interesting tasks can be performed given more input to the robots by adding more sensors.

If robots were able to locate other robots they could evolve self organizing behaviour to complete tasks not possible by a single robot.

The system used for these experiments is relatively cheap and scalable. More robots can be added to solve a more difficult task.

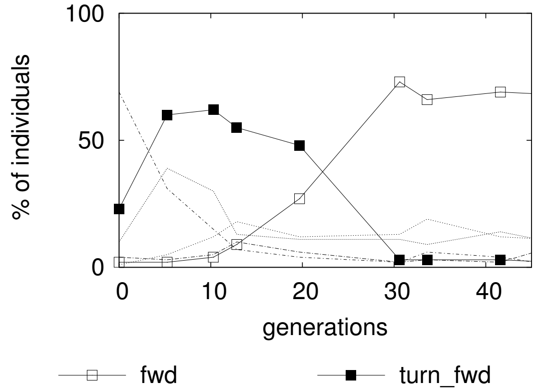

Figure 1: Experiment 1. Basic behaviours for all individuals. Execution time 250 ms per individual. Single robot.

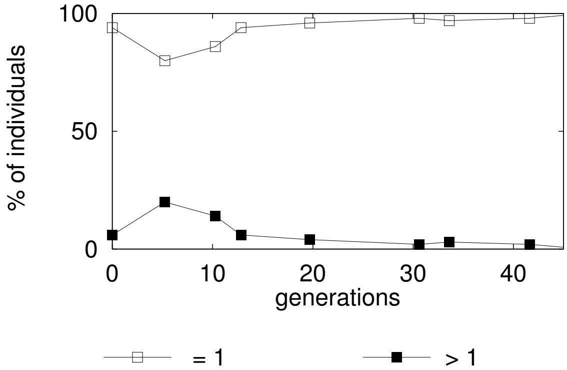

Figure 2: Experiment 1. Number of behaviours per individual. Execution time 250 ms per individual. Single robot.

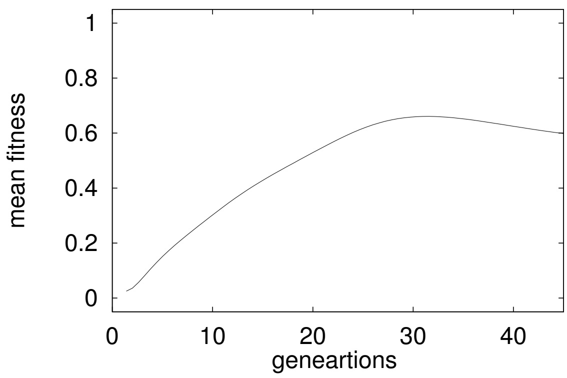

Figure 3: Experiment 1. Mean fitness. Execution time 250 ms per individual. Single robot.

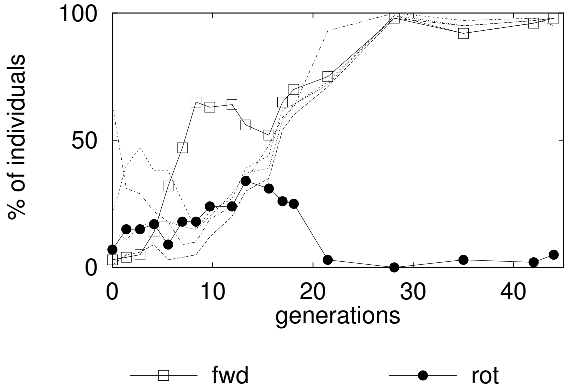

Figure 4: Experiment 2. Basic behaviours for all individuals. Execution time 1000 ms per individual. Single robot.

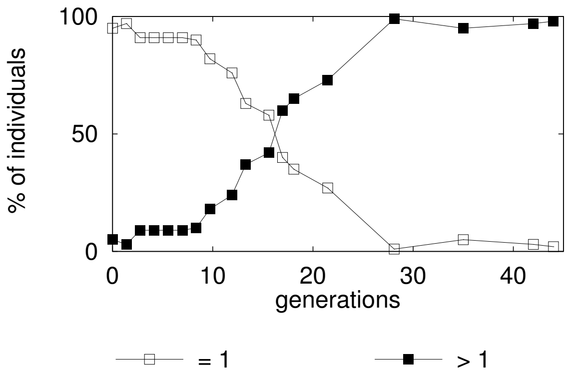

Figure 5: Experiment 2. Number of behaviours per individual. Execution time 1000 ms per individual. Single robot.

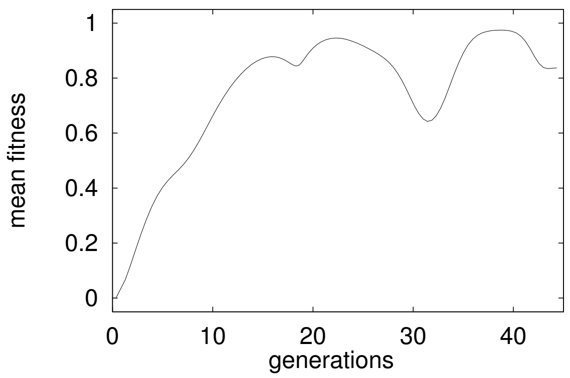

Figure 6: Experiment 2. Mean fitness. Execution time 1000 ms per individual. Single robot.

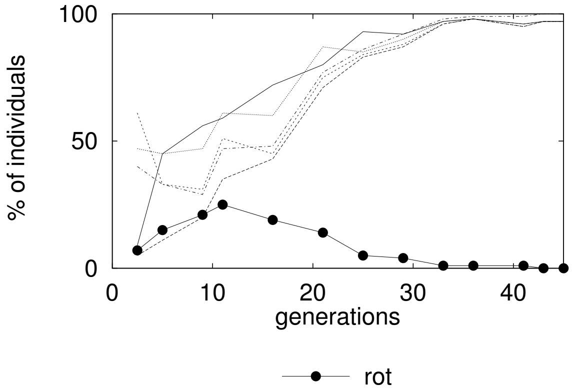

Figure 7: Experiment 3. Basic behaviours for all individuals. Execution time 1000 ms per individual. Four robots with migration rate 25% per generation.

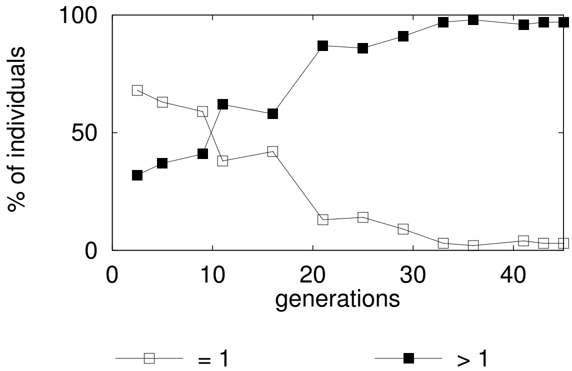

Figure 8: Experiment 3. Number of behaviours per individual. Execution time 1000 ms per individual. Four robots with migration rate 25% per generation.

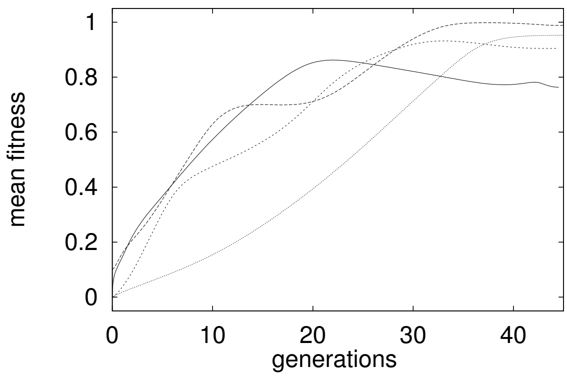

Figure 9: Experiment 3. Mean fitness. Execution time 1000 ms per individual. Four robots with migration rate 25% per generation.

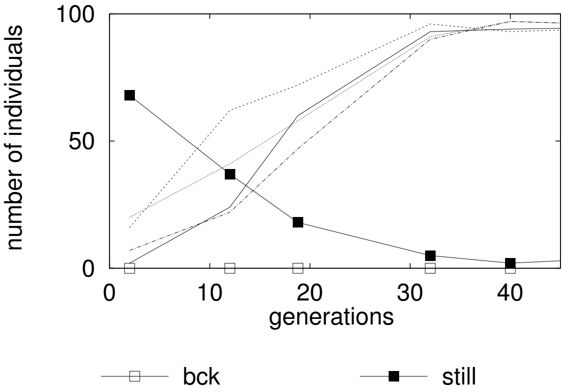

Figure 10: Experiment 4. Basic behaviours for all individuals. Execution time 250 ms per individual. Four robots with migration rate 5% per generation.

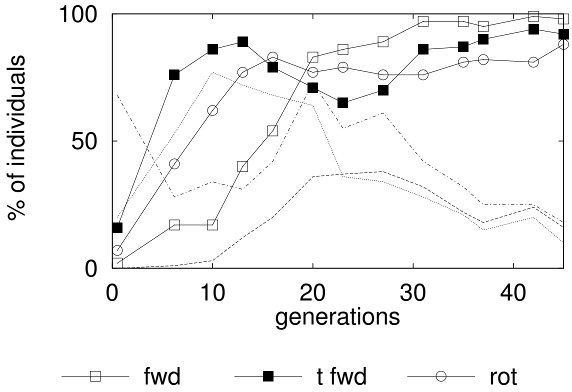

Figure 13: Experiment 5. Basic behaviours for all individuals. Execution time 1000 ms per individual. Four robots with migration rate 5% per generation.

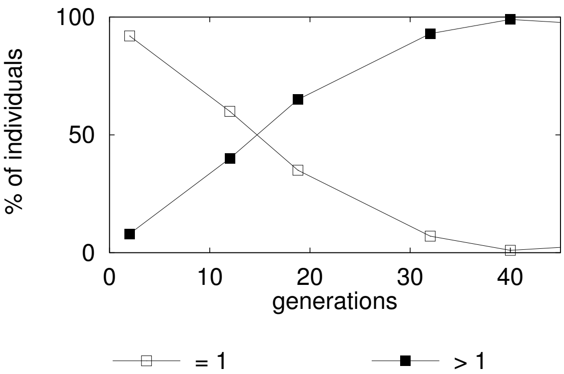

Figure 11: Experiment 4. Number of behaviours per individual. Execution time 250 ms per individual. Four robots with migration rate 5% per generation.

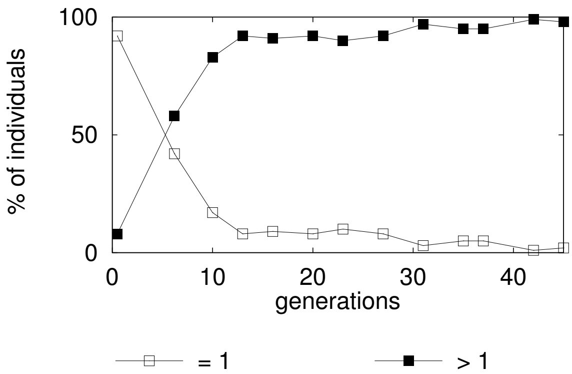

Figure 14: Experiment 5. Number of behaviours per individual. Execution time 1000 ms per individual. Four robots with migration rate 5% per generation.

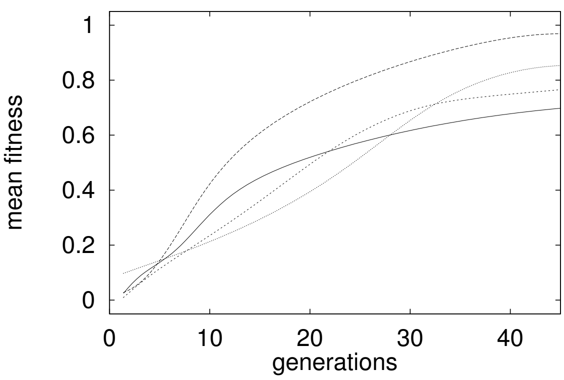

Figure 12: Experiment 4. Mean fitness. Execution time 250 ms per individual. Four robots with migration rate 5% per generation.

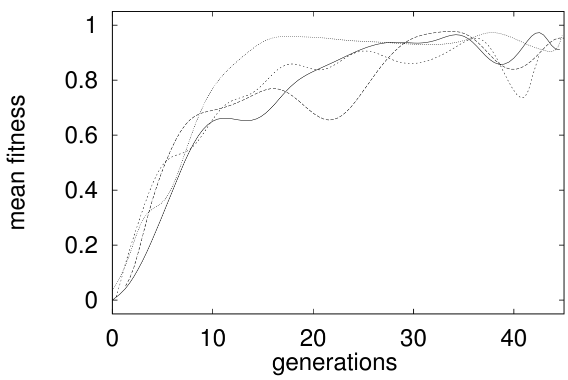

Figure 15: Experiment 5. Mean fitness. Execution time 1000 ms per individual. Four robots with migration rate 5% per generation.

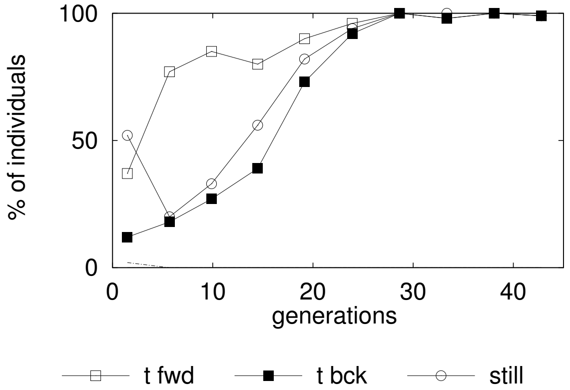

Figure 16: Experiment 6. Basic behaviours for all individuals. Execution time 250 ms per individual. Single robot.

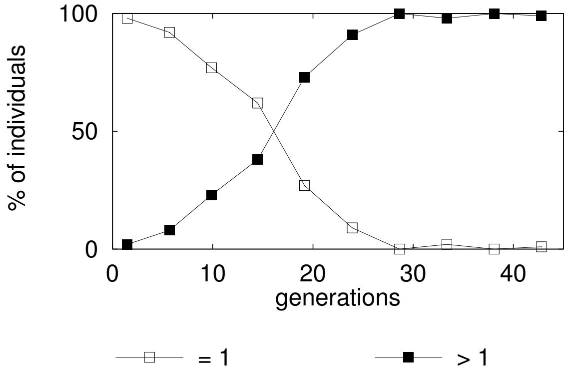

Figure 17: Experiment 6. Number of behaviours per individual. Execution time 250 ms per individual. Single robot.

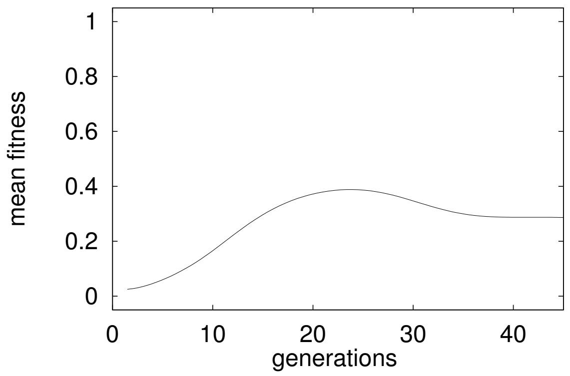

Figure 18: Experiment 6. Mean fitness. Execution time 250 ms per individual. Single robot.

References

[1] Fredrik Samuelsson. Distributed evolution of behaviour for a group of social autonomous agents. Master's thesis, Physical Resource Theory, Chalmers University of Technology, www.dtek.chalmers.se/˜d4sama, 2001.

[2] LegoMindstorms. *Lego Mindstorms.* www.legomindstorms.com, 2001.

[3] LegOS. *LegOS.* www.noga.de/legOS, 2001.

[4] Peter Nordin and Wolfgang Banzhaf. Genetic programming controlling a miniature robot. In E. V. Siegel and J. R. Koza, editors, *Working Notes for the AAAI Symposium on Genetic Programming*, pages 61–67, MIT, Cambridge, MA, USA, 10–12 November 1995. AAAI.

[5] Ronald C. Arkin. *Behaviour-Based Robotics.* 1998. ISBN: 0262011654.

[6] Banzhaf, Nordin, Keller, and Francone. *Genetic Programming an Introduction.* 1998. ISBN:1-55860-510-X.