Project no. 640891

DREAM

Deferred Restructuring of Experience in Autonomous Machines

Horizon 2020 Framework Programme

FETPROACT-2-2014

Knowing, doing, being: cognition beyond problem solving

Start date: 1 January 2015 – Duration: 48 months

D4.2

Publication

Due date: M24 - 12/2016

Actual submission date: M24 - 12/2016

Number of pages: 17

Lead beneficiary: VU/VUmc

Author(s): Heinerman, J., Zonta, A., Haasdijk, E. & Eiben, A.E.

Project Consortium

Beneficiary no.	Beneficiary name	Short name
1 (Coordinator)	UNIVERSITE PIERRE ET MARIE CURIE-PARIS	UPMC
2	UNIVERSIDADE DA CORUNA	UDC
3	QUEEN MARY UNIVERSITY OF LONDON	QML
4	ASSOCIATION POUR LA RECHERCHE ET LE DEVELOPPEMENT DES METHODES ET PROCESSUS INDUSTRIELS	ARMINES
5	STICHTING VU-VUMC	VU/VUmc

Dissemination Level

PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Dissemination Nature

R	Report	X
P	Prototype	
D	Demonstration	
O	Other	

On-line Evolution of Foraging Behaviour in a Population of Real Robots

Jacqueline Heinerman[1], Alessandro Zonta[2], Evert Haasdijk[1], and A.E. Eiben[1]

[1] Vrije Universiteit Amsterdam, The Netherlands
[2] University of Padua, Italy
j.v.heinerman@vu.nl, alessandro.zonta@outlook.com, e.haasdijk@vu.nl, a.e.eiben@vu.nl

Abstract. This paper describes a study in evolutionary robotics conducted completely in hardware without using simulations. The experiments employ on-line evolution, where robot controllers evolve on-the-fly in the robots' environment as the robots perform their tasks. The main issue we consider is the feasibility of tackling a non-trivial task in a realistic timeframe. In particular, we investigate whether a population of six robots can evolve foraging behaviour in one hour. The experiments demonstrate that this is possible and they also shed light on some of the important features of our evolutionary system. Further to the specific results we also advocate the system itself. It provides an example of a replicable and affordable experimental set-up for other researches to engage in research into on-line evolution in a population of real robots.

Keywords: Evolutionary Robotics, Neural Networks, Distributed On-Line Learning, Embodied Evolution, Foraging

1 Introduction

Evolutionary robotics is a research area that "applies the selection, variation, and heredity principles of natural evolution to the design of robots with embodied intelligence" [5]. In particular, evolutionary robotics aims to evolve the controllers, the morphologies, or both, for real and simulated autonomous robots [18]. Considering the complexity of interactions between environment, morphology and controller, employing evolution is a very promising approach to designing intelligent robots for a range of circumstances [1, 13]. However, forced by technical constraints the usual modus operandi in evolutionary robotics is quite limited: evolve robot controllers in simulation and transfer the outcome to real hardware afterwards. Thus, even though the final goal is to obtain physical robots with intelligent behaviour, the evolutionary process is usually digital, which leads to the notorious reality gap problem [9]. Furthermore, the evolutionary algorithm is usually (ab)used as an off-line optimizer. It is only employed during the design stage that ends with finding a good controller that is then deployed on the real robot and kept fixed during the operational stage.

This paper describes a study in evolutionary robotics conducted completely in hardware, without recourse to any simulations. The experiments use Thymio

II robots as hardware platform [14]. These robots – recently on the market– are quite small, cheap, and easily available. Therefore, setting up a physical robot population is much less demanding than a few years ago. The experiments consider on-line evolution, where a collective of robots evolve controllers in the robots' task environment as the robots perform their tasks. This results in a distributed on-line setup where robots learn individually and socially, meaning the exchange of individually evolved controllers with others. The experiments are motivated by a long-term vision of an ongoing evolutionary process that enables adaptation to the environment and the task. This research differs from most evolutionary robotics that employs evolution as an optimisation procedure to obtain well-performing controllers before deploying the robots.

There are earlier experiments in distributed on-line evolutionary robotics that take place exclusively in hardware, e.g., the work of Watson et al. and Simões et al, both in 1999 [19, 15]. Watson et al. coined the phrase "embodied evolution" for such systems. These experiments consider very straightforward robot tasks such as phototaxis and obstacle avoidance, and subsequent research into embodied evolution-like settings has not yet ventured beyond that.

The research in this paper serves as a further stepping stone towards on-line evolutionary adaptation in complex tasks and complex environments. An important goal of this paper is, therefore, to provide a replicable, accessible and affordable experimental set-up for further research into the on-line evolution of non-trivial behaviour in real robots. This paper investigates the feasibility of tackling more complex tasks in a realistic timeframe. In particular, it considers the on-line evolution of foraging behaviour in a population of real robots, with a fitness function that rewards appropriate behaviours for a number of subgoals. We try to answer the following research questions:

1. Can a group of robots evolve appropriate controllers for a non-trivial task in one hour?
2. How important is the feature that the robots share the controllers they evolve individually (social learning)?
3. How important is the most task-specific part of the compound objective function?

An evolutionary process implemented in real hardware has the obvious benefit of avoiding the reality gap, and when considering multiple robots with sophisticated sensors such as cameras simulations may actually run slower than real time, even for groups as small as six robots. An additional benefit is that experimenting with real robots encourages the researcher to review the robots' actual behaviour during the experiments rather than allowing only post-facto analysis of the metrics gathered by unattended simulation runs. Thus, experimenting with real robots enhances the understanding of robot behaviour.

2 Related Work

In principle, the scope of the related work includes all on-line distributed evolution in populations of physical robots as well as research where foraging be-

haviour evolves in simulation and is subsequently transferred to hardware. For the latter category, we restrict the scope of related work to research applying neuro-evolutionary methods to control differential drive robots.

There are few papers that consider on-line distributed evolution in populations of physical robots. Pioneering works in this field are those by Watson *et al.* and Simões *et al.*, both published in 1999. Watson *et al.* implemented "embodied evolution" in a population of six robots [19]. The robot controllers evolved to tackle a phototaxis task in an on-line fashion by broadcasting parts of their genome at a rate proportional to their task performance. Simões *et al.* evolved morphological features as well as the controllers in the same genome for collision-free behaviour [15]. These are the first examples of evolutionary algorithms distributed in a population of physical robots where the robots are not only evaluated, but are also performing real tasks. Since then, four more studies show on-line distributed evolution in populations of physical robots where the task to learn is either only obstacle avoidance ([17], [7]), obstacle avoidance, phototaxis and robot seeking [10] or survival without a specified fitness function [3]. These studies focus on showing that on-line evolution of the controllers for different tasks is possible and that communication between the robots is always beneficial. To our knowledge, the current study is the first to consider evolving foraging behaviour in hardware and in an on-line fashion.

Foraging is often considered as a task in off-line evolution. Here, we consider research where the controllers evolve in simulation and the best controllers are transferred onto robotic hardware where they remain fixed. We found no examples of off-line evolution of foraging behaviour where evaluations were performed on robotic hardware. The first example is [12], where controllers evolved in simulation to locate, recognise and grasp "garbage" objects and transport them outside the arena. The platform of choice was a Khepera robot with a gripper module. With a population size of 100 it took 1,000 generations to generate the desired, complex behaviour. One controller evaluation took 8 seconds in simulation (and 300 seconds in reality). The number of components in the fitness function (penalties and rewards for different observed behaviours) was found to have substantial impact. Transferring the best evolved controller onto hardware resulted in only a small performance decline.

In [8], the author incorporated dynamic rearrangement of neural networks with the use of neuromodulators to push a "peg" to a light source. Both peg and target were arranged in a straight line in front of the robot. A population of 100 controllers evolved for 200 generations, with controllers evaluated by simulating them for 500 time steps. When transferred to a Khepera robot, the best controller showed appropriate behaviour.

In [16], the task was limited to box pushing, and this is the only example where the required behaviour was simple enough that the neural networks did not require hidden nodes. The authors investigated the difference in performance with fitness functions that use internal/external and global/local information. They showed that a global external fitness function (diametrically opposed to the notion of distributed on-line evolution) performed best for their task. The best

controller was transferred onto a Khepera robot and showed adequate behaviour. The controller resulted from a (30+1) evolutionary strategy running for 250 generations; controllers were evaluated by simulating them for 100 time steps, starting from different positions in the arena.

The most recent work that we know of is [4]. Here, multi-objective evolutionary algorithms were applied for two conflicting tasks: protecting another robot by following it closely and collecting objects in the arena. Controllers evolved in an unknown population size over 100 generations, with each controller evaluated in simulation for 70 seconds. Even though the input sensor values for the simulated and real robot (CR robot) did not overlap perfectly, the robot showed the desired behaviour.

Thus, evolving foraging behaviours in simulation and subsequently transferring the results to the appropriate hardware platform has been shown to work in a number of settings. However, once the final controller is transferred, it does not adapt further and so cannot cope with unforeseen circumstances or changes in the environment. Our ambition is to develop foraging behaviour through on-line evolution, with some speed-up afforded by distributing the evolutionary process across a small number of robots. Given the population sizes and number of generations used in simulation-based research, this is not a trivial task to accomplish within a realistic timeframe (i.e., within the robot battery's operational period).

3 System Description

3.1 Robot

The Thymio II robot includes seven Infra-Red (IR) proximity sensors for obstacle detection; five are arranged along the front and two along the back of the robot. The sensors return values between 0 and circa 4,500, with high values corresponding to close obstacles. The robot has differential drive with the maximum wheel actuators set between -500 and 500. The operating speed is set to 30% of the maximum speed, which is between -150 and 150, to prevent overheating and to adapt to the camera's image processing speed. We extend the standard Thymio set-up with a more powerful logic board, a camera, wireless communication, and a high capacity battery. We use a Raspberry Pi 2 that connects to the Thymio's sensors and actuators an processes the data from the Raspberry Pi NoIR Camera. A WiFi dongle (Edimax 150Mbps Wireless

Fig. 1: Thymio II robot, developed by The École Polytechnique Fédérale de Lausanne (EPFL) and École Cantonal d'Arts de Lausanne (ÉCAL), with Raspberry Pi 2, Raspberry Pi NoIR camera, WiFi dongle, external battery and a LEGO gripper.

802.11b/g/n nano USB WiFi Adapter model EW-7811Un) attached to the Raspberry Pi enables inter-robot communication. Battery life is etended through a Verbatim Dual USB 12,000 mAh battery, allowing for a total experimental time of 10 hours. The extended Thymio II is shown in Fig. 1. The hand made LEGO gripper helps the robot maintain control of the pucks when manoeuvring.

3.2 Environment

Two robots operate together in a 1×1 meter arena with six pucks and one target area. This set-up is duplicated three times, facilitating a total population of six robots, where communication across arena instances is possible. A set-up with three arenas with two robots each (as opposed to having all six robots in a single arena) reduces the likelihood that the robots' grippers become entangled.

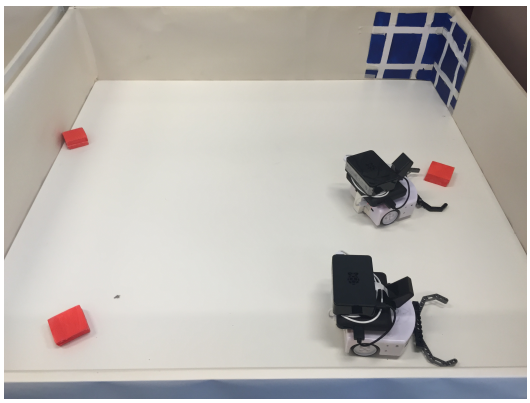


Fig. 2: The 1×1 meter arena with two Thymio II robots searching for the red pucks. The target location is indicated by the blue corner. The arena is duplicated three times to facilitate six robots while minimising robot collisions.

The arena size allows the robot to see the target area from across the whole arena. The arenas have a white floor and white walls for improved obstacle detection. The pucks are red and the target area, located in a corner, is blue, with white stripes added for better colour recognition when the robot is close to the target.

A WiFi router placed close to te arenas facilitates reliable wireless communication between all robots in the three arenas. A motion tracking system monitors the robots and pucks in one of the three arenas for post-hoc qualitative analysis of robot traces.

3.3 Task and Objective Function

In the foraging task, the robots must collect items (pucks) and deliver them to a target location. Although this task may not seem difficult, there is no consensus on how to define an objective function for this task, and in particular there is no prior experience with suitable functions for on-line evolution. Our experiments use an objective function that rewards appropriate behaviour for a number of subgoals to provide a relatively smooth fitness gradient; only rewarding the successful delivery of a puck results in a largely featureless fitness

landscape. The objective function assesses robot behaviour over a period of T timesteps as follows:

$$f_{total} = \sum_{t=0}^{T} f_{obs} + f_{puck} + f_{target} + f_{bonus}, \tag{1}$$

where:

$$f_{obs} = v_{trans} \cdot (1 - v_{sens}), \tag{2a}$$

$$f_{puck} = b_{puck} + 2 \cdot b_{push}, \tag{2b}$$

$$f_{target} = b_{tar}, \tag{2c}$$

$$f_{bonus} = b_{push} \times b_{tar}, \tag{2d}$$

and:

- v_{trans} is the translational speed (normalised between 0 and 1), calculated as the sum of the absolute speeds assigned to the left and right motor;
- v_{sens} is the value of the proximity sensor closest to an obstacle and normalised between 0 and 1;
- b_{puck} is a boolean value indicating whether the camera detects a puck in sight;
- b_{push} is a boolean value indicating whether the robot is pushing a puck;
- b_{tar} is a boolean value indicating whether the camera detects the target area.

Note, that the analyses in Sec. 5 report overall system performance as the number of pucks collected over a period of time, not as the fitness function as described above.

3.4 Controller

The controller is a feed forward neural network with 13 input nodes, 5 hidden nodes and 2 output nodes as depicted in Fig. 3. The nodes have *tanh* activation functions. Five input nodes, denoted PS_1 to PS_5, connect to the proximity sensors (three in the front and two in the back of the robot). The remaining seven inputs relate to camera information. To extract the salient information from the camera image, the image is divided into four parts (left, middle, right and bottom, or l, r, m and b) as shown in the top left in Fig. 3.

Masking red values to detect pucks in the four areas yields the (P_l, P_m, P_r, P_b) inputs that denote the size of the largest red area in each sub-image. A further three inputs denote the total percentage of blue in the three top sub-images (T_l, T_m, T_r). The input and hidden layer each have an additional bias node. The output nodes drive the left and right motor actuators. All input nodes are connected to all hidden nodes (except the bias node) and all hidden nodes are connected to the output nodes, resulting in a neural network with 62 weights. The sensor input values are normalised between -1 and 1 and the weights are between -4 and 4.

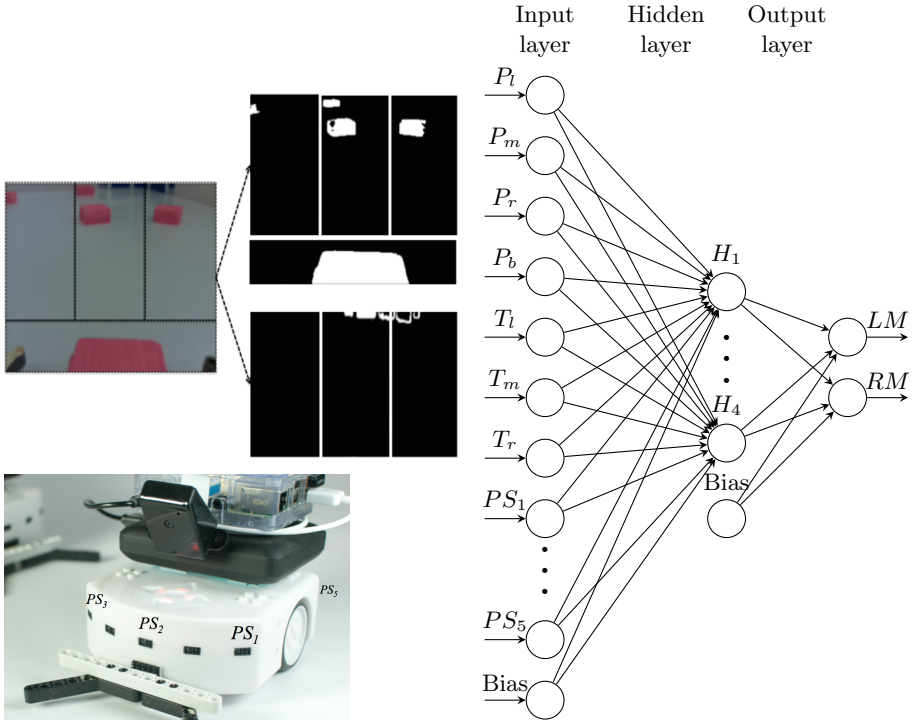


Fig. 3: Visualisation of salient information extracted from the camera image (left) and the neural network controller with 13 input nodes, 5 hidden nodes and 2 output nodes (right). Input nodes P_x detect pucks as the largest area of red pixels in each sub-image (l, m, r, b denoting the left, middle, right and bottom sun-image), T_x detect the total amount of blue pixels in each of the three upper sub-images and PS_x detect obstacles.

3.5 Evolutionary or Learning Mechanisms

The evolutionary process employs the common direct encoding scheme of an array of neural network weights. In this case, the length of that array is 62. Controllers evolve on-line and on-board; the robots encapsulate a self-contained evolutionary algorithm and augment this with a distributed evolutionary system, amounting to what was labelled as a hybrid scheme in [6]. Such a combination of evolutionary processes can also be cast as a combination of individual learning (the encapsulated evolutionary process) and social learning (the distributed evolutionary process). We adopt the individual and social learning terminology to align with the DREAM project[1] of which this research is part.

[1] `http://robotsthatdream.eu`

Individual Learning The robots implement individual learning through a $(1+1)$ evolutionary strategy, similar to the approach in [7] and [2], using the objective function defined in Eq. 1. The neural network weights are mutated using Gaussian perturbation with $N(0, \sigma)$, where the value of σ doubles when the offspring controller (the challenger) does not improve on the current controller (the champion). When a challenger outperforms the current champion, it replaces the champion. Each controller is evaluated over a testing epoch of 1,000 time steps, equating to a period of circa 3.6 seconds.

This is a very short evaluation time, certainly too short to complete one round of detecting and approaching a puck and then transporting it to the target area. One could argue that this evaluation time should be longer, but the difficulty of a longer evaluation time is the need for more controller evaluations because it will take the robot longer to discover all the subtasks required to perform the overall task. The short evaluation time results in a quick response to objects of interest — the pucks and target. To enable robust assessment of controllers, even though the robot cannot experience all relevant situations during a single evaluation, the robots re-evaluate their champion controllers with a 20% chance. The champion's fitness value is updated upon re-evaluation by taking the weighted sum of the current and the re-evaluated fitness with a 20-80 weight distribution (20% of the re-evaluated and 80% of the current fitness value).

Social Learning The robots broadcast their champion genome after every evaluation, provided that its fitness exceeds 50% of the maximum fitness. Robots cache received genomes in their *social learning storage*. When a robot embarks on a round of social learning, it takes the most recently received genome from the storage and either evaluates it directly as a challenger (in 75% of the cases) or creates a challenger through averaging crossover with the current champion. Directly re-evaluating received controllers rather than copying the sender's fitness value makes sense because the robot may be in a very different situation than the sending robot when it evaluated the controller. Just as with individual learning, the challenger replaces the champion if it outperforms it.

The robots randomly alternate between individual and social learning with a rate of 70:30.

4 Experimental Set-up

The duration of each experiment is one hour, allowing for 1,000 controller evaluations of 1,000 time steps (ca 3.6 seconds) each. Unless indicated otherwise, each set-up was repeated 20 times. The table on the right lists the most important system parameters[2].

System parameters	
Total controller evaluations	1000
Evaluation duration (sec)	3.6
Re-evaluation chance	20%
Re-evaluation champion weight	80%
Individual learning chance	70%
Social learning chance	30%
Individual Learning	
Maximum fitness	6000
Weight range	8
σ initial	1
σ maximum	4
σ minimum	0.01
Social Learning	
Broadcast threshold	50%
Averaging crossover	25%
Import	75%
Social learning storage size	20

The robots start every experiment in the same position, facing the middle of one of the walls that is not adjacent to the target area. The robots are not repositioned during an experiment unless two robots' grippers or wiring become entangled or when a robot is pushing multiple pucks against the wall (in that case, the pucks prevent the robot from moving close enough to the wall for obstacle sensors to detect it). In case the robot loses its gripper, it is reattached and the experiment continues. Furthermore, the temperature of the robots is constantly tracked to prevent the robots from overheating. As a result, no robots broke during the experiments and all collected data can be used.

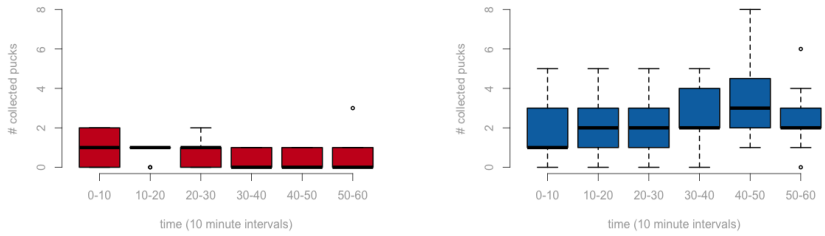
The robots are preprogrammed with a threshold for the amount of blue that needs to be exceeded before we can say that a puck is collected. When a puck is pushed to the target area, the robot internally registers the time and emits a sound to alert the experimenter. The puck is then manually replaced in the centre of the arena and the experimenter records the number of collected pucks.

5 Experimental Results

To quantify the performance of the robots, we consider the number of pucks collected in ten-minute intervals. The objective function defined in Eq. 1 is not suitable for this purpose: controller evaluations are so short that a perfectly good controller scores poorly, e.g., because the robot spends the entire evaluation period looking for pucks.

Figure 4 compares a baseline experiment with the performance when learning is enabled. For the baseline experiment, the evolutionary mechanisms are disabled and the robots run with randomly generated weights for every evaluation.

[2] The code for implementation is available on `https://github.com/ci-group/Thymio_swarm`.



(a) Random controller

(b) Evolving controller

Fig. 4: Number of collected pucks per 10 minute intervals by the robot population for random and evolving controllers. The box plots show the median, interquartile range, min and max values. The evolutionary process was repeated 20 times, the random baseline was repeated 10 times. Although the increase in pucks collected when learning increases only marginally, the increase is significant (Mood's Median test with Fisher's Exact Test, $p = 0.02$).

There is a clear difference in performance between the random and evolving controllers, so learning definitely improves task performance. If the robots do learn the foraging task successfully, the number of pucks collected per interval should increase over the course of an experiment. The performance does indeed increase over time in Fig. 4(b), but the increase is slight. To ascertain whether the increase in performance is statistically significant, we use Mood's Median test on the median number of pucks collected the first and the second half hour.

Mood's Median test is calculated as follows: the median number of pucks collected in the first and second half of each experiment is calculated. The data for the 20 repeats is then summarised in a 2×2 table: each cell in the table contains the count of repeats in its class. Classes are assigned to the cells according to whether the performance metric was calculated over the first or the second half hour (first or second column, respectively) and whether the metric is lower or equal or higher than the overall median (first and second row, respectively). The standard Mood's Median test then calls for a χ^2 test, but in this comparison requires a one-tailed test to establish whether the robots collect significantly more pucks in the seconds half hour. Therefore, we substitute it with Fisher's Exact test. For these data, this yielded $p = 0.02$ – a significant increase at $\alpha = 5\%$.

Performance is in fact somewhat disappointing and also worse than at least some behaviour that was observed (one of the benefits of running experiments in real robots is that the experimenters monitor robot behaviour as a matter of course and do not rely only on quantitative post-hoc analysis as is often the case with simulation-based experiments). From observations of the robots during the experiment, it appeared that the robots do learn appropriate behaviour, but subsequently revert to much less efficient behaviour.

To investigate whether the robots do indeed learn (and subsequently forget) efficient foraging behaviour, we ran ten repeats of an experiment with a high-fitness controller without further development. This controller was selected manually from controllers that consistently showed high fitness values, also over multiple re-evaluations. All six robots were programmed with the neural network weights set according to this individual genome, positioned in the standard starting position and subsequently ran for one hour, with all other settings as described for the runs with the learning mechanisms enabled.

The box plots in Fig. 5 show the performance of ten repeats of this experiment in ten-minute intervals. The first ten minutes show exceptional performance due to the convenient starting position of the pucks in the middle of the arena. But also in the other intervals, performance beats the on-line performance with an active learning process by a factor of three (note, that the y-axis in Fig.5 had a larger range than that in Fig 4).

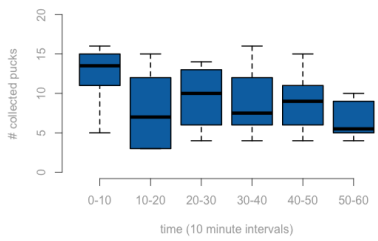


Fig. 5: Number of collected pucks over ten runs per 10 minute interval of one of the better controllers developed during an evolutionary run. The box plot shows the median, interquartile range, min and max values.

Thus, it appears that the evolutionary process does discover very good controllers, but subsequently discards these individuals because they perform poorly at some point. We hypothesise that this is a result of the brief evaluation periods that these experiments require: if, for instance, no pucks are picked up during a re-evaluation of a controller, its assessment is revised, even if it is actually very relevant. Combating this 'forgetting' behaviour is the focus of further research that is now underway.

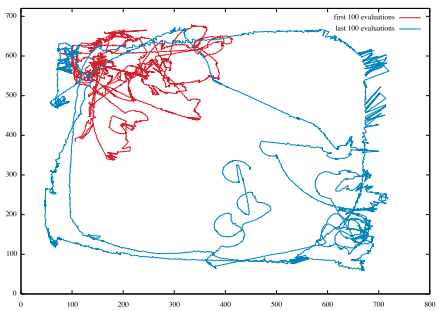


Fig. 6: Traces of one robot showing the first (red) and last (blue) 100 controller evaluations. The target location is at position (800,0). The traces show the mid-point of each robot, meaning that the robots learn wall-following behaviours.

For a qualitative analysis of robot behaviour and the effect of evolution, Fig. 6 shows traces of one robots at the beginning and towards the end of an experiment (the first and last 100 evaluations in red and blue, respectively). The traces indicate typical behaviour that was observed in most runs. The robots learn wall-following behaviour – the traces show the mid-point of each robot, its sides are actually very close to the wall for large parts of the trace.

Because the robots often push pucks towards the wall in the early stages of learning, this is quite efficient behaviour to move pucks to the target area. When the robots deviate from this behaviour, they often end up pushing more pucks to the wall to resume wall-following.

The Need for Social Learning. To investigate the influence of the social learning mechanisms, another set of 20 experiments was conducted where communication between robots was disabled, so that they only learn using the encapsulated $1 + 1$ evolution strategy. Other than that, all settings are as in the original runs presented above. Fig. 7 compares the results from the experiments with both social and individual learning with experiments where social learning is disabled.

The plot shows that the social learning mechanisms yields better performance and is in fact necessary to prevent a decline in performance. Social learning implements a shared repository of controllers that goes some way to exacerbate the 'forgetting' behaviour.

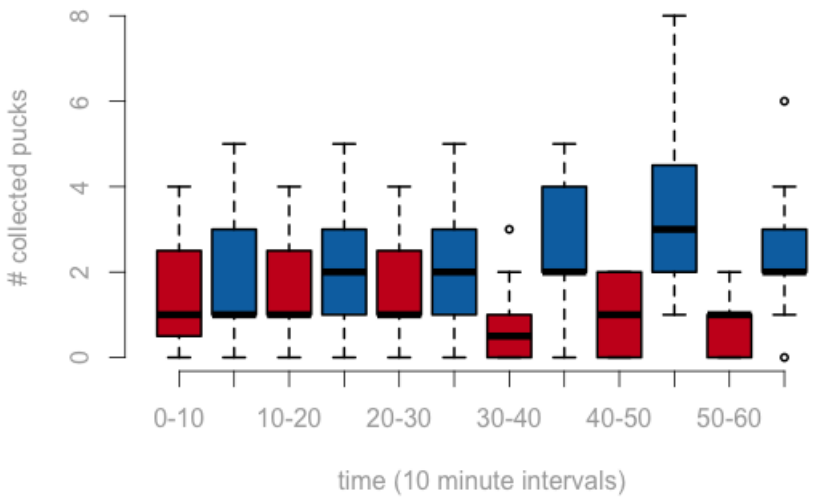


Fig. 7: Number of collected pucks per 10 minute interval by the robot population for evolving controllers without (red) and with communication (blue). Box plots show median, interquartile range, min and max values. Both experiments were repeated 20 times. Communication is shown to be necessary to ensure a performance increase for the foraging task.

Objective Function Design. This is known to be a difficult problem in general [11]. In our case, the observation of robot trajectories (cf. Fig. 6) suggests that evolution solves the task differently than we, the experimenters, have expected. In particular it seems that the behaviours are optimsed for grabbing a puck and carrying it around along the walls.

This raises the question whether rewards f_{target} and f_{bonus} that correspond to the subtask of reaching the target area are important or not. To find out we ran a series of experiments with a reduced objective function that contains only the first two terms f_{obs} and f_{puck} of formula 1. Based on this function the optimal behavior is continuously pushing a puck without hitting the walls.

The results of these experiments are shown in red in Fig. 8, with the original results in blue for reference. Using these data we can ask two questions regarding the role of the target related rewards: 1) Whether the robots are still learning the puck collecting task without these rewards? and 2) Whether the omission of these rewards decreases performance?

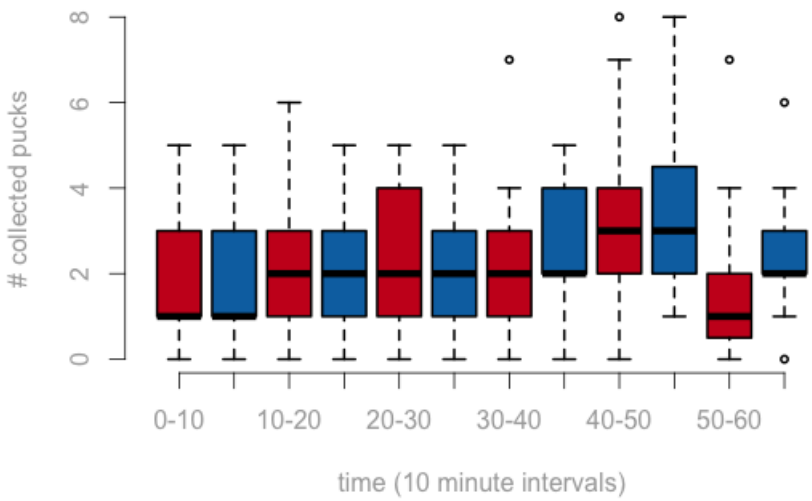


Fig. 8: Number of collected pucks per 10 minute interval by the robot population for evolving controllers without (red) and with the bonus component of the fitness function (blue). Box plots show median, interquartile range, min and max values. Both experiments were repeated 20 times. Although there seems to be an increase in pucks collected when there is no bonus component, the increase is not significant (Mood's Median test with Fisher's Exact Test, $p = 0.57$).

Calculating the Mood's Median test for the increase in performance for first and second half an hour, gives a value of $p = 0.57$. In other words, the performance in the second half of the experiment is not significantly higher than in the first half. Thus, with the reduced fitness function the population does not seem to learn over time.

To answer the second question, Mood's Median tests are performed for every ten minutes between the two algorithms. Values of $p = 0.62, 0.63, 0.9, 0.5, 0.37, 0.09$ indicate that there is no significant decrease in performance when removing the rewards related to the target area – using a significance level $\alpha = 5\%$.

These results are seemingly contradictory. On the one hand, the system with the full fitness function (blue) is different from the one with the reduced fitness function (red), because it does learn over time, while the other one does not. On the other hand, the system with the full fitness function is not different from the one with the reduced fitness function, because their performance differences over the whole experiment are not significant. This "contradiction" is caused by the statistics and indeed the last 10 minutes when the red system exhibits poor performance.

A more interesting conclusion we can draw from these data concerns the role of the target related fitness rewards. The fact that red and blue do not differ significantly regarding the number of pucks collected can be explained by taking a closer look at the environment. The target area is positioned in a corner and can be reached by wall following. This feature implies that a robot that is continuously pushing around a puck without hitting the walls will eventually deliver the puck to the target area. This solution was not intended by us, but it nicely illustrates how evolution can exploit environmental features to the surprise of the experimenters.

6 Conclusions and Future Work

In this paper we described an evolutionary robotics study completely conducted in hardware, without using software simulations up front. An important result is the experimental evidence that a population of six robots can evolve foraging behaviour in one hour. The significance of this finding is apparent from a comparison with existing work; on-line evolution embodied in real robots has previously been only applied to relatively simple tasks, such as obstacle avoidance and phototaxis.

Our results show that the robots do learn foraging behaviour and that performance rises steadily if marginally. The robots learn appropriate control as shown by an analysis of one of the better controllers considered, but these controllers are lost, presumably when the robots re-evaluate them in inauspicious circumstances. This is probably linked to the short evaluation times necessary for our experiments. Further research is required to investigate possibilities to mitigate this 'forgetting' of good controllers.

Experiments disabling certain components of the evolutionary system revealed the importance of these components. In particular, we observed that al-

lowing the robots to communicate and cross-fertilise brings significant improvements with respect to a system where each robot is running an isolated internal evolutionary learning process whose results are not shared with the others. The robots' performance increase was found to be only significant when the robots are able to share information about their controller. Without the sharing of information, performance declines, probably because of the short evaluation time where the robot is not able to experience all subtasks in the arena within an evaluation. Thus, individual learning augmented with social learning outperforms individual learning alone. In the terminology of [2] this shows the advantage of a hybrid (encapsulated plus distributed) system over encapsulated evolution.

We also looked into the composition of the fitness function and learned that rewarding the subtask of reaching the target area is not necessary to obtain overall good performance. This is a consequence of a particular feature of the environment that can be exploited by evolution.

The main ambition of the research in this paper was to push for more complex tasks to be considered in on-line evolutionary robotics outside of simulations. Although the performance achieved by these experiments can be improved upon, this paper provides an important stepping stone towards this goal by providing the research community with a well-documented and affordable example of an experimental set-up to investigate an evolving collective of robots with rich sensory inputs in a non-trivial task. We hope that this will provide inspiration and opportunity for other researchers to research physical, not simulated, robot collectives that evolve to tackle tasks beyond the complexity of obstacle avoidance.

7 Acknowledgements

This work was made possible by the European Union FET Proactive Initiative *Knowing, Doing, Being: Cognition Beyond Problem Solving,* funding the Deferred Restructuring of Experience in Autonomous Machines (DREAM) project under grant agreement 640891.

References

1. Bongard, J.: Evolutionary robotics. Communications of the ACM 56(8), 74–85 (2013)
2. Bredeche, N., Haasdijk, E., Eiben, A.E.: On-line, on-board evolution of robot controllers. In: Collet, P., Monmarché, N., Legrand, P., Schoenauer, M., Lutton, E. (eds.) Proceedings of the 9th international conference on Artificial Evolution. pp. 110–121. Springer, Berlin (2009)
3. Bredeche, N., Montanier, J.M., Liu, W., Winfield, A.F.: Environment-driven distributed evolutionary adaptation in a population of autonomous robotic agents. Mathematical and Computer Modelling of Dynamical Systems 18(1), 101–129 (2012)
4. Capi, G.: Multiobjective evolution of neural controllers and task complexity. Robotics, IEEE Transactions on 23(6), 1225–1234 (2007)

5. Doncieux, S., Bredeche, N., Mouret, J.B., Eiben, A.: Evolutionary robotics: what, why, and where to. Frontiers in Robotics and AI 2(4) (2015)
6. Eiben, A.E., Haasdijk, E., Bredeche, N.: Embodied, on-line, on-board evolution for autonomous robotics. In: Levi, P., Kernbach, S. (eds.) Symbiotic Multi-Robot Organisms: Reliability, Adaptability, Evolution, chap. 5.2, pp. 361–382. Springer (May 2010), http://www.springer.com/engineering/mathematical/book/978-3-642-11691-9
7. Heinerman, J., Rango, M., Eiben, A.: Evolution, individual learning, and social learning in a swarm of real robots. In: Proceedings of the 2015 IEEE International Conference on Evolvable Systems (ICES). vol. (to appear). IEEE (2015)
8. Ishiguro, A., Tokura, S., Kondo, T., Uchikawa, Y., Eggenberger, P.: Reduction of the gap between simulated and real environments in evolutionary robotics: a dynamically-rearranging neural network approach. In: Systems, Man, and Cybernetics, 1999. IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on. vol. 3, pp. 239–244. IEEE (1999)
9. Jakobi, N., Husbands, P., Harvey, I.: Noise and the reality gap: The use of simulation in evolutionary robotics. In: Moran, F., Moreno, A., Merelo, J., Chacon, P. (eds.) Advances in Artificial Life, pp. 704–720. No. 929 in LNAI, Springer (1995)
10. Nehmzow, U.: Physically embedded genetic algorithm learning in multi-robot scenarios: The pega algorithm. In: Prince, C., Demiris, Y., Marom, Y., Kozima, H., Balkenius, C. (eds.) Proceedings of The Second International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems. No. 94 in Lund University Cognitive Studies, LUCS, Edinburgh, UK (August 2002)
11. Nelson, A.L., Barlow, G.J., Doitsidis, L.: Fitness functions in evolutionary robotics: A survey and analysis. Robotics and Autonomous Systems 57(4), 345 – 370 (2009), http://www.sciencedirect.com/science/article/B6V16-4TTMJV3-1/2/2549524d8e0f3982730659e49ad3fa75
12. Nolfi, S.: Evolving non-trivial behaviors on real robots: A garbage collecting robot. Robotics and Autonomous Systems 22(3), 187–198 (1997)
13. Nolfi, S., Floreano, D.: Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines. MIT Press, Cambridge, MA (2000)
14. Riedo, F., Chevalier, M.S.D., Magnenat, S., Mondada, F.: Thymio II, a robot that grows wiser with children. In: Advanced Robotics and its Social Impacts (ARSO), 2013 IEEE Workshop on. pp. 187 – 193. IEEE (2013)
15. Simoes, E., Dimond, K.R.: An evolutionary controller for autonomous multi-robot systems. In: Systems, Man, and Cybernetics, 1999. IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on. vol. 6, pp. 596–601. IEEE (1999)
16. Sprinkhuizen-Kuyper, I.G., Kortmann, R., Postma, E.O.: Fitness functions for evolving box-pushing behaviour. In: Proceedings of the Twelfth Belgium-Netherlands Artificial Intelligence Conference. pp. 275–282 (2000)
17. Usui, Y., Arita, T.: Situated and embodied evolution in collective evolutionary robotics. In: Proceedings of the 8th International Symposium on Artificial Life and Robotics. pp. 212–215 (2003)
18. Vargas, P., Paolo, E.D., Harvey, I., Husbands, P. (eds.): The Horizons of Evolutionary Robotics. MIT Press (2014)
19. Watson, R., Ficiei, S., Pollack, J.B., et al.: Embodied evolution: Embodying an evolutionary algorithm in a population of robots. In: Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on. vol. 1, pp. 335–342. IEEE (1999)