A Case Study on the Scalability of Online Evolution of Robotic Controllers

Fernando Silva[1,2,4](✉), Luís Correia[4], and Anders Lyhne Christensen[1,2,3]

[1] BioMachines Lab, Lisboa, Portugal
fsilva@di.fc.ul.pt, anders.christensen@iscte.pt
[2] Instituto de Telecomunicações, Lisboa, Portugal
[3] Instituto Universitário de Lisboa (ISCTE-IUL), Lisboa, Portugal
[4] BioISI, Faculdade de Ciências, Universidade de Lisboa, Lisboa, Portugal
luis.correia@ciencias.ulisboa.pt

Abstract. Online evolution of controllers on real robots typically requires a prohibitively long evolution time. One potential solution is to distribute the evolutionary algorithm across a group of robots and evolve controllers in parallel. No systematic study on the scalability properties and dynamics of such algorithms with respect to the group size has, however, been conducted to date. In this paper, we present a case study on the scalability of online evolution. The algorithm used is odNEAT, which evolves artificial neural network controllers. We assess the scalability properties of odNEAT in four tasks with varying numbers of simulated e-puck-like robots. We show how online evolution algorithms can enable groups of different size to leverage their multiplicity, and how larger groups can: (i) achieve superior task performance, and (ii) enable a significant reduction in the evolution time and in the number of evaluations required to evolve controllers that solve the task.

Keywords: Evolutionary robotics · Artificial neural network · Evolutionary algorithm · Online evolution · Robot control · Scalability

1 Introduction

Evolutionary computation has been widely studied and applied to synthesise controllers for autonomous robots in the field of evolutionary robotics (ER). In *online* ER approaches, an evolutionary algorithm (EA) is executed onboard robots during task execution to continuously optimise behavioural control. The main components of the EA (evaluation, selection, and reproduction) are performed by the robots without any external supervision. Online evolution thus enables addressing tasks that require online learning or online adaptation. For instance, robots can evolve new controllers and modify their behaviour to respond to unforeseen circumstances, such as changes in the task or in the environment.

F. Pereira et al. (Eds.) EPIA 2015, LNAI 9273, pp. 189–200, 2015.
DOI: 10.1007/978-3-319-23485-4_20

Research in online evolution started out with a study by Floreano and Mondada [1], who conducted experiments on a real mobile robot. The authors successfully evolved navigation and obstacle avoidance behaviours for a Khepera robot. The study was a significant breakthrough as it demonstrated the potential of online evolution of controllers. Researchers then focused on how to mitigate the issues posed by evolving controllers directly on real robots, especially the prohibitively long time required [2]. Watson *et al.* [3] introduced an approach called *embodied evolution* in which an online EA is distributed across a group of robots. The main motivation behind the use of multirobot systems was to leverage the potential speed-up of evolution due to robots that evolve controllers in parallel and that exchange candidate solutions to the task.

Over the past decade, numerous approaches to online evolution in multirobot systems have been developed. Examples include Bianco and Nolfi's open-ended approach for self-assembling robots [4], mEDEA by Bredeche *et al.* [5], and odNEAT by Silva *et al.* [6]. When the online EA is decentralised and distributed across a group of robots, one common assumption is that online evolution inherently scales with the number of robots [3]. Generally, the idea is that the more robots are available, the more evaluations can be performed in parallel, and the the faster the evolutionary process [3]. The dynamics of the online EA itself, and common issues that arise in EAs from population sizing such as convergence rates and diversity [7] have, however, not been considered. Furthermore, besides ad-hoc experiments with large groups of robots, see [5] for examples, there has been no systematic study on the scalability properties of online EAs across different tasks. Given the strikingly long time that online evolution requires to synthesise solutions to any but the simplest of tasks, the approach remains infeasible on real robots [8].

In this paper, we study the scalability properties of online evolution of robotic controllers. The online EA used in this case study is odNEAT [9], which optimises artificial neural network (ANN) controllers. One of the main advantages of odNEAT is that it evolves both the weights and the topology of ANNs, thereby bypassing the inherent limitations of fixed-topology algorithms [9]. odNEAT is used here as a representative efficient algorithm that has been successfully used in a number of simulation-based studies related to adaptation and learning in robot systems, see [6,8–11] for examples. We assess the scalability properties and performance of odNEAT in four tasks involving groups of up to 25 simulated e-puck-like robots [12]: (i) an aggregation task, (ii) a dynamic phototaxis task, and (iii, iv) two foraging tasks with differing complexity. Overall, our study shows how online EAs can enable groups of different size to leverage their multiplicity for higher performance, and for faster evolution in terms of evolution time and number of evaluations required to evolve effective controllers.

2 Online Evolution with odNEAT

This section provides an overview of odNEAT; for a comprehensive introduction see [9]. odNEAT is an efficient online neuroevolution algorithm designed for multirobot systems. The algorithm starts with minimal networks with no hidden neurons, and with each input neuron connected to every output neuron. Throughout evolution, topologies are gradually complexified by adding new neurons and new connections through mutation. In this way, odNEAT is able find an appropriate degree of complexity for the current task, and a suitable ANN topology is the result of a continuous evolutionary process [9].

odNEAT is distributed across multiple robots that exchange candidate solutions to the task. The online evolutionary process is implemented according to a physically distributed island model. Each robot optimises an internal population of genomes (directly encoded ANNs) through intra-island variation, and genetic information between two or more robots is exchanged through inter-island migration. In this way, each robot is potentially self-sufficient and the evolutionary process opportunistically capitalises on the exchange of genetic information between multiple robots for collective problem solving [9].

During task execution, each robot is controlled by an ANN that represents a candidate solution to a given task. Controllers maintain a virtual energy level reflecting their individual performance. The fitness value is defined as the mean energy level. When the virtual energy level of a robot reaches a minimum threshold, the current controller is considered unfit for the task. A new controller is then created via selection of two parents from the internal population, crossover of the parents' genomes, and mutation of the offspring. Mutation is both structural and parametric, as it adds new neurons and new connections, and optimises parameters such as connection weights and neuron bias values.

odNEAT has been successfully used in a number of simulation-based studies related to long-term self-adaptation in robot systems. Previous studies have shown: (i) that odNEAT effectively evolves controllers for robots that operate in dynamic environments with changing task parameters [11], (ii) that the controllers evolved are robust and can often adapt to changes in environmental conditions without further evolution [9], (iii) that robots executing odNEAT can display a high degree of fault tolerance as they are able to adapt and learn new behaviours in the presence of faults in the sensors [9], (iv) how to extend the algorithm to incorporate learning processes [11], and (v) how to evolve behavioural building blocks prespecified by the human experimenter [8,10]. Given previous results, odNEAT is therefore used in our study as a representative online EA. The key research question of our study is if and how online EAs can enable robots to leverage their multiplicity. That is, besides performance and robustness criteria, we are interested in studying scalability with respect to the group size, an important aspect when large groups of robots are considered.

3 Methods

In this section, we define our experimental methodology, including the simulation platform and robot model, and we describe the four tasks used in the study: aggregation, phototaxis, and two foraging tasks with differing complexity.

3.1 Experimental Setup

We use JBotEvolver [13] to conduct our simulation-based experiments. JBotEvolver is an open-source, multirobot simulation platform and neuroevolution framework. In our experiments, the simulated robots are modelled after the e-puck [12], a 7.5 cm in diameter, differential drive robot capable of moving at speeds up to 13 cm/s. Each robot is equipped with infrared sensors that multiplex obstacle sensing and communication between robots at a range of up to 25 cm.[1] The sensor and actuator configurations for the tasks are listed in Table 1. Each sensor and each actuator are subject to noise, which is simulated by adding a random Gaussian component within ± 5% of the sensor saturation value or of the current actuation value.

The robot controllers are discrete-time ANNs with connection weights in the range [-10,10]. odNEAT starts with simple networks with no hidden neurons, and with each input neuron connected to every output neuron. The ANN inputs are the readings from the sensors, normalised to the interval [0,1]. The output layer has two neurons whose values are linearly scaled from [0,1] to [-1,1] to set the signed speed of each wheel. In the two foraging tasks, a third output neuron sets the state of a gripper. The gripper is activated if the output value of the neuron is higher than 0.5, otherwise it is deactivated. If the gripper is activated, the robot collects the closest resource within a range of 2 cm, if there is any. Depending on the foraging task (see below), the robot may need to actively select which type of resources to collect and to avoid other types.

Aggregation Task. In an aggregation task, dispersed robots must move close to one another to form a single cluster. Aggregation combines several aspects of multirobot tasks, including distributed individual search, coordinated movement, and cooperation. Furthermore, aggregation plays an important role in robotics because it is a precursor of other collective behaviours such as group transport of heavy objects [14]. In our aggregation task, robots are evaluated based on criteria that include the presence of robots nearby, and the ability to explore the arena and move fast, see [9] for details. The initial virtual energy level E of each controller is set to 1000 and limited to the range $[0, 2000]$ units. At each control cycle, the update of the virtual energy level, E, is given by:

$$\frac{\Delta E}{\Delta t} = \alpha(t) + \gamma(t) \tag{1}$$

where t is the current control cycle, $\alpha(t)$ is a reward proportional to the number n of different genomes received in the last $P = \mathit{10}$ control cycles. Because robots executing odNEAT exchange candidate solutions, the number of different

[1] The original e-puck infrared range is 2-3 cm [12]. In real e-pucks, the *liblrcom* library, see http://www.e-puck.org, extends the range up to 25 cm and multiplexes infrared communication with proximity sensing.

Table 1. Controller details. Light sensors have a range of 50 cm (phototaxis task). Other sensors have a range of 25 cm.

Aggregation task – controller details	
Input neurons: 18	
	8 for IR robot detection
	8 for IR wall detection
	1 for energy level reading
	1 for reading the number of different genomes received
Output neurons: 2	Left and right motor speeds
Phototaxis task – controller details	
Input neurons: 25	
	8 for IR robot detection
	8 for IR wall detection
	8 for light source detection
	1 for energy level reading
Output neurons: 2	Left and right motor speeds
Foraging tasks – controller details	
Input neurons: 25	
	4 for IR robot detection
	4 for IR wall detection
	1 for energy level reading
	8 for resource A detection
	8 for resource B detection
Output neurons: 3	
	2 for left and right motor speeds
	1 for controlling the gripper

genomes received is used to estimate the number of robots nearby. $\gamma(t)$ is a factor related to the quality of movement computed as:

$$\gamma(t) = \begin{cases} \text{-1} & \text{if } v_l(t) \cdot v_r(t) < 0 \\ \Omega_s(t) \cdot \omega_s(t) & \text{otherwise} \end{cases} \tag{2}$$

where $v_l(t)$ and $v_r(t)$ are the left and right wheel speeds, $\Omega_s(t)$ is the ratio between the average and maximum speed, and $\omega_s(t) = \sqrt{v_l(t) \cdot v_r(t)}$ rewards controllers that move fast and straight at each control cycle.

Phototaxis Task. In a phototaxis task, robots have to search and move towards a light source. Following [9], we use a dynamic version of the phototaxis task in which the light source is periodically moved to a new random location. As a result, robots have to continuously search for and reach the light source, which eliminates controllers that find the light source by chance. The virtual energy level $E \in [0, 100]$ units, and controllers are assigned an initial value of 50 units. At each control cycle, E is updated as follows:

$$\frac{\Delta E}{\Delta t} = \begin{cases} S_r & \text{if } S_r > 0.5 \\ 0 & \text{if } 0 < S_r \leq 0.5 \\ \text{-0.01} & \text{if } S_r = 0 \end{cases} \tag{3}$$

where S_r is the maximum value of the readings from light sensors, between 0 (no light) and 1 (brightest light). Light sensors have a range of 50 cm and robots are therefore only rewarded if they are close to the light source. Remaining sensors have a range of 25 cm.

Foraging Tasks. In a foraging task, robots have to search for and pick up objects scattered in the environment. Foraging is a canonical testbed in cooperative robotics domains, and is evocative of tasks such as toxic waste clean-up, harvesting, and search and rescue [15].

We setup a foraging task with different types of resources that have to be collected. Robots spend virtual energy at a constant rate and must learn to find and collect resources. When a resource is collected by a robot, a new resource of the same type is placed randomly in the environment so as to keep the number of resource constant throughout the experiments. We experiment with two variants of a foraging task: (i) one in which there are only type A resources, henceforth called *standard foraging task*, and (ii) one in which there are both type A and type B resources, henceforth called *concurrent foraging task*. In the concurrent foraging task, resources A and B have to be consumed sequentially. That is, besides learning the foraging aspects of the task, robots also have to learn to collect resources in the correct order. The energy level of each controller is initially set to 100 units, and limited to the range [0,1000]. At each control cycle, E is updated as follows:

$$\frac{\Delta E}{\Delta t} = \begin{cases} reward & \text{if right type of resource is collected} \\ penalty & \text{if wrong type of resource is collected} \\ \text{-0.02} & \text{if no resource is consumed} \end{cases} \tag{4}$$

where $reward = 10$ and $penalty =$ -10. The constant decrement of 0.02 means that each controller will execute for a period of 500 seconds if no resource is collected since it started operating. Note that the *penalty* component applies only to the concurrent foraging task. To enable a meaningful comparison of performance when groups of different size are considered, the number of resources of each type is set to the number of robots multiplied by 10.

3.2 Experimental Parameters and Treatments

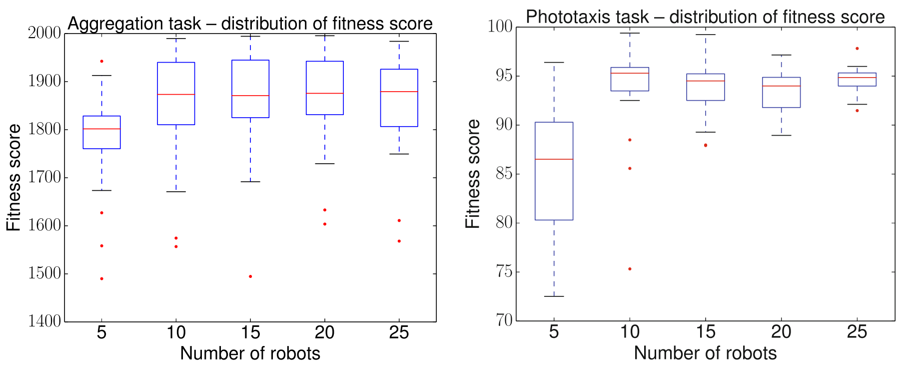
We analyse the impact of the group size on the performance of odNEAT by conducting experiments with groups of 5, 10, 15, 20, and 25 robots. For each experimental configuration, we conduct 30 independent evolutionary runs. Each run lasts 100 hours of simulated time. odNEAT parameters are set as in previous studies [9], including a population size of 40 genomes per robot and a control cycle frequency of 100 ms. Robots operate in a square arena surrounded by walls. In the aggregation and phototaxis tasks, the area of the arena is increased proportionally to the number of robots (5 robots: 9 m^2, 10 robots: 18 m^2, ..., 25 robots: 45 m^2). Notice that if we maintained the same size of the environment, comparisons would not be meaningful. For instance, in the aggregation task, with the increasing density of robots in the environment, the task becomes easier to solve simply because robots encounter each other more frequently. In the phototaxis task, the number of light sources in the environment is also increased proportionally to the number of robots.

4 Experimental Results and Discussion

In this section, we present our experimental results. We analyse: (i) the task performance of controllers in terms of their individual fitness score, (ii) the number of evaluations, that is, the number of controllers tested by each robot before a solution to the task is found, and (iii) the corresponding evolution time. We use the two-tailed Mann-Whitney U test to compute statistical significance of differences between results because it is a non-parametric test, and therefore no strong assumptions need to be made about the underlying distributions.

4.1 Quality of the Solutions and Population-Mixing

We first compare the individual fitness scores of the final controllers. In the aggregation task and in the phototaxis tasks, groups of 5 robots are typically



(a) Aggregation task (b) Phototaxis task

Fig. 1. Distribution of the fitness score of the final controllers in: (a) aggregation task, and (b) phototaxis task.

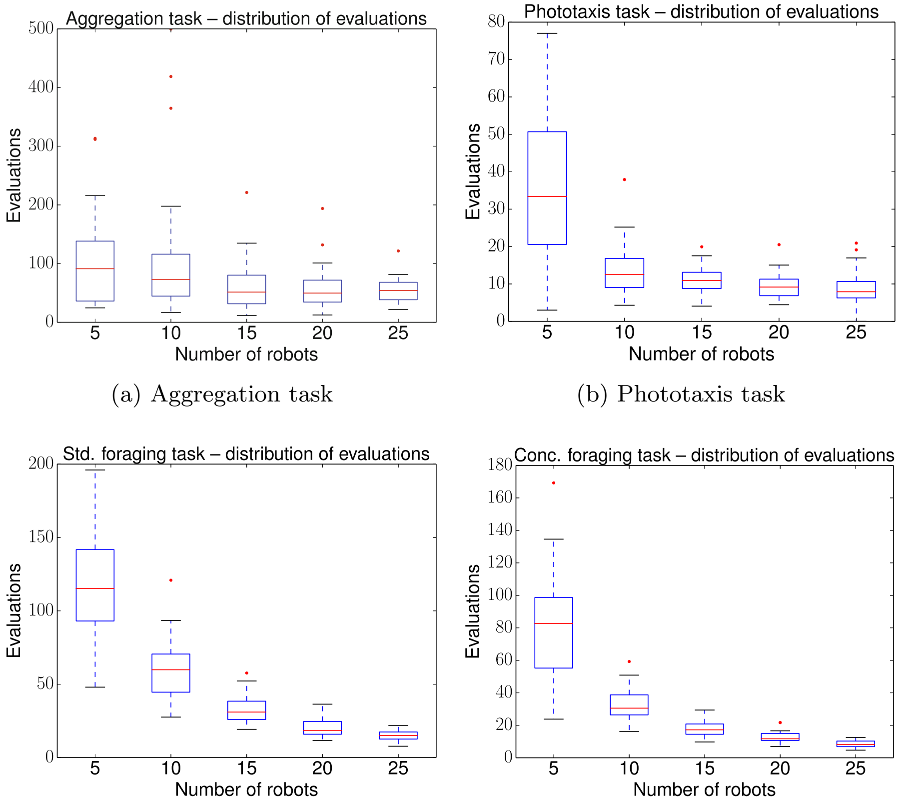
Table 2. Summary of the individual fitness score of final solutions in the two foraging tasks.

Task	Robots	Mean	Std. dev.	Minimum	Maximum
Standard foraging	5	96.03	45.62	41.88	268.02
	10	105.31	62.31	29.57	396.47
	15	107.98	119.79	33.64	981.34
	20	112.06	109.29	36.69	968.27
	25	136.37	158.39	39.03	994.47
Concurrent foraging	5	104.02	73.11	32.62	459.83
	10	112.02	115.61	39.67	949.35
	15	144.54	153.29	38.85	975.81
	20	165.39	165.82	38.77	971.42
	25	179.29	196.58	38.08	978.56

outperformed by larger groups ($\rho < 0.001$, see Fig. 1). In the phototaxis task, groups of 25 robots also perform significantly better than groups with 20 robots ($\rho < 0.01$). Specifically, results suggest that a minimum of 10 robots are necessary for high-performing controllers to be evolved in a consistent manner.

A summary of the results obtained in the two foraging tasks is shown in Table 2. Given the dynamic nature of task, especially as the number of robots increases, the fitness score of the final controllers displays a high variance. The results, however, further show that larger groups typically yield better performance both in terms of the mean and of the maximum fitness scores, and is an indication that decentralised online approaches such as odNEAT can indeed capitalise on larger groups to evolve more effective solutions to the current task.

To quantify to what extent is a robot dependent on the candidate solutions it receives from other robots, we analyse the origin of the information stored in the population of each robot. In the phototaxis task, when capable solutions have been evolved approximately 86.85% (5 robots) to 93.95% (25 robots) of genomes maintained in each internal population originated from other robots, whereas the remaining genomes stored were produced by the robots themselves (analysis of the results obtained in the other tasks revealed a similar trend). The final solutions executed by each robot to solve the task have on average from 87.26% to 89.10% matching genes. Moreover, 39.73% (5 robots) to 47.70% (25 robots) of these solutions have more than 90% of their genes in common. The average weight difference between matching connection genes varies from 2.48 to 4.37, with each weight in [-10, 10], which indicates that solutions were refined by the EA on the receiving robot. Local exchange of candidate controllers therefore appears to be a crucial part in the evolutionary dynamics of decentralised online EAs because it serves as a substrate for collective problem solving. In the following section, we analyse how the exchange of such information enables online EAs to capitalise on increasingly larger groups of robots for faster evolution of solutions to the task.



(a) Aggregation task

(b) Phototaxis task

(c) Standard foraging task

(d) Concurrent foraging task

Fig. 2. Distribution of evaluations in: (a) aggregation task, (b) phototaxis task, (c) standard foraging task, and (d) concurrent foraging task.

4.2 Evaluations and Time Analysis

The distribution of evaluations with respect to the group size is shown in Fig. 2. In the aggregation task, the number of evaluations required to evolve solutions to the task decreases as the group size is increased, and becomes significantly lower when the group size is increased from 10 to 15 robots ($\rho < 0.001$). On average, the number of evaluations decreases from 104 for groups of 5 robots to 55 for groups of 25 robots. The mean evolution time is of 6.22 hours for groups of 5 robots, 2.34 hours for 10 robots, 1.80 hours for 15 robots, 1.48 hours for 20 robots, and 1.12 hours for 25 robots. Hence, adding more robots also enables a significant reduction of the evolution time ($\rho < 0.01$ for every group increment). With the increase in the size of the environment, there is a larger area to search for other robots and to explore. Task conditions become more challenging because, in relative terms, each robot senses a smaller portion of the environment. Robots are, however, still able to evolve successful controllers in fewer evaluations and less evolution time.

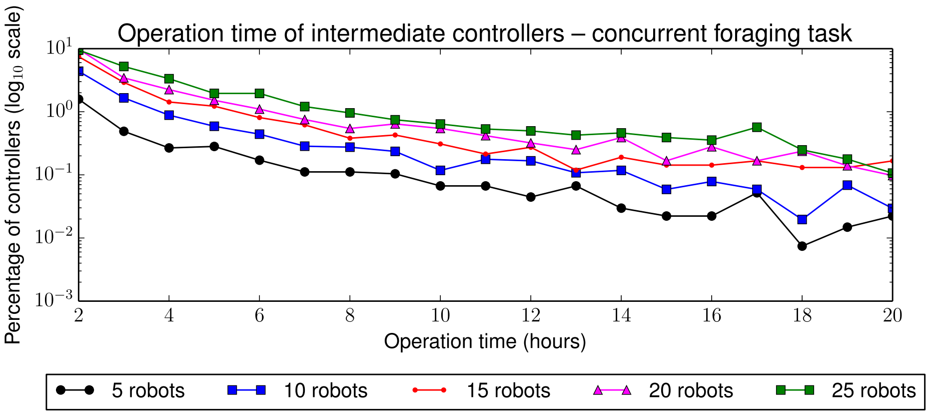


Fig. 3. Operation time of intermediate controllers in the concurrent foraging task. 67% to 96% of intermediate controllers operate for few minutes before they fail (not shown for better plot readability).

The speed up of evolution with the increase of group size also occurs in the phototaxis task. The number of evaluations is significantly reduced ($\rho < 0.001$) with the increase of the group size from 5 to 10 robots (mean number of evaluations of 39 and 14, respectively). The mean evolution time is of 39.16 hours for groups of 5 robots, 9.51 hours for 10 robots, 7.20 hours for 15 robots, 6.30 hours for 20 robots, and 5.27 hours for 25 robots. Similarly to the number of evaluations, the evolution time yields on average a 4-fold-decrease when the group is enlarged from 5 to 10 robots ($\rho < 0.001$). Larger groups enable further improvements ($\rho < 0.001$ for increases up to 20 robots, $\rho < 0.01$ when group size is changed from 20 to 25 robots), but at comparatively smaller rates. Chiefly, the results of the aggregation task and of the phototaxis task show quantitatively distinct speed-ups of evolution when groups are enlarged.

With respect to the two foraging tasks, the distribution of the number of evaluations shown in Fig. 2 is inversely proportional, with a gentle slope, to the number of robots in the group. For both tasks, differences in the number of evaluations are significant across all comparisons ($\rho < 0.001$). In effect, the number of evaluations is reduced on average: (i) from 115 evaluations (5 robots) to 15 evaluations (25 robots) in the standard foraging task, which corresponds to a 7.67-fold decrease in terms of evaluations, and (ii) from 82 evaluations (5 robots) to 8 evaluations in the concurrent foraging task, which amounts to a 10.25-fold decrease. These results show that decentralised online evolution can scale well in terms of evaluations, even when task complexity is increased.

Regarding the evolution time, results show a similar trend for both foraging tasks. On average, the evolution time varies from approximately 35 and 36 hours for groups of 5 robots to 21 and 23 hours for groups of 25 robots. That is, despite significant improvements in terms of the number of evaluations, the evolution time required to evolve the final controllers to the task is still prohibitively long. This result is due to the controller evaluation policy. Online evolution approaches typically employ a policy in which robots substitute controllers at regular time intervals, see [5] for an example. This approach has been shown to lead to incongruous group behaviour and to poor performance in collective tasks that explicitly require continuous collective coordination and cooperation [9]. odNEAT, on the other hand, adopts a different approach by allowing a controller to remain active as long as it is able to solve the task. A new controller is thus only synthesised if the current one fails. As shown in Fig. 3 for the concurrent foraging task, the evaluation policy results in intermediate controllers that operate for a significant amount of time (the standard foraging task displays a similar trend). While 67% to 96% of intermediate controllers only operate for a few minutes (data not shown for better plot readability), there are a few intermediate controllers that operate up to 20 hours of consecutive time before they fail. Although such controllers yield high fitness scores comparable to those of the final solutions, typically 1% to 4% less, they delay the synthesis of more effective solutions.

5 Concluding Discussion and Future Work

In this paper, we presented a case study on the scalability properties and performance of online evolutionary algorithms. We used odNEAT, a decentralised online evolution algorithm in which robots optimise controllers in parallel and exchange candidate solutions to the task. We conducted experiments with groups of up to 25 e-puck-like-robots [12] in four tasks: (i) aggregation, (ii) dynamic phototaxis, and (iii, iv) two foraging tasks with differing complexity.

We showed that larger groups of robots typically enable: (i) superior task performance in terms of the fitness score, and (ii) significant improvements both in terms of the number of evaluations required to evolve solutions to the task and of the corresponding evolution time. There are, however, specific conditions in which intermediate controllers are able to operate up to 20 hours of consecutive time. These controllers yield high performance levels as their fitness score is typically 1% to 4% less than the fitness score of the final solutions. In addition, while additional robots may further speed up evolution, there are specific group sizes after which speed-ups are comparatively smaller. One key research question regarding scalability is therefore how to best leverage *all* robots so that they can learn appropriate behaviours, constitute differentiated groups, and perform cooperative or competitive actions reflecting the structure of the task.

The immediate follow-up work includes studying novel evaluation policies for online evolution of robotic controllers. Regarding odNEAT, the algorithm typically enables a significant reduction in the number of evaluations as the group increases. Hence, if intermediate controllers that operate for long periods of time can be detected and discarded via, for instance, established methods such as early stopping algorithms or racing techniques, there is the potential to enable timely and efficient online evolution in real robots.

Acknowledgments. This work was partly supported by FCT under grants UID/EEA/50008/2013, SFRH/BD/89573/2012, UID/Multi/04046/2013, and EXPL/EEI-AUT/0329/2013.

References

1. Floreano, D., Mondada, F.: Automatic creation of an autonomous agent: Genetic evolution of a neural-network driven robot. In: 3rd International Conference on Simulation of Adaptive Behavior, pp. 421–430. MIT Press, Cambridge (1994)
2. Matarić, M., Cliff, D.: Challenges in evolving controllers for physical robots. Robotics and Autonomous Systems **19**(1), 67–83 (1996)
3. Watson, R.A., Ficici, S.G., Pollack, J.B.: Embodied evolution: Distributing an evolutionary algorithm in a population of robots. Robotics and Autonomous Systems **39**(1), 1–18 (2002)
4. Bianco, R., Nolfi, S.: Toward open-ended evolutionary robotics: evolving elementary robotic units able to self-assemble and self-reproduce. Connection Science **16**(4), 227–248 (2004)
5. Bredeche, N., Montanier, J., Liu, W., Winfield, A.: Environment-driven distributed evolutionary adaptation in a population of autonomous robotic agents. Mathematical and Computer Modelling of Dynamical Systems **18**(1), 101–129 (2012)
6. Silva, F., Urbano, P., Oliveira, S., Christensen, A.L.: odNEAT: An algorithm for distributed online, onboard evolution of robot behaviours. In: 13th International Conference on the Simulation and Synthesis of Living Systems, pp. 251–258. MIT Press, Cambridge (2012)
7. De Jong, K.A.: Evolutionary computation: a unified approach. MIT Press, Cambridge (2006)
8. Silva, F., Duarte, M., Oliveira, S.M., Correia, L., Christensen, A.L.: The case for engineering the evolution of robot controllers. In: 14th International Conference on the Synthesis and Simulation of Living Systems, pp. 703–710. MIT Press, Cambridge (2014)
9. Silva, F., Urbano, P., Correia, L., Christensen, A.L.: odNEAT: An algorithm for decentralised online evolution of robotic controllers. Evolutionary Computation (2015) (in press). http://www.mitpressjournals.org/doi/pdf/10.1162/EVCO_a_00141
10. Silva, F., Correia, L., Christensen, A.L.: Speeding Up online evolution of robotic-controllers with macro-neurons. In: Esparcia-Alcázar, A.I., Mora, A.M. (eds.) EvoApplications 2014. LNCS, vol. 8602, pp. 765–776. Springer, Heidelberg (2014)
11. Silva, F., Urbano, P., Christensen, A.L.: Online evolution of adaptive robot behaviour. International Journal of Natural Computing Research **4**(2), 59–77 (2014)
12. Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., Magnenat, S., Zufferey, J., Floreano, D., Martinoli, A.: The e-puck, a robot designed for education in engineering. In: 9th Conference on Autonomous Robot Systems and Competitions, pp. 59–65, IPCB, Castelo Branco (2009)
13. Duarte, M., Silva, F., Rodrigues, T., Oliveira, S.M., Christensen, A.L.: JBotEvolver: A versatile simulation platform for evolutionary robotics. In: 14th International Conference on the Synthesis and Simulation of Living Systems, pp. 210–211. MIT Press, Cambridge (2014)
14. Groß, R., Dorigo, M.: Towards group transport by swarms of robots. International Journal of Bio-Inspired Computation **1**(1–2), 1–13 (2009)
15. Cao, Y., Fukunaga, A., Kahng, A.: Cooperative mobile robotics: Antecedents and directions. Autonomous Robots **4**(1), 1–23 (1997)