Incremental Evolution of Robot Controllers for a Highly Integrated Task

Anders Lyhne Christensen and Marco Dorigo

IRIDIA, CoDE, Université Libre de Bruxelles,
50, Av. Franklin. Roosevelt CP 194/6,
1050 Bruxelles, Belgium
alyhne@iridia.ulb.ac.be, mdorigo@ulb.ac.be

Abstract. In this paper we apply incremental evolution for automatic synthesis of neural network controllers for a group of physically connected mobile robots called *s-bots*. The robots should be able to safely and cooperatively perform phototaxis in an arena containing holes. We experiment with two approaches to incremental evolution, namely behavioral decomposition and environmental complexity increase. Our results are compared with results obtained in a previous study where several non-incremental evolutionary algorithms were tested and in which the evolved controllers were shown to transfer successfully to real robots. Surprisingly, none of the incremental evolutionary strategies performs any better than the non-incremental approach. We discuss the main reasons for this and why it can be difficult to apply incremental evolution successfully in highly integrated tasks.

1 Introduction

Automatic synthesis of robot controllers is an interesting field, which is likely to some day contribute significantly to the advancement and adoption of robots by industry and the general public. Techniques such as artificial evolution of controllers for autonomous robots can free us from having to understand every detail related to mapping sensory inputs to actuator outputs. Instead, we can focus on more high-level aspects in order to obtain a controller capable of solving a given task.

A robotics setup where artificial evolution can be applied usually starts off with one or more robots and some task. A fitness function is defined, which, given a behavior, assigns a number reflecting the goodness of that behavior with respect to the task. An evolutionary algorithm is then used to find an appropriate controller. The controllers themselves may consist of rule sets, decision trees or similar, but it has become common to use artificial neural networks (ANNs) due to their versatility and tolerance to noisy sensory input. If the controller is represented as an ANN, an evolutionary algorithm can be employed to optimize the weights, and possibly the morphology, of the network. Solutions found in this way can exploit subtle environmental features as they are perceived through the robot's sensors. Therefore, artificial evolution might not only be a time-saving

S. Nolfi et al. (Eds.): SAB 2006, LNAI 4095, pp. 473–484, 2006.

approach for synthesizing controllers: better controllers than those hand-crafted by human developers can be obtained in some cases [1].

The field in which evolutionary techniques are applied in order to develop robotics hardware and/or software is called *evolutionary robotics*. One direction of studies in this field is concerned with cognitive science and psychology [2], while another direction focuses on the use of evolutionary techniques as an engineering tool. Our interest falls in the latter category. We focus on the feasibility and efficiency of different approaches to automatic synthesis of controllers. Hence, our objective is to find evolutionary setups that frequently produce controllers capable of solving a given task.

The task we are concerned with is the evolution of controllers for a number of autonomous mobile robots called *s-bots* [3]. Each *s-bot* has a variety of sensors and actuators. Among these, particularly important is the gripper, which enables multiple robots to physically connect and form an artifact called a *swarm-bot*. In *swarm-bot* formation each *s-bot* maintains autonomous control. Our objective is to obtain controllers for a group of real *s-bots*, in *swarm-bot* formation, that allow them to safely navigate through an arena containing holes. The target location is indicated by a light source.

In our previous work we managed to evolve controllers for the combined phototaxis and hole-avoidance task in simulation, and we showed that the controllers could be transferred successfully to real robots [4]. In that work we also compared the performance of various non-incremental evolutionary algorithms: genetic algorithms [5,6], (μ, λ) evolutionary strategies [7], and cooperative coevolutionary genetic algorithms [8,9]. We found that the (μ, λ) evolutionary strategy in general out-performed the other evolutionary algorithms with respect to the number and quality of the successful solutions found. Furthermore, we tested a number of ANN structures and found that a multilayer perceptron with a hidden layer of two neurons is sufficient to represent successful solutions that can be transferred to real robots. For the study presented in this paper, we use the neural network topology, the fitness function components, and the (μ, λ) evolutionary strategy, which we previously found be the highest performing while resulting in transferable controllers [4], [10].

In the following section we discuss what incremental evolution is and provide examples of studies in which this technique has been applied in the field of autonomous robots. The task and the robotic hardware are explained in Section 3 and 4. Our approach and experimental setup is discussed in Section 5. In Section 6, our results are presented, discussed, and compared to results obtained in our previous work.

2 Incremental Evolution and Related Work

Incremental evolution, applied in order to obtain controllers for a given task, is a method in which evolution begins with a population that has already been trained for a simpler, but in some way related, task [11]. This is done by changing the fitness function during evolution in order to make the task progressively more complex. In this way, bootstrapping problems can possibly be overcome

and evolution can be sped up. The use of incremental evolution can, however, require a substantial engineering effort, because the goal-task has to be organized into a number of sub-tasks of increasing complexity.

Note that some authors use the term *incremental evolution* for algorithms in which the morphology of ANNs is under evolutionary control. Such algorithms include SAGA where the morphology of ANNs is evolved in an incremental manner by the algorithm itself [12]. Another example is the SGOCE paradigm in which ANNs are constructed based on developmental programs that change size and composition during evolution [13]. We will not consider such algorithms here, but instead we use the term incremental evolution to denote evolutionary setups in which the fitness function and/or the environment in which the robots operate are modified during evolution.

A number of studies of incremental evolution of robot controllers has already been performed. Nolfi et al. [14] used incremental evolution to overcome some of the discrepancies between simulation and the real world. Controllers evolved in simulation were transferred to real robots on which evolution was continued. After a few generations, the performance of the controllers on real robots reached the same level as achieved in simulation. Thus, incremental evolution was used to adapt controllers, trained in simulation, to the sensory noise and behavior of the physical robot hardware, which are both impossible to simulate accurately [15].

Harvey et al. [11] evolved controllers incrementally to let a robot distinguish between white triangular and rectangular objects on a dark background. The goal was to evolve controllers that would move robots towards triangles only. The task was divided into sub-tasks where the robots would first learn to orient themselves to face a large rectangle easily detectable by their sensors, then to face and approach a smaller, moving rectangle, and finally to distinguish between rectangles and triangles, and only move towards triangles. Thus, controllers were first trained to follow white rectangles and then later trained not to follow them, but instead to follow triangles only. The authors divided the goal-task into sub-tasks in which recognition and pursuit were learnt in the first evolutionary phase, or *increment*, while discrimination between the two geometric shapes was learnt during later increments. The controllers obtained with incremental evolution were shown to be more robust than controllers trained on the complete task from an initial random population.

Gomez and Miikkulainen [16] used incremental evolution, combined with enforced sub-population and delta-coding, to evolve obstacle avoidance and predator evasion. Incremental evolution was performed by first evolving populations of neurons capable of avoiding a single enemy moving at low speed on a discrete 10x10 grid. The size of the grid was then increased to a 13x13 grid and another enemy was added. Two increments followed in which the speed of the two enemies was increased. The authors found that evolving controllers for the complete task directly was infeasible, while incremental evolution yielded satisfactory results.

In this paper we test different approaches to dividing the goal-task into sub-tasks, one inspired by Harvey et al. [11], which we denote *behavioral decomposition*, and one inspired by Gomez and Miikkulainen [16], which we call *envi-*

ronmental complexity increase. By performing incremental evolution we hope to guide the evolution search towards regions of the fitness landscape containing successful solutions. This should make the evolutionary process more efficient and should therefore increase the likelihood that an evolutionary run finds a good solution.

3 The Task

A group of physically connected *s-bots* should be able to navigate through each of the four arenas shown in Fig.1. The *s-bots* are physically connected using the rigid grippers mounted on the *s-bots*. The group is initially located in a starting zone and should navigate to the location of the light source without falling into any holes or over the side of the arena. Phototaxis and obstacle avoidance for physically connected robots has previously been studied in simulation by Baldassarre et al. [17].

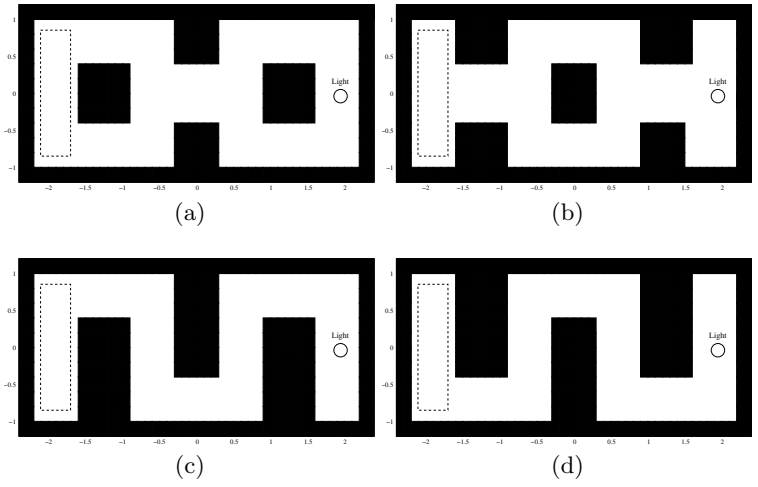

Fig. 1. The four arenas used to evolve controllers. Each of the arenas measures 480x240 cm. The dark areas denote holes, while the white patches denote the arena surface on which the robots can move. The *swarm-bot* must move from the initial location shown on the left-hand side of each arena to the light source on the right without falling into any of the holes or over the edge of the arena.

4 The Robots

In this study, we develop controllers for the SWARM-BOTS platform [3]. Fig. 2 shows photos of an *s-bot* and a *swarm-bot*. Each *s-bot* is equipped with four infrared ground sensors, one pointing 45 degrees forward, two pointing straight downward, and one pointing 45 degrees backward. The ground sensors are mounted between the differential treels© (a combination of tracks and wheels, see Fig. 2).

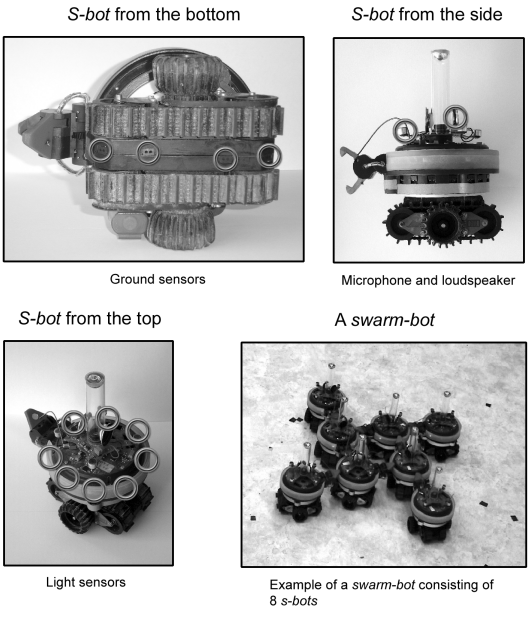

Fig. 2. Different views of an *s-bot* highlighting the location of the sensors used and a *swarm-bot*. An *s-bot* has a diameter of 120 mm and a height of 190 mm.

Microphones and speakers allow *s-bots* to emit and perceive sounds. An *s-bot* can sense forces exerted on its body in the horizontal plane via a traction sensor. These forces allow the *s-bot* to gauge the direction of motion of the rest of the group. Thus, each *s-bot* can align its own direction of motion to that of the other *s-bots*, allowing the *swarm-bot* to move coordinately. The traction sensor is mounted inside the robot between the bottom part (the chassis) and the top part (the turret). The turret can rotate independently of the chassis: up to 180 degrees in each direction from the neutral position. The result of an action in a given situation is likely to depend on the current rotational difference between the top and bottom parts of the *s-bot*. We therefore use two sensors that read the rotational difference in the clockwise and counter-clockwise directions, respectively, at every control step. The relative direction of the target, identified by a light source, is perceived via 8 light-sensors distributed evenly around the plastic ring on the chassis of the *s-bot* as shown in Fig. 2.

The ground sensors are located directly under the *s-bot*, which means that the *s-bot* will only detect the presence of a hole once it is already partly over it. If a single robot tries to navigate through an arena containing holes, it is very likely to fall into a hole unless it approaches the hole perpendicularly. In *swarm-bot* formation, however, the *s-bots* should be able to cooperate to safely navigate through the arena and reach the location of the light source.

We have preprogrammed the *s-bots* to emit a sound, which can be perceived by the other *s-bots* in the *swarm-bot*, when the presence of a hole is detected. This

has previously been found to be an efficient aid when evolving hole-avoidance for a *swarm-bot* [18].

5 Methodology

5.1 Preliminary Fitness Function Components

In this section we present four components of the fitness function, which are used in the incremental evolution setups. In our previous studies, we found that these fitness components assist evolution in finding controllers capable of solving the combined phototaxis and hole-avoidance task on real robots [10], [4].

The first component scores controllers depending on how close they manage to get to the light source. In case they manage to get in the immediate vicinity (within 50 cm) of the light source they are scored based on how fast they do so:

$$f_{light} = \begin{cases} 1 - \frac{min\ distance}{initial\ distance} & \text{if the light source is not reached,} \\ 2 - \frac{time\ light\ is\ reached}{total\ time} & \text{otherwise,} \end{cases} \quad (1)$$

where *total time* is 240 seconds. A component penalizing controllers for falling into holes:

$$f_{stayalive} = \begin{cases} 0.5 \text{ if the } \textit{swarm-bot} \text{ falls into a hole,} \\ 1.0 \text{ otherwise.} \end{cases} \quad (2)$$

Previous studies have shown that coordinated-motion in a group of connected *s-bots* can be obtained by minimizing the traction between the *s-bots* [19]. The traction forces are measured in the two dimensions of the horizontal plane with 0 corresponding to no traction perceived and 1 to the maximum traction force perceivable. At each control step i, we record the maximum traction τ_i^{max} perceived by any of the *s-bots* in the simulation:

$$f_{minimizetraction} = \frac{\sum_i (1 - \tau_i^{max})}{total\ number\ of\ control\ steps}. \quad (3)$$

The three components above are multiplied to form the function f_{final}:

$$f_{final} = f_{light} \cdot f_{stayalive} \cdot f_{minimizetraction}. \quad (4)$$

Finally, we introduce an additional fitness component f_{move} that is used in one of our proposed incremental evolutionary setups. This component rewards coordinated-motion and exploration by measuring the distance covered, measured in a straight line, during different time intervals. Initial experiments showed that measuring the distance moved during multiple time intervals is necessary in order to prevent circular paths and therefore three "good" intervals were found by trial-and-error. Every 7, 13 and 29 seconds, the position of the *swarm-bot* is compared to its position respectively 7, 13, and 29 seconds earlier. The controller achieves a fitness score based on the accumulated distances covered during these intervals divided by the maximum theoretical distance coverable.

5.2 Sub-task Divisions

We divide our goal-task in two different ways based on *behavioral decomposition* and *environmental complexity increase*, respectively.

Behavioral decomposition: Assuming that a successful overall behavior can be decomposed into the sub-behaviors: coordinated-motion, hole-avoidance, and phototaxis, it is possible that these behaviors can be learnt in an incremental fashion. That is, the first learning task is concerned with coordinated-motion in an arena without holes under a fitness function that rewards coordinated-motion ($f_{minimizetraction} \cdot f_{move}$). Once a satisfactory solution has been found, holes are added and the fitness function is extended with a component, which rewards controllers that avoid steering the *s-bots* into holes ($f_{stayalive}$). Finally, phototaxis too is rewarded and the evolved controllers should be able to solve the goal-task (f_{final}). Hence, we assume that the most fundamental task is coordinated-motion, since coordinated-motion is needed for performing both phototaxis and hole-avoidance, followed by combined coordinated-motion and hole-avoidance, and finally the goal task (including phototaxis) in an arena containing holes[1].

Environmental complexity increase: Evolution is started in one of two simple arenas, one containing no holes and the other containing a single hole. More holes and different arena layouts are added as evolution finds solutions. The purpose of applying incremental evolution in this manner is to shape the initial fitness landscapes in such a way that solutions are easier to find because the task is less difficult. When the complexity of the arena is increased, we expect the evolutionary search to resume in region(s) of the fitness landscapes closer to good solutions than a random population would cover. This way of performing incremental evolution could, for instance, prevent evolutionary runs in which the *s-bots* fail to coordinate and move, because in the first increments *swarm-bots* are less likely to encounter a hole. Evolutionary pressure is therefore towards controllers that cause the *swarm-bot* to move. In all increments fitness scores are computed by f_{final}.

For our experiments with environmental complexity increase we use the additional arenas shown in Fig. 3. Two different evolutionary setups are used to test incremental evolution based on environmental complexity increase. One setup consists of six increments while the other consists of three. In the setup comprising six increments, fitness scores of individuals are computed based on trials in the simplified arenas shown in Fig. 3a, 3b, 3c, and 3d, in the first four increments, respectively. In the 5th increment individuals are scored in two of the final arenas shown in Fig. 1a and 1b during the fitness evaluation, while in the 6th and final increment all four arenas shown in Fig. 1 are used. In the environmental complexity increase setup consisting of three increments, a population

[1] We experimented a different ordering of sub-tasks and with an initial increment in which only $f_{minimizetraction}$, as opposed to $f_{minimizetraction} \cdot f_{move}$, was used. However, the behavioral decomposition described above was the highest performing of those tested.

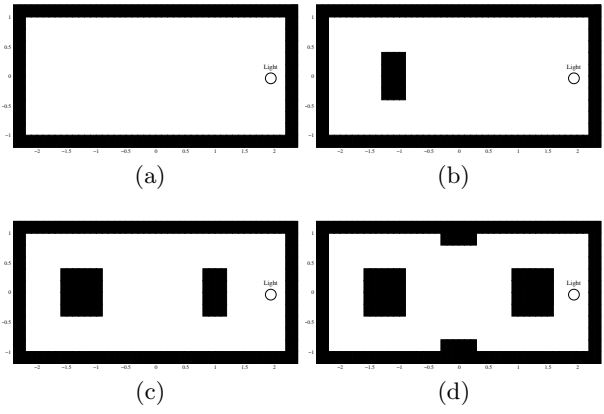

Fig. 3. Additional arenas used for incremental evolution based on environmental complexity increase. A population is first trained in (a) until an acceptable performance is achieved, then in (b) and so on.

starts in the arena shown in Fig. 3b containing a single hole. In the second increment the arena shown in Fig. 3d containing multiple holes is used. Finally, in the last increment, the goal-task is used and individuals are evolved based on fitness scores in all of the four arenas shown in Fig. 1.

We base the transition from one increment to the next on the performance of the population on the current sub-task. The fitness components we use are all relatively noisy. We have to take this into account to avoid that a noisy fitness function makes evolution move from one increment to the next before the current sub-task has been truly learnt. We therefore require the fitness score of the best performing individual to be above a certain threshold for 10 consecutive generations before a transition is made. The thresholds are different for each increment and they are determined based on the fitness function, stagnation of fitness scores during trial runs, and visual inspection of strategies found during the trial runs. Thus, the task of finding these thresholds is, like finding a suitable sub-division of the goal-task, an engineering effort.

For each of the evolutionary setups described above, we run 20 evolutions. Each evolutionary run comprises 1000 generations with a population size of 100 individuals. In all cases, we have used a (μ, λ) evolutionary strategy with $\mu = 20$ and a mutation rate of 15% on a chromosome of floating-point genes. This evolutionary algorithm was found to be the highest performing in our previous study [4]. All evolutionary runs are conducted in our software simulator `TwoDee` [10].

6 Results

In order to compare the performance of the controllers evolved in the different evolutionary setups, we took the highest scoring controller from each of the final generations, post-evaluated them 25 times in each of the four arenas shown in

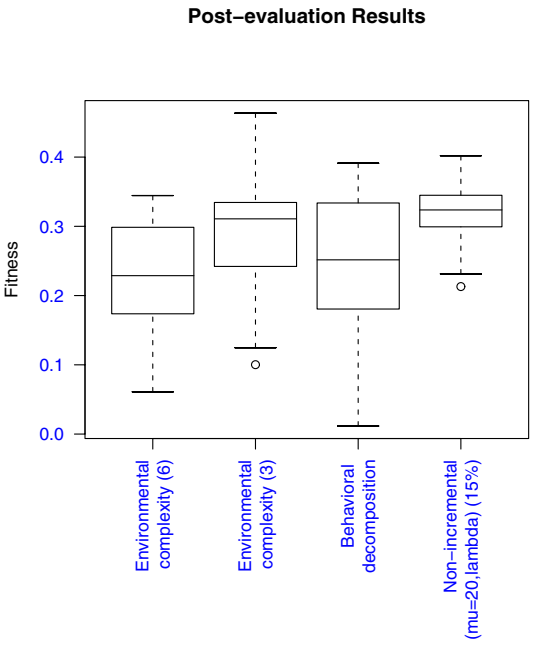

Fig. 4. Box-plot of post-evaluation results for incremental evolution through environmental complexity increase (results for both six and three increments, see text) and behavioral decomposition. We have included the results for non-incremental evolutionary runs using a (μ, λ) evolutionary strategy with a mutation rate of 15%. Each box comprises observations ranging from the first to the third quartile. The median is indicated by a bar, dividing the box into the upper and lower part. The whiskers extend to the farthest data points that are within 1.5 times the interquartile range. Outliers are shown as circles.

Fig. 1, and recorded their average fitness scores. The results for incremental evolution through behavioral decomposition and through environmental complexity increase are shown on the box-plot in Fig. 4. Each box represents post-evaluation results for 20 evolutionary runs. We have included results for the (μ, λ) evolutionary strategy with $\mu = 20$, obtained without the use of incremental evolution.

An example of a successful strategy can be seen in Fig. 5. All evolved controllers capable of performing integrated hole-avoidance and phototaxis displayed a similar strategy: The *swarm-bot* moves coordinately towards an edge of the arena and follows that edge in the direction of the light until the light source is reached.

As it can be seen in Fig. 4, on average the incremental evolutions did not produce better controllers than the evolutionary runs without increments. In the following, we discuss why this is the case.

In the evolutionary setup where integrated phototaxis and hole-avoidance behaviors were evolved incrementally based on behavioral decomposition, we assumed that a successful behavior can be decomposed into coordinated-motion,

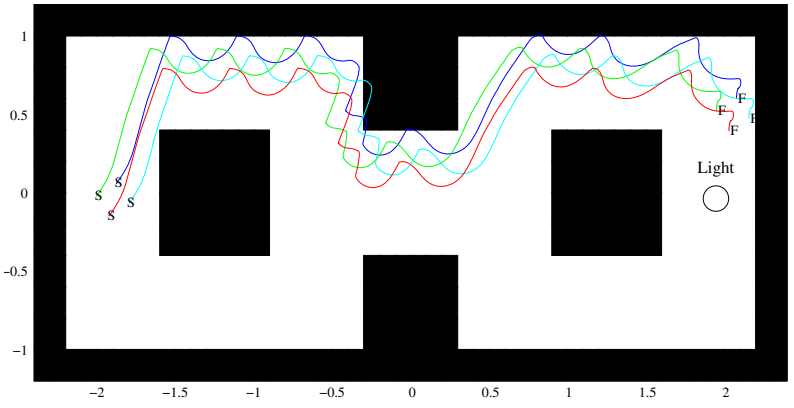

Fig. 5. Example of how a controller capable of solving the task behaves using a simple, but nonetheless general and successful, hole-following strategy

hole-avoidance, and phototaxis. If we take a closer look at the successful solutions found by artificial evolution (an example is shown in Fig.5), it is questionable if such a decomposition is valid and if a suitable decomposition can be found at all. Regardless of the initial position of the *swarm-bot*, the initial orientation of the *s-bots*, and the layout of the arena, the result we observe is always that the *swarm-bot* starts moving left (or right) with respect to the light. Therefore, it appears that the *s-bots* mainly coordinate based on the sensed direction of the light, which serves as a global point of reference, and not on the readings of the traction sensors. Hence, in a successful integrated behavior, coordinated-motion is partly a by-product of phototaxis and it is therefore not beneficial to evolve the two behaviors independently.

In the evolutionary setup based on environmental complexity increase with six increments, only 11 out of the 20 evolutionary runs reached all increments. In the remaining 9 setups the populations did not reach an adequate performance in one of the previous increments. If an evolutionary run does not reach the final increment, then the controllers produced by this run have not been evolved with respect to the goal-task but only with respect to a simpler task. We assume that such controllers obtain a lower post-evaluation score. In the evolutionary setups using only three increments, 17 of the 20 evolutionary runs reached the final increment before the 1000th generation.

No assumptions regarding decomposition of behaviors were made in the experiments for incremental evolution through environmental complexity increase. Nonetheless, the resulting controllers did not on average perform better than controllers evolved non-incrementally. According to the results in Fig. 4, it appears as if the results for incremental evolution through environmental complexity increase are not better than the results for the non-incremental approach. This is surprising given that we started evolution in a simplified arena and increased the complexity gradually in order to avoid bootstrapping issues and assist evolution in finding good solutions. We believe the major reason for the lack of superior

performance is that some solutions found in the earlier increments are in fact not in regions of the fitness landscape that contain successful strategies for later increments. To illustrate this, take for instance the arena shown in Fig. 3b containing a single hole. In this arena, successful solutions include moving directly towards the light and then to move either left or right around the hole if/when it is encountered. However, this strategy does not work in the arenas shown in Fig. 1c and 1d, which both contain a number of turns and which cannot be solved by controllers that turn either only left or only right around holes. Given the reactive nature of the ANN controller, such strategies cause the *swarm-bot* to get trapped in one of the corners. In this way, incremental evolution through environmental complexity increase does in fact results in evolution taking a detour, because the highest scoring solutions in the initial increments are not simpler versions of successful behaviors in later increments.

The results of our study illustrate a fundamental issue related to applying artificial evolution in the field of robotics: The fact that evolution can find solutions, which we as human developers would not have anticipated, is a double-edged sword. On the one hand we can obtain novel solutions, while on the other it can complicate the applicability of techniques like incremental evolution for the very same reason.

Acknowledgements

Anders Christensen acknowledges support from COMP2SYS, a Marie Curie Early Stage Research Training Site funded by the European Community's Sixth Framework Programme (grant MEST-CT-2004-505079). The information provided is the sole responsibility of the authors and does not reflect the European Commission's opinion. The European Commission is not responsible for any use that might be made of data appearing in this publication. Marco Dorigo acknowledges support from the Belgian FNRS, of which he is a Research Director.

References

1. Nolfi, S., Floreano, D.: Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines. MIT Press/Bradford Books, Cambridge, MA (2000)
2. Harvey, I., Di Paolo, E.A., Wood, R., Quinn, M., Tuci, E.: Evolutionary robotics: A new scientific tool for studying cognition. Artificial Life **11**(1-2) (2005) 79–98
3. Mondada, F., Gambardella, L.M., Floreano, D., Nolfi, S., Deneubourg, J.L., Dorigo, M.: The cooperation of swarm-bots: Physical interactions in collective robotics. IEEE Robotics and Automation Magazine **12**(2) (2005) 21–28
4. Christensen, A.L., Dorigo, M.: Evolving an integrated phototaxis and hole-avoidance behavior for a swarm-bot. In: Proceedings of ALIFE X, MIT Press, Cambridge, MA (2006) In press
5. Goldberg, D.E.: The Design of Innovation: Lessons from and for Competent Genetic Algorithms. Kluwer Academic Publishers, Boston, MA (2002)
6. Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press, Cambridge, MA (1996)

7. Schwefel, H.P.: Evolution and Optimum Seeking. Wiley & Sons, New York (1995)
8. Potter, M.A., De Jong, K.: A cooperative coevolutionary approach to function optimization. In: Proceeding of the Third Conference on Parallel Problem Solving from Nature – PPSN III. Volume 866 of LNCS, Springer Verlag, Berlin, Germany (1994) 249–257
9. Potter, M.A., De Jong, K.: Cooperative coevolution: An architecture for evolving coadapted subcomponents. Evolutionary Computation **8**(1) (2000) 1–29
10. Christensen, A.L.: Efficient neuro-evolution of hole-avoidance and phototaxis for a swarm-bot. Technical Report TR/IRIDIA/2005-14, IRIDIA, Université Libre de Bruxelles, Belgium (2005) DEA Thesis.
11. Harvey, I., Husbands, P., Cliff, D.: Seeing the light: artificial evolution, real vision. In: Proceedings of the third international conference on Simulation of adaptive behavior: From animals to animats 3, Cambridge, MA, MIT Press (1994) 392–401
12. Harvey, I.: Species adaptation genetic algorithms: a basis for a continuing SAGA. In Varela, F.J., Bourgine, P., eds.: Proceedings of the First European Conference on Artificial Life. Toward a Practice of Autonomous Systems, Paris, France, MIT Press, Cambridge, MA (1992) 346–354
13. Kodjabachian, J., Meyer, J.A.: Evolution and development of neural controllers for locomotion, gradient-following, and obstacle-avoidance in artificial insects. IEEE Transactions on Neural Networks **9**(5) (1998) 796–812
14. Nolfi, S., Floreano, D., Miglino, O., Mondada, F.: How to evolve autonomous robots: Different approaches in evolutionary robotics. In: Proceedings of Artificial Life IV, Cambridge, MA, MIT Press/Bradford Books (1994) 190–197
15. Jakobi, N.: Evolutionary robotics and the radical envelope-of-noise hypothesis. Adaptive Behavior **6**(2) (1998) 325–368
16. Gomez, F., Miikkulainen, R.: Incremental evolution of complex general behavior. Adaptive Behavior **5** (1997) 317–342
17. Baldassarre, G., Parisi, D., Nolfi, S.: Distributed coordination of simulated robots based on self-organisation. Artificial Life (2006) In press
18. Trianni, V., Tuci, E., Dorigo, M.: Cooperative hole avoidance in a swarm-bot. Robotics and Autonomous Systems **54**(2) (2006) 97–103
19. Trianni, V., Labella, T.H., Dorigo, M.: Evolution of direct communication for a swarm-bot performing hole avoidance. In: Ant Colony Optimization and Swarm Intelligence – Proc. of ANTS 2004 – 4th Int. Workshop. Volume 3172 of LNCS, Springer Verlag, Berlin, Germany (2004) 131–142