

Noname manuscript No.  
(will be inserted by the editor)

# The Distributed Co-Evolution of an On-Board Simulator and Controller for Swarm Robot Behaviours

Paul J. O'Dowd · Matthew Studley · Alan F.T. Winfield

Received: date / Accepted: date

**Abstract** We investigate the reality gap, specifically the environmental correspondence of an on-board simulator. We describe a novel distributed co-evolutionary approach to improve the transference of controllers that co-evolve with an on-board simulator. A novelty of our approach is the the potential to improve transference between simulation and reality without an explicit measurement between the two domains. We hypothesise that a variation of on-board simulator environment models across many robots can be competitively exploited by comparison of the real controller fitness of many robots. We hypothesise that the real controller fitness values across many robots can be taken as indicative of the varied fitness in environmental correspondence of on-board simulators, and used to inform the distributed evolution an on-board simulator environment model without explicit measurement of the real environment. Our results demonstrate that our approach creates an adaptive relationship between the on-board simulator environment model, the real world behaviour of the robots, and the state of the real environment. The results indicate that our approach is sensitive to whether the real behavioural performance of the robot is informative on the state real environment.

**Keywords** Evolutionary Robotics · Swarm Robotics · Distributed Evolution · Online Evolution

P. J. O'Dowd  
University of the West of England, Bristol. UK  
Tel.: +441173284704  
E-mail: paul3.o'dowd@uwe.ac.uk

M. Studley  
Tel.: +441173283380  
E-mail: matthew2.studley@uwe.ac.uk

A. F. T. Winfield  
Tel.: +441173282644  
E-mail: alan.winfield@uwe.ac.uk

## 1 Introduction

Swarm robotics is regarded as being a difficult class of robotic system to design. Multiple autonomous robots are expected to produce useful group behaviour as an emergent consequence of their interactions. From a designer's point of view, only a single robotic agent is defined and the result of complex interactions must be extrapolated outwards. Through decentralisation, self-organising robotic systems they are cited as being robust, flexible, and scalable; although this is not without caveats[1].

Evolutionary computation is an appealing design approach to swarm robotics. The design outcome can be defined as a group behaviour, and an evolutionary algorithm addresses the hard problem of a solution for the individual robot. Often a simulation is used and provides convenient access to group-level evaluative metrics [16][14][17][6][23].

The use of simulation in evolutionary robotics has been heavily debated. To avoid a prohibitively slow simulation it must be designed to balance the accuracy of the representation against the time of computation, inherently encapsulating errors [13]. Inaccuracies in a simulation can be exploited by the evolutionary process, producing robotic solutions with a discrepancy between simulated and actual performance. This issue of discrepancy is referred to as the *reality gap*[7], and discussed in terms of the transferability of solutions[9].

The alternative to utilising a simulation is to evaluate evolved solutions directly on a robot, termed embodied evolution by Watson et al[24]. Eiben et al [3] elaborate on embodied evolution and discuss three binary features to clarify where, when and how an embodied evolutionary algorithm can be implemented:

*Online / Offline*: whether the evolutionary algorithm operates as part of their “real” operation, or as a prior design phase of operation before actual deployment.

*On-board / Off-board*: whether the algorithm executes on the actual robot hardware, or is computed external to the robot with only the resultant solution evaluated on the robot hardware.

*Encapsulated / Distributed*: whether a robot operates the evolutionary algorithm independently on its own hardware, or if the evolutionary algorithm is designed to operate across a group of robots.

There have been several recent investigations into online, on-board, distributed evolutionary robotics motivated by the vision of a multi-robot system capable of continuous unsupervised evolutionary adaptation [10][8][20][19] [4][5]. Whilst the online on-board distributed approach is suitable for swarm robotics, three problematic issues are highlighted, and form part of the underlying motivation to develop our work:

*Spatial* : Referred to as the boot-strapping problem. The spatial mobility of robots is determined by the solutions developed by the evolutionary algorithm. Early explorative evolutionary development often creates incorrect sensory-motor mappings, causing robots to collide and spatially interfere with each other. Therefore each successive evaluation occurs in a new non-deterministic environment which can disrupt the reliable evaluation of newly evolved solutions.[10].

*Temporal* : Online evolution is proposed as a mechanism to produce functional behaviour to solve a task, as opposed to a study of evolution in of itself. This applies pressure to generate solutions at a rate comparable to the dynamic change within the task environment [8].

*Selection* : The migration of solutions across the group of robots is non-deterministic since the robots are mobile. Furthermore, because of the noisy evaluation circumstances, the evaluative metric is not reliable between robots[20].

The benefits and shortfalls of the simulated and embodied approaches appears to be leading to a converged methodology. Koos et al[9] define a category of evolutionary robotics as *robot-in-the-loop simulation-based optimisation*, encompassing a body of work that investigates the use of simulated evaluations with periods of evaluation in reality to correct for transference problems.

Evolving robot controllers, Koos et al[9] develop a ‘Simulation To Reality disparity measure’ of transference between an offline off-board simulator and periods

of evaluation in reality, used to bias the evolutionary selection mechanism towards controller solutions with better transference. Evolving walking gait behaviours, Bongard et al[2] develop the ‘estimation-exploration’ algorithm which utilises evaluations in reality to capture limb-joint sensor data to adapt an offline off-board simulation of the robot morphology. Zagal et al[25] develop the ‘Back To Reality’ algorithm, which co-evolves an offline off-board simulation of a quadruped robot and a walking gait controller, by using a single measure of discrepancy between the achieved walking gait in simulation versus reality.

This work concerns advancing an online on-board distributed approach suitable for application in swarm robotics that maintains the vision of an unsupervised evolutionary system. Motivated by the design context of swarm robotics and the previously isolated problems with the online on-board evolutionary approaches, we propose a distributed *robot-in-the-loop simulation-based* methodology. This work presents novelty in extending previous online on-board distributed approaches with an on-board simulator for each robot, allowing controller evaluation to be encapsulated virtually per robot, and selectively transferring a controller on to the same robot for use in reality.

Zagal et al[25] describe the potential utility of an on-board simulator in terms of an incorporated aspect of an embodied robot controller, drawing analogy to the faculty of dreaming in cognitive neuroscience. This work proposes a different utility; an on-board simulator may aid the aforementioned problems with an on-line on-board distributed evolutionary approach. Spatial problems could be minimised by conducting the majority of evaluations within an on-board simulation; temporal attributes could be accelerated by allowing evaluations to happen within an on-board simulation; selection could be improved by allowing a communicated solution from one robot to be re-evaluated by the recipient robot’s on-board simulator.

This work addresses the primary issue of the reality gap associated with an on-board simulator. Zagal et al[25] address the reality gap of an off-board simulator with a co-evolutionary approach encapsulated on a single robot. Their co-evolutionary approach uses the difference in fitness of a robot controller between simulation and reality (a measure of transference) to steer the evolution of the simulator. Importantly, their approach evaluates a population of controller solutions in reality, and then the same controller population is evaluated within an evolving population of simulators to create an explicit measure of transference. This paper also proposes a co-evolutionary approach to the reality gap, but has novelty in distributing the on-board sim-

ulator evolution across a swarm of robots. Therefore each robot owns only one on-board simulator at any time, and the number of robots represents the total evolutionary population of simulator genotypes. This removes the need to correlate which controller is the product of which simulator. Furthermore our approach does not utilise an explicit measure of transference between the two. We propose that the on-board simulator can gain improving transference by competitive distributed co-evolution between many robots, by taking the success of a robots evolved real behaviour as an implicit indicator of the fitness of the associated on-board simulator. We are interested in investigating this distributed and implicit selection mechanism of on-board simulators to avoid the need to evaluate multiple on-board simulators per robot, and to leverage the variety of evaluations across many robots against the possibility of uninformative circumstances of a single robot.

We are able to make a distinction in our approach by the aspect of the reality gap we wish to address. We propose that the reality gap can be decomposed in to three elements of correspondence between reality and simulation:

*Robot-robot correspondence* : Refers to physical robot aspects, such as differences in morphology. The work of Bongard et al[2] is a primary example of a robot that is able to adapt a self-model of morphology.

*Robot-environment correspondence* : Refers to differences in the dynamic interactions between a robot and the environment, both sensory and through actuation. Bongard et al[2] demonstrates how the relationship between morphology and a known state of the environment can be usefully exploited. Zagal et al[25] co-evolve the physical dynamics of a simulator coupled to walking gait evolution.

*Environment-environment correspondence* : Relates the representation of salient features of the environment. Notably, such relationships are not intended as a navigational map. Rather, it should represent characteristics of the environment that can be altered behaviours over time, such as spatial density.

To date we have found no examples that specifically adapt a simulator for environment-environment correspondence. The environment is of special significance for swarm robotics as it is often used as the cue, memory or coordinating aspect of a system comprised of self-organising agents[21]. This work documents an experimental investigation on the environmental correspondence of the reality gap using a swarm of physically simplistic robots.

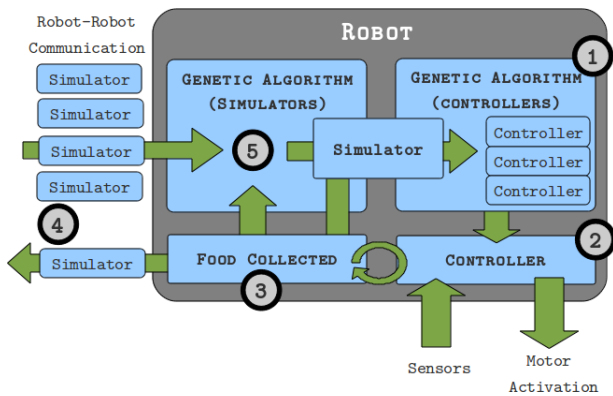
In this work a swarm of ten real e-puck robots are used to investigate the distributed co-evolution of an

on-board simulator to adapt to a changing task environment through the coupled evolution of controller solutions. The correspondence between simulation and reality has a consequence on the transferability of controller solutions. If the on-board simulator environment model can be appropriately evolved, we can expect to observe changes in the resultant behaviour from co-evolved robot controllers to complete a task. A novelty of the approach is the the potential to improve transference between simulation and reality without an explicit measurement between the two domains. We hypothesise that the variation of on-board simulator environment models across many robots can be competitively exploited by comparison of the real controller fitness of many robots. We hypothesise that the real controller fitness values across many robots can be taken as indicative of the varied fitness in environmental correspondence of on-board simulators, and used to inform the distributed evolution an on-board simulator environment model without explicit measurement of the real environment. To test this hypothesis, the foraging problem is selected, where a swarm of robots must discover and deposit food items to a designated nest site, and have the potential to use a moving light source as an environmental aid.

The remainder of this article is structured as follows: Section 2 provides a brief overview of our distributed co-evolutionary approach to the evolution of on-board simulator and controller. Section 3 describes the hardware used to conduct the experiments. Section 4 details the specifics of the co-evolutionary algorithm used and the settings used for the experiments. Section 5 details the results gained and ends with a discussion. Section 6 draws conclusions from our presented work and gives projections for future work.

## 2 Distributed Co-Evolution of an On-Board Simulator and Controller

This section provides an overview, and specific details of the implementation of these algorithms are detailed in the following sections. The proposed co-evolutionary method has two evolutionary components. One genetic algorithm is encapsulated on each robot and evolves a population of controller genotypes within a robot's on-board simulator. A second genetic algorithm is distributed across the physical swarm, where each robot owns a single instance of an on-board simulator genotype, and the swarm of robots constitute an evolving population of on-board simulators. These algorithms execute concurrently with each other and the operation of the mobile robot. Fig.1 illustrates the co-evolutionary algorithm in overview.



**Fig. 1** An illustration of the co-evolutionary implementation. Addressing numbered points: 1) A genetic algorithm evolves a local population of controller genotypes through the on-board simulator. 2) The best controller genotype from simulation is transferred to the real robot. 3) A controller fitness in reality, in this work foraging efficiency, is used to indicate the fitness of the associated on-board simulator. 4) A robot transmits and receives on-board simulator genotypes and real fitness values. 5) Synchronised with the end of virtual controller evaluation, the on-board simulator is evolved against the robot’s own perceived fitness and any encountered robots’ fitness values.

Similar to Zagal et al[25], we utilise a fitness metric of the evolved controller behaviour within both evolutionary components. The encapsulated controller evolution is informed by evaluations within the on-board simulator. After each generation of encapsulated simulated controller evolution, a controller is instantiated on the real robot and a real fitness measure of the controller is generated for use with the distributed simulator evolution. The use of a controller fitness to assess the on-board simulator is as opposed to an explicit measurement of correspondence between the on-board simulator environment model and reality, such as the extensive set of sensor recordings used for the estimation-exploration algorithm developed by Bongard et al[2]. We also do not explicitly compare the controller fitness between the on-board simulator and real performance of a robot. Instead we create a competitive system based on the variation of on-board simulators and real evaluations across many robots to attempt to remove the need for explicit correlation.

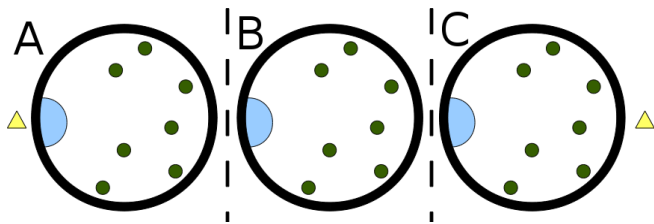
Dissimilar to Zagal et al[25] we distribute the simulator evolution. Therefore each robot owns and instantiates only one on-board simulator genotype at any time, and the number of robots represents the total evolutionary population of simulator genotypes. This implementation detail removes the need to correlate which controller is the product of which simulator, and we make no explicit measure of transference. We hypothesise that the inherent variation in on-board simulators and the real behavioural performance between many

robots can be used to competitively co-evolve towards improving simulator transference. From the encapsulated controller evolution, we choose to use the controller genotype with the highest fitness within the on-board simulator to instantiate on the real robot, resulting in a single instance of real activity of a robot as the sole indicator of the fitness of the on-board simulator. These implementation choices are for an approach that maximises the consistency of a robot’s real behaviour by minimising the interleaving between simulator and controller evaluations and correlation between the two.

Each robot evaluates a population of controller genotypes within its on-board simulator. Within this same time-frame the robot is operating in reality and constructs a real fitness measure. The real fitness measure is broadcast with its current on-board simulator genotype as part of the distributed evolution of on-board simulators. Therefore the swarm constitutes many real fitness assessments (representative of the simulator) occurring in parallel, which is sampled by communication encounters between mobile robots. An encounter is defined by the communication range between robots (25cm), which is necessarily short range for a decentralised self-organising system. Each robot constructs a temporary population of encountered simulator genotypes and their associated real-world controller fitness. The on-board simulator is subjected to its own evolution once the current generation of controller evaluations within the on-board simulator has elapsed. Therefore the population of controller genotypes are evaluated within the on-board simulator within a single real world evaluation of a controller, and the computation of evolution for a single generation of both the controller genotypes and on-board simulator genotype is a momentary synchronisation event in the operation of the robot.

### 3 Experiment Method

We use ten e-puck mobile robots (documented by Mondada et al[15]) each equipped with a Linux extension board for parallel computation and Wi-Fi connectivity (documented by Liu and Winfield[11]). The Linux extension board is used to operate a noise-based[7] minimal simulation written in C (see prior work[18]), and for all evolutionary computation. We use the e-puck infra-red proximity sensors for obstacle avoidance, determining ambient light levels, and for short range communication between robots. The short range infra-red communication is used to initiate further communication between robots over a Wi-Fi network. The Wi-Fi communication provides superior bandwidth but remains decentralised through the locality of the infra-



**Fig. 2** An illustration of the three environment scenarios. Large circular outlines represent the arena enclosure. Small green circles represent food. The blue semi-circle represents the nest area. Yellow triangles represent a light source location (when present).

red communication. A Vicon tracking system monitors the position of e-pucks and is used in conjunction with Wi-Fi to facilitate a virtual sensor by informing a robot if it is spatially located within virtually superimposed food items or the designated nest site.

## 4 Experiments

We investigate the distributed evolution of an on-board simulator environment model against a dynamic task environment through the co-evolution of controller solutions. If the on-board simulator environment model can be appropriately adapted, we can expect to observe changes in the resultant behaviour from co-evolved robot controllers to complete a task. The proposed method does not rely on an explicit measure of transference between simulation and real operation. Rather, it is proposed that the inherent variation in on-board simulators and the real performance between many robots can be used to competitively co-evolve on-board simulators with improving controller transference. To test this hypothesis, the foraging problem[12] is selected, where robots must discover and deposit food items to a designated nest site, and have the potential to use a moving light source as an environmental aid.

### 4.1 Experimental Setup

Around the foraging problem, three basic *environment scenarios* are applied (Fig.2); a light source over the nest site (A), no light source (B), or the light source opposite the nest site (C). The presence of a light source should act as a navigational aid, improving the foraging efficiency of a robot through phototaxis behaviour. The three basic environment scenarios are combined into five *experiment cases*, and a sixth control of fixed random movement obstacle avoidance behaviour without the co-evolutionary approach:

1. *No Light Source*
2. *Light Fixed Over Nest*
3. *Light Fixed Opposite Nest*
4. *Light Over Nest*  $\rightarrow$  *Light Opposite Nest*
5. *Light Opposite Nest*  $\rightarrow$  *Light Over Nest*.
6. *Random Movement*

In the first five experiment cases, the hypothesised outcome is that the distributed on-board simulator evolution should adapt relative to the light stimulus available in the real environment, and the encapsulated controller evolution should exploit the on-board simulator model to evolve behaviours with improving foraging efficiency in the real environment.

### 4.2 Encapsulated Evolution of Robot Controller

The encapsulated evolution of controllers occurs only within the on-board simulator of each robot. For each robot controller genotype to evaluate, one robot is simulated to forage for 60 virtual seconds. Each robot operates a steady state genetic algorithm to adapt a genotype mapping of sensory input to behavioural output, with the following parameters:

- *Genotype Length*: 2 ( $G_0$ ,  $G_1$ )
- *Gene values*: in range [0.00:0.99]
- *Population Size*: 10
- *Mutation Rate*: 20%
- *Mutation*: Gaussian noise, mean=0 s.d.=2
- *Cross-over*: None
- *Selection*: Rank-based elitist, top 4 seed lower 6

An internal Food state signifies if a robot is in possession of a food item.  $G_0$  corresponds to state *Food = True*.  $G_1$  corresponds to state *Food = False*. The values of  $G_0$ ,  $G_1$  are mapped to select a behaviour, as per table 1. These values were chosen for an equal distribution between the possible behaviours.

Value	$G_0$ (Food = True)	$G_1$ (Food = False)
[0.00 : 0.32]	<i>negative phototaxis</i>	<i>negative phototaxis</i>
[0.33 : 0.65]	<i>random search</i>	<i>random search</i>
[0.66 : 0.99]	<i>positive phototaxis</i>	<i>positive phototaxis</i>

**Table 1** Genes  $G_0$ ,  $G_1$  mapping of state to behaviour selection, providing a variety of possible responses to the state *Food* of the robot.

Selection for reproduction is rank based and elitist. 40% of the population is used to overwrite the lower ranking percentage. Each gene of the child genotype is subjected to a 20% chance of a random mutation on a Gaussian distribution (mean = 0, s.d. = 2). Mutation

is the only mechanism to introduce variation. We take these operator parameters from prior related work[18]. The fitness of each genotype is determined by evaluating the performance of the controller phenotype as a single simulated robot in the on-board simulator as summation of deposited food as a function of time:

$$F = \sum_{D=1}^{D_{Total}} T_{Max} - T_D \quad (1)$$

where  $F$  is the derived fitness metric,  $D$  is a deposited food item,  $T_{Max}$  is the evaluation time limit of 60 seconds,  $T_D$  is the recorded time to successfully deposit a food item. Time is used rather than quantity of food for stronger differentiation between efficiency in solutions. When all 10 genotypes have been evaluated in the on-board simulator, the genotype with the highest simulated fitness value is immediately instantiated for use on the real robot.

#### 4.3 Distributed Evolution of On-Board Simulator

The distributed evolution on-board simulators operates across the swarm of mobile robots. A simplistic genetic algorithm operates on each robot, drawing from a temporary population of simulator genotypes constructed through communication encounters with other robots. After each evolution, each robot retains one simulator genotype, discarding the temporary population which is reseeded through subsequent encounters with robots. The following parameters are used:

- *Genotype Length*: 1 ( $S_0$ )
- *Gene value*: in range [0.00:0.99]
- *Population Size*: variable, maximum 10
- *Mutation Rate*: 100%
- *Mutation*: Gaussian noise, mean=0 s.d.=2
- *Cross-over*: None
- *Selection*: Rank-based elitist, 1 retained, others discarded

The environmental model of the on-board simulator is determined by the single gene value mapping of  $S_0$  (see table 2). The mapping values of  $S_0$  to the environment scenarios are chosen for an equal distribution. Each robot maintains the value of  $S_0$  for the duration of a complete generation of controller evaluations within the on-board simulator, after which it is subjected to distributed evolution operators, and the on-board simulator is subsequently re-instantiated with the new mapping. The real robot operates and is evaluated for 60 real-time seconds, which also serves as the time period to encounter other robots and accumulate

foreign  $S_0:F_R$  pairs. Concurrently, an average of 34 real-time seconds are taken to conduct the necessary ten instances of sixty simulated second evaluations of controller genotypes within the on-board simulator.

Value	$S_0$
[0.00 : 0.32]	<i>Light Opposite Nest</i>
[0.33 : 0.65]	<i>No Light</i>
[0.66 : 0.99]	<i>Light Over Nest</i>

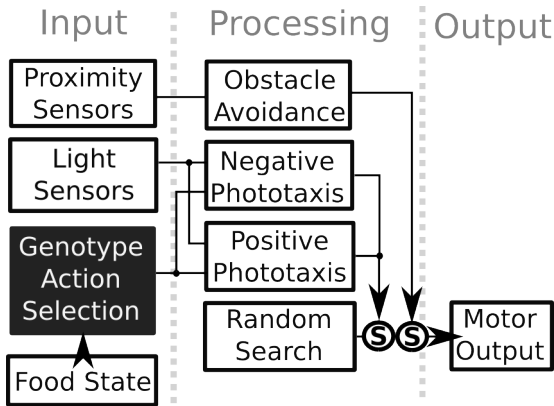
**Table 2** Gene  $S_0$  mapping to the embedded simulator scenario.

As the robot operates in the real world it broadcasts its current  $S_0$  and current real world fitness value  $F_R$ , and receives the  $S_0$  and  $F_R$  values of encountered robots, over a maximum distance of 25cm.  $F_R$  is determined as the robot operates by the same equation used in the encapsulated simulated evaluation (see Eqn.1). A temporary population of 10  $S_0:F_R$  pairs are stored and updated by each robot, representing the variation and fitness of environment models across the swarm. The population size of 10 has been selected for a conveniently matched proportion to the number of robots used in our investigation, and has not been empirically evaluated. Selection from the  $S_0:F_R$  pairs is rank based elitist, and always subjected to a random mutation on a Gaussian distribution (mean = 0, s.d. = 2).

An individual robot compares its own  $S_0:F_R$  pair against the  $S_0:F_R$  values encountered from other robots. Therefore, with fewer than two robots there is no selective pressure to form the distributed evolution of  $S_0$ . A robot’s accumulated population of foreign  $S_0:F_R$  pairs and its own controller  $F_R$  value are cleared at the update transition of controller and on-board simulator environment model.

#### 4.4 Robot Controller

A set of discrete behaviours are pre-defined: *obstacle avoidance*, *random search*, *positive phototaxis* and *negative phototaxis*. The modular behaviours are arranged in a hierarchy of priority within the subsumption architecture illustrated in fig. 3. A behaviour based approach is used to reduce the number of variables in the experiment and maintain a focus on the adaptation of controller solutions with respect to the simulator environment model. A summary of the controller illustrated in fig. 3 is as follows. Obstacle avoidance is activated with the highest priority when triggered by a robot’s proximity sensors. Negative phototaxis and positive phototaxis can be activated depending on the Food State and the



**Fig. 3** An illustration of the robot controller as an implementation of the subsumption architecture.

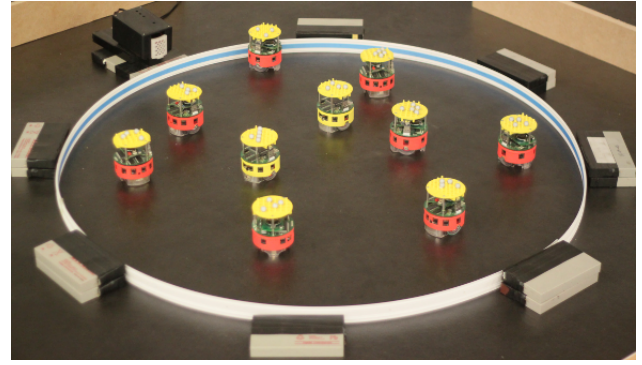
controller genotype mapping. The random search is always active, but can be over-riden by any of the previous behaviours. The same controller mechanism is used for both the simulated robot within the on-board simulator and the real robot. The controller can be adapted by changing the genotype mapping of the Food state to enable the negative phototaxis or positive phototaxis behaviours.

#### 4.5 Experiment Settings

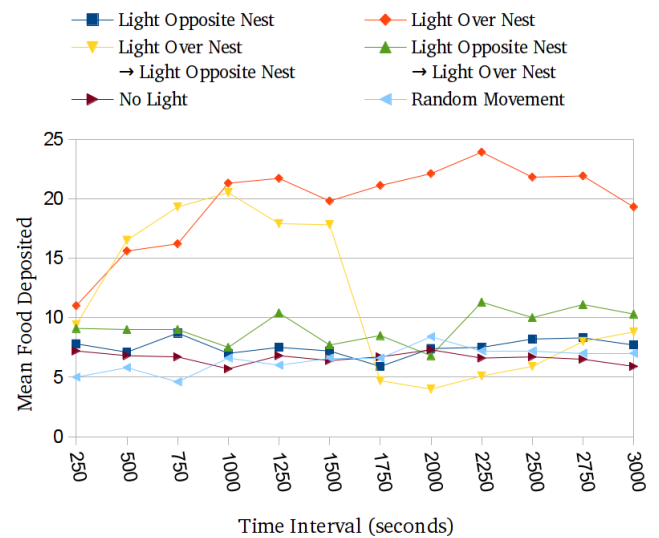
The five experiment cases outlined are each run 10 times for a duration of 50 minutes. If the light sourced is moved, this occurs at the 25 minute mark. The light source is placed either directly behind the nest site or exactly opposite on the other side of the arena. Experiments are conducted within an enclosed circular arena measuring 120cm diameter. The arena is free from obstructions. A single circular nest site is superimposed with a radius of 20cm to intersect the arena boundary and maintains the same coordinates through all experiment runs. Seven food items are randomly placed within the arena. These food items always appear outside the nest area. A total of 10 e-puck robots are used which are randomly positioned and orientated at the beginning of an experiment. All e-pucks are activated by an on-board switch. A photograph of this setup is shown in fig. 4.

## 5 Results and Discussion

Fig.5 plots the mean foraging rate for each experiment case. Using the control case *Random Movement*, which does not use the co-evolutionary approach, the Student's t-test (sample size 50, taking mean foraging efficiency at 60 second intervals) indicates that the case *No*



**Fig. 4** A photograph of the real e-pucks within the arena, and the light source box located in the top left of the picture. The blocks around the arena enclosure are lead-acid batteries used to keep the arena in place.

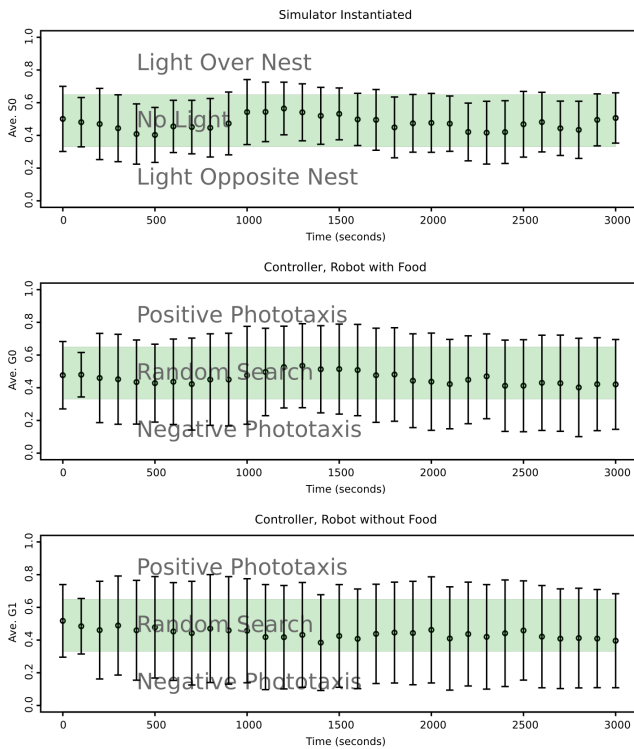


**Fig. 5** Graph plotting the foraging rate, calculated as mean food deposited in 250 second intervals during each experiment case.

*Light* had no significant difference from *random movement* ( $p > 0.5$ ), whilst the other experiment cases differ significantly from *Random Movement* ( $p < 0.005$ ). This suggests that the co-evolutionary approach is able to make beneficial adaptation to the on-board simulator when a light source is present, and improving the transference of controllers. However there is a stark contrast in foraging efficiency dependent on the location of the light source. The light source over the nest appears to double the effective foraging efficiency. The following sections investigate each experiment case.

#### 5.1 No Light Source

Fig. 6 shows that the mean value of  $S_0$  maps to no light source within the on-board simulator consistently throughout the experiment. Another simulator environ-

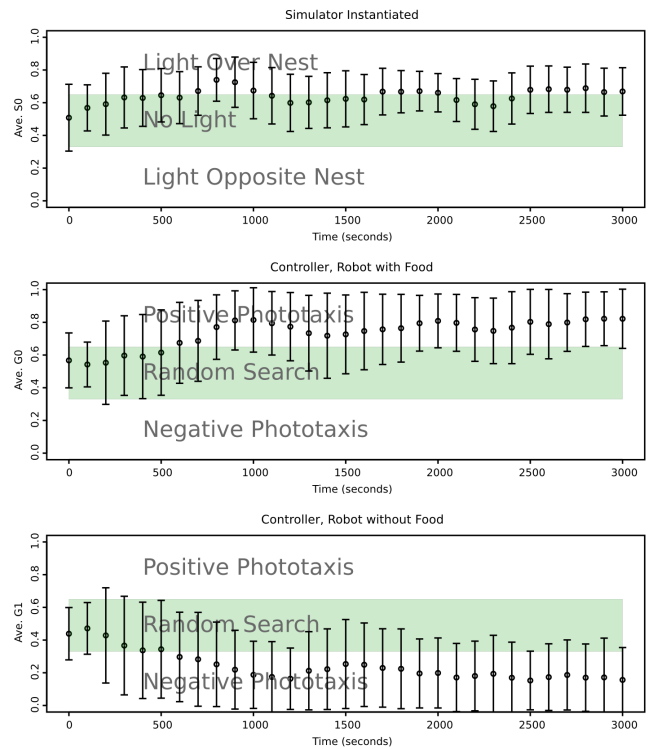


**Fig. 6** *No Light Source*: Three graphs plotting the mean value of the genes  $S_0$ ,  $G_0$  and  $G_1$  over time. The error bars are the standard deviation of the results. The green horizontal bands mark the mapping of the gene value to the controller behaviour or simulator model.

ment mapping would likely lead to the co-evolution of controllers utilising phototaxis within simulation and a poor transference. In this case the on-board simulator has been co-evolved with a strong correlation to the real environment. The plots for  $G_0$  and  $G_1$  show a wide distribution centred on random search behaviours when with or without food. A wide distribution in  $G_0$  and  $G_1$  controller mapping is representative of a poor consensus of which behaviours lead to efficient searching without a light source.

## 5.2 Light Source Fixed Over Nest

Fig. 7 shows that the evolved value of  $S_0$  averages around the boundary mapping value of 0.66 with a distribution that indicates a co-evolved simulator model with a light source over the nest or no light source.  $G_0$  shows a clear trend towards the use of positive phototaxis when with food, and  $G_1$  trends toward negative phototaxis to search for food. The narrow distribution of  $G_0$  and  $G_1$  controller mapping indicates that these behaviours provided a consistent means to inform the distributed evolution of  $S_0$ , and that  $S_0$  gives a strong controller transference. In this experiment case, the co-evolution-



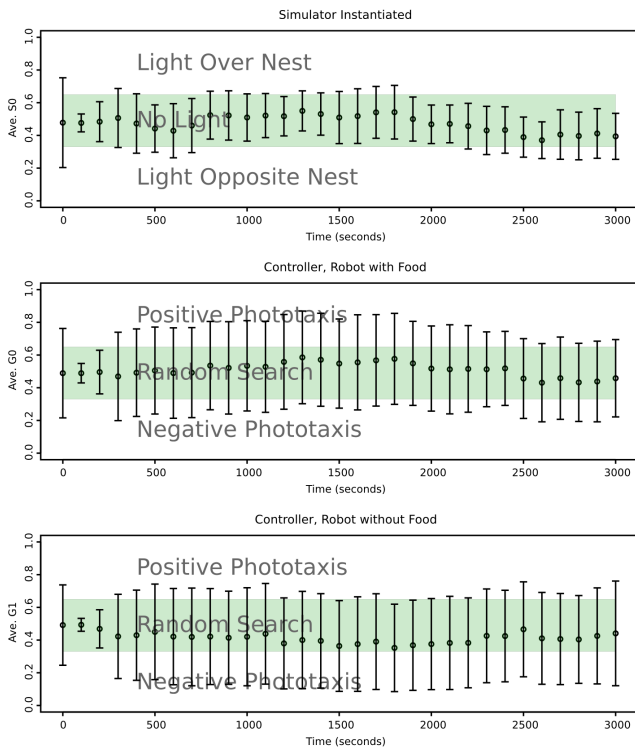
**Fig. 7** *Light Fixed Over Nest*: Three graphs plotting the mean value of the genes  $S_0$ ,  $G_0$  and  $G_1$  over time. The error bars are the standard deviation of the results. The green horizontal bands mark the mapping of the gene value to the controller behaviour or simulator model.

ary approach appears to converge on and exploit the environment circumstance. The evolutionary development in fig.7 is consistent with the superior foraging efficiency shown in fig.5.

## 5.3 Light Source Fixed Opposite Nest

Fig. 8 shows a mean value of  $S_0$  to map to an on-board simulator environment model with no light source for the duration of the experiment, which does not correspond to the actual position of the light source in this experiment case. Using the no light simulator model, the  $G_0$  and  $G_1$  evolve for controllers on average in random search behaviour but with a wide distribution. Despite generally evolving random search behaviour, fig.5 gave a statistical difference in foraging efficiency for this experiment case against the *Random Movement* control. Importantly, there is a light source in this scenario, and it is the wide distribution of evolved controller behaviour mappings that is able to stochastically utilise the light source. In which case, an extra foraging efficiency shown in fig.5 can be explained through the explorative behaviour of the controller genotype evolution, rather than a strong controller transference from



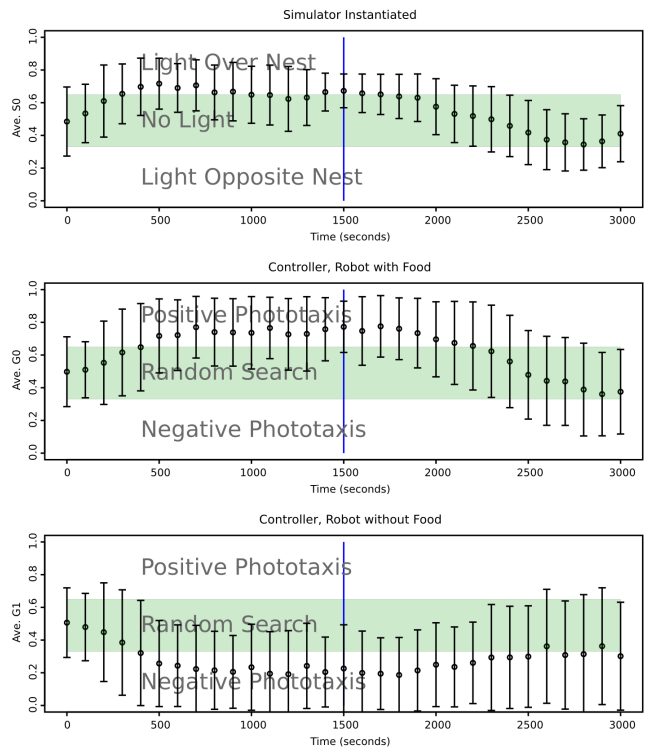


**Fig. 8** *Light Fixed Opposite Nest*: Three graphs plotting the mean value of the genes  $S_0$ ,  $G_0$  and  $G_1$  over time. The error bars are the standard deviation of the results. The green horizontal bands mark the mapping of the gene value to the controller behaviour or simulator model.

the on-board simulator. In which case, the success of a stochastic deviation in controller genotype evolution would not be an exploitation of the on-board simulator, and would not correlate to and inform the distributed evolution of the on-board simulator genotype. This may indicate that there is a problem of precedence between the co-evolution of an on-board simulator and controller, and whether one can provide a reliable fitness indication of the other through our distributed co-evolutionary approach.

#### 5.4 Light Source Over Nest to Light Source Opposite Nest

In this experiment case the light source is initially located over the nest site, and then moved to opposite the nest half way through the experiment. Fig.9 shows the mean value of  $S_0$  correctly evolving the on-board simulator to the Light Over Nest scenario for the first half of the experiment, and the mean values of  $G_0$  and  $G_1$  co-evolve appropriately. This relates to the strong initial foraging efficiency shown in fig.5, and also the strong foraging efficiency for the Light Fixed Over Nest experiment case.

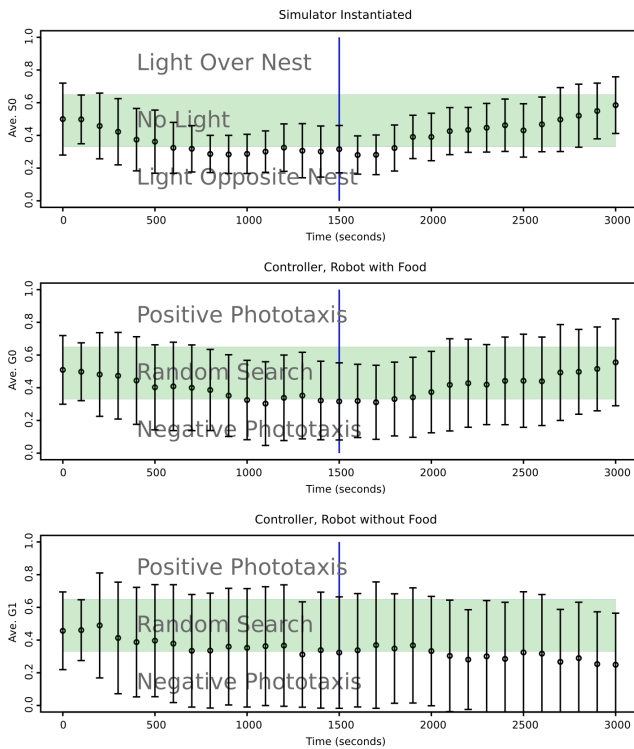


**Fig. 9** *Light Over Nest to Light Opposite Nest*: Three graphs plotting the mean value of the genes  $S_0$ ,  $G_0$  and  $G_1$  over time. The error bars are the standard deviation of the results. The green horizontal bands mark the mappings of the gene value to the controller behaviour or simulator model. The vertical blue line represents the point of light source relocation.

Fig.9 shows a slow adaptation of  $S_0$  after the environment transition point in time, which would cause the evolution of poorly transferring controller solutions and would relate to the sharp drop in foraging efficiency shown in fig.5. Whilst the  $S_0$  mapping of the light scenario does not successfully converge to the corresponding state of the environment, it does alter in value beyond the time of the environmental change. This is as opposed to the co-evolutionary exploitation shown in the results for the *Light Fixed Over Nest* experiment case. Therefore, we can draw that the exploitation in *Light Fixed Over Nest* was related to the stability of the environment, and this transitional *Light Over Nest to Light Opposite Nest* experiment case provokes explorative behaviour from the distributed co-evolutionary approach.

#### 5.5 Light Source Opposite Nest to Over Nest

In this experiment case the light source is initially located over the nest site, and then moved to opposite the nest half way through the experiment. Before the environment transition, the mean value of  $S_0$  moves to-



**Fig. 10** *Light Opposite Nest to Light Over Nest*: Three graphs plotting the mean value of the genes  $S_0$ ,  $G_0$  and  $G_1$  over time. The error bars are the standard deviation of the results. The green horizontal bands mark the mappings of the gene value to the controller behaviour or simulator model. The vertical blue line represents the point of light source relocation.

wards the boundary value of the mapping between a simulator environment model with no light source and a light source opposite the nest site. The exact reason for the adaptation towards the correct simulator environment scenario in this instance and not in the experiment case *Light Fixed Opposite Nest* (fig.8) is not known, and may relate to a potential problem of precedence between the evolution of an on-board simulator and subsequent evolution of controllers, noted earlier. This suggests a larger number of experiment iterations are required to isolate the anomaly in future work. However, despite the apparent convergence of  $S_0$  toward an appropriate environment correspondence,  $G_0$  and  $G_1$  evolve for a wide distribution of controller behaviour mappings. This indicates that the controller evolution did not provide a clear behavioural advantage between random search behaviour and negative phototaxis to inform the simulator evolution.

## 5.6 Discussion

The correspondence between simulation and reality has a consequence on the transferability of controller solutions. We hypothesise that the variation of on-board simulators across many robots can be competitively exploited via the associated real controller fitness of each robot to inform the evolution of an on-board simulator environment model without explicit measurement of the real environment. Our principle result on foraging efficiency across varying experiment cases (fig.5) suggests that our distributed co-evolutionary approach is able to adapt an on-board simulator environment model to the presence of a light source, and consequently improves the evolution of controller solutions tasked with foraging. On closer inspection the results are mixed.

In support of our hypothesis, despite the *No Light* experiment drawing no significant difference in foraging efficiency to the *Random Movement* control, the on-board simulators evolve with a convergence on the correct environment correspondence. If the on-board simulator was entirely disassociated from reality, we would expect to observe a wide distribution of simulator models. The foraging efficiency appears similar to the control due to the inefficient common mode of random movement behaviour in the absence of a light source. However, the real controller performance does inform the on-board simulator evolution.

Furthermore, the experiment cases *Light Fixed Over Nest* and *Light Over Nest to Light Opposite Nest* show a convergence of on-board simulators to the relevant environment model scenario and a higher foraging efficiency. In the case of the light source relocating, the on-board simulator does not successfully re-converge to the relevant environment model scenario, but there is a visible response in evolutionary development. These two experiment cases, having the same initial environment condition, help to demonstrate that the distributed co-evolutionary approach is able to exploit a stable environment circumstance or respond to a changing environment. This supports our hypothesis that .

Compromising our hypothesis, despite a significant improvement in foraging efficiency relative to the control, the *Light Opposite Nest* experiment case failed to evolve an on-board simulator with the relevant environment model scenario. In actuality, the on-board simulator evolved with a convergence to the no light scenario, and evolved a wide controller mapping distribution comparable to the *No Light* experiment case. In which case, the approach was unable to identify and utilise the light source through the real behaviour of the robots. The statistical difference in foraging efficiency from the control was likely gained through the explo-

rative behaviour of the controller evolution to make use of a light source regardless of the on-board simulator.

Furthermore, whilst the *Light Opposite Nest to Light Over Nest* experiment case appears to initially evolve the relevant environment model scenario, the controller mapping evolves with a wide distribution, indicating that there is an ambiguity as to which behaviours transfer well to the real environment when the light is opposite the nest. There is a change in evolutionary development related to the light source relocation, but not enough to reach the much higher foraging efficiency otherwise apparent when the experiments start with the light source over the nest.

Our results indicate that it is possible to couple the distributed evolution of an on-board simulator with the encapsulated evolution of a controllers, providing that the environment gives a strong enough stimulus draw a meaningful real world fitness assessment. When this is not true, the evolutionary development reflects the ambiguity. In our investigation this weakness is when the light is opposite the nest. We hypothesise that when the light is above the nest it acts as a strong attractor, but opposite the nest site the light disperses in all directions providing only a weak repulsive navigational aid.

## 6 Conclusions and Future Work

In this work a background motivation toward an on-line on-board distributed co-evolutionary approach for swarm robotics is described. We propose that on-board simulation and evolutionary computation is an appealing design approach for swarm robotics. We propose that an on-board simulator may aid the currently documented issues facing online on-board distributed evolutionary robotics. We investigate the reality gap, specifically the environmental correspondence of an on-board simulator, by a novel distributed co-evolutionary approach to improve the transference of controllers evolved within an on-board simulator. A novelty of our approach is the the potential to improve transference between simulation and reality without an explicit measurement between the two domains. We are interested in a distributed and implicit selection mechanism of on-board simulators to avoid the need to evaluate multiple on-board simulators per robot, and to leverage the variety of evaluations across many robots against the possibility of uninformative circumstances of a single robot. We hypothesise that the variation of on-board simulator environment models across many robots can be competitively exploited by comparison of the real controller fitness of many robots. We hypothesise that the real controller fitness values across many robots can be

taken as indicative of the varied fitness in environmental correspondence of on-board simulators, and used to inform the distributed evolution an on-board simulator environment model without explicit measurement of the real environment.

Our results demonstrate that our online on-board distributed co-evolutionary approach creates an adaptive relationship between the on-board simulator environment model, the real world behaviour of the robots, and the state of the real environment. The results indicate that our approach is sensitive to whether the real behavioural performance of the robot is able to inform on the state real environment. Our results demonstrate a good co-evolutionary convergence of controllers and on-board simulators when a light source can be used as a navigational attractor to the nest site (*Light Fixed Over Nest*, initially in *Light Over Nest to Light Opposite Nest*). However, if the light source is used as a repulsive navigational aid (*Light Fixed Opposite Nest*, initially in *Light Opposite Nest to Light Over Nest*), a wide distribution of controller genotype mappings evolved, indicating an ambiguity in useful controller behaviours, and may cause a problem of precedence between the co-evolution of an on-board simulator and controller, which will be investigated in the future. The anomaly in our results, where a different evolutionary convergence of the on-board simulator occurs to the same initial environment scenario between the *Light Fixed Opposite Nest* and *Light Opposite Nest to Light Over Nest* experiment cases requires further investigation.

The dependence of our approach on the informative quality of the environment through robot behaviours may be similar to the boot-strapping problem highlighted by Konig et al[10], which links the distributed evolutionary development of robot behaviours to their spatial mobility. In future work we would like to vary the number of robots, as the number of robots constitutes the evolutionary population of on-board simulators, to investigate any gains of parallelism in evaluations towards evolutionary convergence. Logically the number of robots has a relationship to the available space of operation, creating a further variable of spatial density of robots. In our decentralised approach, which necessitates short range communication, we hypothesise that the spatial density and mobility of robots will impact the connectivity of the distributed evolutionary algorithm. In this context, our approach with an on-board simulator bears resemblance to the Island Model spatially structured evolutionary algorithm[22]. Future work would specifically investigate spatial aspects relating to connectivity in distributed evolution on mobile robots as a parallel to the field of spatially structured evolutionary algorithms, and the utility of an on-board

simulator to improve the mechanism of evolutionary selection through virtual evaluations.

## References

1. Bjercknes, J., Winfield, A.: On fault tolerance and scalability of swarm robotic systems. In: A. Martinoli, F. Mondada, N. Correll, G. Mermoud, M. Egerstedt, M.A. Hsieh, L.E. Parker, K. Sty (eds.) *Distributed Autonomous Robotic Systems, Springer Tracts in Advanced Robotics*, vol. 83, pp. 431–444. Springer Berlin Heidelberg (2013)
2. Bongard, J., Zykov, V., Lipson, H.: Resilient machines through continuous self-modeling. *Science* **314**, 1118–1121 (2006)
3. Eiben, A., Haasdijk, E., Bredeche, N.: Embodied, on-line, on-board evolution for autonomous robotics, chap. 7, pp. 361–382. Springer (2010)
4. Eiben, A., Karafotias, G., Haasdijk, E.: Self-adaptive mutation in on-line, on-board evolutionary robotics. In: In Proceedings of the Workshop on Self-Organisation in Pervasive Adaptive Systems (PerAda) at the Fourth IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2010). IEEE Press, Piscataway, NJ (2010)
5. Haasdijk, E., Eiben, A., Karafotias, G.: On-line evolution of robot controllers by an encapsulated evolution strategy. In: In Proceedings of the 2010 IEEE Congress on Evolutionary Computation, Barcelona, Spain, 2010b. IEEE Computational Intelligence Society, IEEE Press (2010)
6. Hayes, A., Martinoli, A., Goodman, R.: Swarm robotic odor localization: Off-line optimization and validation with real robots **21(4)**, 427–441 (2003)
7. Jakobi, N., Husbands, P., Harvey, I.: Noise and the reality gap: The use of simulation in evolutionary robotics. In: *Advances in Artificial Life: Proc. 3rd European Conference on Artificial Life*, pp. 704–720. Springer-Verlag (1995)
8. Kernbach, S., Meister, E., Scholz, O., Humza, R., Liedke, J., Ricotti, L., Jemai, J., Havlik, J., Liu, W.: Evolutionary robotics: The next-generation-platform for on-line and on-board artificial evolution. In: *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, pp. 1079–1086 (2009)
9. Koos, S., Mouret, J.B., Doncieux, S.: The transferability approach: Crossing the reality gap in evolutionary robotics. *Evolutionary Computation, IEEE Transactions on* **17(1)**, 122–145 (2013)
10. Knig, L., Jebens, K., Kernbach, S., Levi, P.: Stability of on-line and on-board evolving of adaptive collective behavior. In: H. Bruyninckx, L. Preucil, M. Kulich (eds.) *European Robotics Symposium 2008, Springer Tracts in Advanced Robotics*, vol. 44, pp. 293–302. Springer Berlin / Heidelberg (2008)
11. Liu, W., Winfield, A.: Open-hardware e-puck linux extension board for experimental swarm robotics research. *Microprocessors and Microsystems* **35(1)**, 60–67 (2011)
12. Liu, W., Winfield, A., Sa, J.: A macroscopic probabilistic model of adaptive foraging in swarm robotics systems (2009)
13. Mataric, M., Cliff, D.: Challenges in evolving controllers for physical robots. *Robotics and Autonomous Systems* **19(1)**, 67–83 (1996)
14. Miglino, O., Lund, H., Nolfi, S.: Evolving mobile robots in simulated and real environments. *ARTIFICIAL LIFE* **2**, 417–434 (1996)
15. Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klapotocz, A., Magnenat, S., Zufferey, J., Floreano, D., Martinoli, A.: The e-puck, a robot designed for education in engineering. In: *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, pp. 59–65 (2009)
16. Nolfi, S., Parisi, D.: Evolving non-trivial behaviors on real robots: An autonomous robot that picks up objects. In: M. Gori, G. Soda (eds.) *Topics in Artificial Intelligence, Lecture Notes in Computer Science*, vol. 992, pp. 243–254. Springer Berlin / Heidelberg (1995)
17. Nolfi, S., Parisi, D.: Learning to adapt to changing environments in evolving neural networks. In: *Adaptive Behavior*, pp. 75–98 (1997)
18. O'Dowd, P., Winfield, A., Studley, M.: Towards accelerated distributed evolution for adaptive behaviours in swarm robotics. In: in *Proc. Towards Autonomous Robotic Systems (TAROS 2010)*, pp. 169–175 (2010)
19. Schlachter, F., Schwarzer, C., Kernbach, S., Michiels, N., Levi, P.: Incremental online evolution and adaptation of neural networks for robot control in dynamic environments. In: *Adaptive 2010, Lisboa, Portugal* (2010)
20. Schut, M., Haasdijk, E., Eiben, A.: What is situated evolution? In: *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, pp. 3277–3284 (2009)
21. Seeley, T.D.: When is self-organization used in biological systems? **202(3)**, 314–318 (2002)
22. Tomassini, M.: *Spatially Structured Evolutionary Algorithms: Artificial Evolution in Space and Time (Natural Computing Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA (2005)
23. Trianni, V.: On the evolution of self-organising behaviours in a swarm of autonomous robots (2006)
24. 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)
25. Zagal, J.C., Ruiz-Del-Solar, J.: Combining simulation and reality in evolutionary robotics. *Journal of Intelligent Robotics Systems* **50(1)**, 19–39 (2007)