# Towards Accelerated Distributed Evolution for Adaptive Behaviours in Swarm Robotics.

Paul O'Dowd, Alan FT Winfield, Matthew Studley

*Abstract*— **The major problem facing swarm robotics is that of design. A recent promising approach is the application of evolutionary algorithms to solve the problem of decomposing group behaviour to that of interacting individual behaviours. This paper presents work conducted so far towards implementing the necessary framework to distribute an evolutionary algorithm across a swarm of robots, augmented with an embedded simulator to provide rapid life-time behavioural adaptability to each robot. The principle has been demonstrated through the distributed evolution of obstacle avoidance behaviour on varying group sizes of physical robots This paper provides preliminary results that validate the principle of an embedded simulator incorporated with distributed evolution, and shows a positive correlation between an increase in robot group size and the rate of distributed evolution. Finally, this paper describes further points to investigate in the presented experiment scenario and potential directions for future research.**

## I. Introduction

> *Swarm robotics is the study of how a large number of relatively simple physically embodied agents can be designed such that a desired collective behaviour emerges from the local interactions among the agents, and between the agents and the environment.* Erol Sahin[16].

Swarm Robotics is an alternative to the approach of engineering a single robot to perform useful functions. Swarm Robotics draws inspiration from the success of swarming biological systems, such as the social insects, that demonstrate the ability to complete tasks as a group that exceed the physical and/or cognitive capability of any singular entity from the swarm [16]. Such a characteristic would be useful where the implementation scale is prohibitive to individual robot complexity (such as nano-scale robotics) or where the swarm would operate in remote or dangerous situations (such as space). Particularly in the latter case, the redundancy of a self-organised swarm system combined with individual flexibility theoretically creates a system capable of graceful degradation.

The major problem facing swarm robotics is that of design. The desired group behaviour must first be decomposed into interactions between individual robots, and the individual robots must have a decomposition of their interaction with the environment into an executable program [22]. A recent promising approach is the application of evolutionary algorithms (EAs) to solve the problem of decomposing group behaviour to that of interacting individual behaviours. Whilst there have been many differing research examples of working swarm robotic systems, the application of EAs provides a consistent methodology for advancement of the field.

The goal of this project is to implement similar evolutionary algorithms (EAs) as used in previous Evolutionary Swarm Robotics research but to distribute the algorithm throughout a swarm of operating robots. The motivation is to allow the evolutionary algorithm to become a feature of the self-organised swarm of robots, to provide life time behavioural adaptation to each of the robots. By placing the EA on a robot, it should become capable of responding to the local situated demands of the robot. Furthermore, distributing the EA capitalises on an inherent abundance of asynchronous computing units within a swarm of robots.

However, in order to provide behavioural adaptability in a rapid and continuous manner this implementation of a distributed evolutionary algorithm utilises a minimal simulation embedded upon each operating robot in which to execute evolutionary evaluations. This combination of a distributed evolutionary algorithm and an embedded simulation is hereafter referred to as an Accelerated Distributed Evolutionary Algorithm (ADEA). An important characteristic of ADEA toward swarm robotics is the ability of an individual robot to simulate itself interacting with other robots on the premise that all robots will interact homogeneously. The isolated development of individual robots necessitates the distribution of the algorithm to facilitate cooperative convergence across the swarm.

The typical constraints of a self-organised swarm of robots combined with the idea of an embedded simulator to accelerate distributed evolution, present several research questions.

- Can a distributed evolutionary algorithm provide adaptability and scalability to the evolution of self-organised behaviours?
- Can a swarm of robots running independent simulations to evolve controllers for group behaviour converge in reality on a self-organised solution?
- Will spatial and temporal distribution throughout the swarm of robots have significant effect upon the evolutionary development process of individual robots?
- Concerning the fitness function to guide artificial evolution; Is it possible to evolve self-organising behaviour from the scope of individual performance?
- Can the temporal credit assignment problem between

Paul O'Dowd, Alan FT Winfield and Matthew Studley are with the Bristol Robotics Laboratory, University of the West of England, Coldharbour Lane, Bristol, UK (email: paul.odowd@brl.ac.uk, alan.winfield@uwe.ac.uk, matthew2.studley@uwe.ac.uk ).

real world performance and simulated performance be bridged or bypassed successfully?

This paper presents work conducted so far towards implementing the necessary framework to investigate these questions. To date, the principle has been demonstrated through the distributed evolution of obstacle avoidance behaviour augmented with an embedded simulator on physical robots. Section I proceeds to provide background context and related work in biology, swarm robotics and evolutionary robotics. Section II details the experiment methodology used for the results presented in section III. Section IV draws conclusions from these results, providing a scope to discuss the future potential of research.

### A. Biological Plausibility

Swarm robotics has a particularly strong influence from biological systems where self-organisation creates the observable characteristics of scalability, flexibility and robustness to disturbances [16]. To help position this work along a scale of biological plausibility and engineering expectations, the following brief section considers attributes of natural swarm systems to justify subsequent technological implementation decisions.

Self-organisation is a prevalent phenomenon in biology whereby organisms attenuate their behaviour on the basis of local information, resulting in coordination [3]. Local information is context dependent, such as chemical secretions from neighbouring cells, the deposition of material from other worker ants or the trajectory of nearby flocking birds. The locality of information refers to the perceptual abilities of the individual, and the absence of an absolute perception of the global state of the whole swarm. Therefore, a system of many subunits using only local information can coordinate without a centralised control mechanism, scaling to vast population sizes. Furthermore, the subunits are adaptable to their unique situations, demonstrating a robust response to disturbances in their environment and flexibility toward new challenges.

Whilst it may be possible to coordinate large numbers through centralised control (such as multi-cellular organisms with central nervous systems), Thomas Seeley [18] states that:

> *...a biological system will be self-organized when it possesses a large number of subunits, and these subunits lack either the communicational abilities or the computational abilities, or both, that are needed to implement centralized control.*

When considering examples like the social insects, it is clear that they lack a biological mechanism to centrally coordinate vast numbers of autonomous subunits over great distances. Individuals seem selflessly predisposed to work toward the benefit of the colony. In such a system, subunits must independently ascertain their role in tasks based on their local perception of their environment. Stigmergy is a special case where subunits are influenced by the product of anothers behaviour upon the environment. Whilst stigmergy is a way for a single subunit to influence many other subunits, the perception of the affected subunit remains local.

It is also important to note that the cognitive ability of subunits need not necessarily be constrained in a self-organised system [18]. Indeed, the decision making and communicative abilities of the honey bee are astounding (for example, see [19]). More importantly, the constraint to perceive local information creates a robust, scalable and flexible decentralised system, which can be mistaken for a requirement of individual minimalism.

With these realisations in mind, we can isolate the principles that create a self-organised system and allow ourselves the freedom to utilise appropriate levels of technology. As previously stated, the state of available technology is primarily governed by the implementation scale of the system. The available hardware to this research, and therefore the types of application domains, are definitively in the macro scale. Whilst an embedded simulator may not be biologically plausible, it should not necessarily inhibit the decentralisation of a self-organised system. Furthermore, the implementation of distributed evolutionary algorithm, so long as transactions between robots occur locally, should not compromise the benefits of flexibility, scalability and robustness.

### B. The Progression of Swarm Robotics

Early experiments by Melhuish et al [11] demonstrate the ability to effect group behaviour solely through direct interaction with the environment (stigmergy). These robots were exceptionally simple, but were able to sort objects into clusters by colour, by each obeying simple rules without centralised control. These experiments concentrated on how much could be achieved with minimally complex robots, referred to as Minimalist Collective Robotics. However it is important to note that stigmergy is not limited to minimalist robotics, as demonstrated by the complex team task investigated by Ijspeert et al [5] in the multi robot stick-pulling experiments. A uniting feature of these experiments is the use of stigmergy as the sole mechanism for coordination.

The emphasis on minimalist implementation remains a current research interest [24], but the advancement of technology has allowed for the development of higher numbers of more complex and reliable robots [20]. Whilst stigmergy remains a vital mechanism [16], the dissimilarities between natural and artificial systems has prompted the use of 'less-natural' mechanisms to facilitate self-organised group behaviour with robots. For example, Nouyan and Dorigo utilise a colour based signalling mechanism to facilitate chain formations with the SWARM-BOTS [14], and Jones and Mataric investigated colour based signalling for division of labour [8]. Furthermore, Ijspeert et al [5] found that stigmergic collaboration between robots was improved with a simple signalling mechanism. These recent advancements hint that minimalism and stigmergy are not in fact the only mechanisms for self-organised swarms of robots, so long as decentralisation is facilitated through a constraint on local interactions.

A recent trend is the movement away from purely reactive robotic controllers. In early minimalist experiments with stigmergy the behaviour of a robot was usually a rule set that tightly coupled sensory stimulus with immediate action [11], meaning that the robots were essentially working asynchronously with little direct robot-to-robot interaction. The development of experiments less reliant on stigmergy has lead to robots equipped with communication [14], learning [15], mapping and localisation [17]. This may be to provide information to the robots that would otherwise have been gained through stigmergic interaction.

The movement away from reactive robots and stigmergy has increased the difficulty of designing the systems, as the decomposition of group behaviour must further take into account the cognitive abilities of robots and how this will influence the interactions of the swarm [20]. Because interactions are specifically between robots (not just through the environment), spatial and temporal distribution have a greater effect on the dynamics of interactions. The traditional Swarm Intelligence approach to model the problem to extract rules of interaction may become infeasible if the dynamics of interaction continue to increase in complexity. This issue has been termed the design problem by [22].

This brief review of the field of Swarm Robotics helps to highlight the growing complexity in the field. A recent advance has been the application of Evolutionary Algorithms to bypass the design problem of group behaviour decomposition. The iterative trial-and-evaluate process of evolutionary algorithms allows for the development and selection of individual behaviours that optimise for the specifics of an embodied robot and the dynamics of interaction [22]. An important characteristic in this related work is the ability to assess the group level behaviour from the global perspective of interacting individual controllers which may not be feasible in a decentralised and distributed implementation. Relating to this work, this issue remains a topic open for discussion (Section IV).

### C. Evolutionary Robotics

Evolutionary Robotics is the application of Evolutionary Algorithms (EAs) for the generation of controller solutions or physical morphology (or both) of autonomous robots [13]. Instead of a human decomposing the task into subtasks to ease the generation of a robotic solution, EAs work by an iterative process of evaluation, searching for a solution against a selective criterion. An EA is therefore an incremental approach, synthesising solutions from the bottom-up. In this way EAs are attractive because they abstract the design problem by a process of trial-and-error.

When EAs are applied to robotics the fitness of evolved solutions is either assessed on a real robot, or in a simulation. In the first case, there is not a requirement to accurately model the robot and the environment as this is experienced directly by the robot, but development times can be prohibitively long. In the second case, using a simulation tool greatly improves the speed of synthesis, but there remains a debate as to the viability of sufficiently representing ever more complex problems.

This obvious conflict and the finer details [10] have been the subject of research on many occasions. Watson [23] successfully distributed an embodied evolutionary algorithm across multiple robots to take advantage of the inherent parallism in the system and speed up the evolutionary process. Jackobi provides evidence that it is possible to utilise minimal simulations to synthesis complex transferable behaviours, provided that sufficient noise is injected upon features that can not be modelled accurately [7][6]. There are also many examples of success utilising the simulate-and-transfer methodology [9], including the application of the technique to Swarm Robotics [22].

Recently, the work of Bongard [2] has sought to unify embodiment and simulation. By using adaptable self-models, a quadrupedal robot is able to develop faster through the utilisation of a simulation to develop walking gait patterns. Importantly, the adaptable self-model (and consequently the simulation) are grounded by periods of explorative behaviour in the real world, representing a closed and complimentary system. Further research proved that distributing the process using multiple robots accelerated the self-modelling [1].

The work in this paper differs from previous work in that the simulation is intended to model interactions between robots towards the development of cooperative self-organised behaviours. Specifically, this research will address issues concerning the effects of distributing the algorithm and the means to assess cooperative utility from the individual perspective. So far, this work does not yet incorporate a mechanism to integrate feedback from the real world operation but it is a definitive criterion to ensure the transferability of solutions developed in simulation.

## II. Experimental Methodology

The Accelerated Distributed Evolutionary Algorithm framework has three core components: the simulation tool, a genetic algorithm, and a communication mechanism to distribute solutions to neighbouring robots. In this application, the genetic algorithm searches for weightings for a simplistic feed-forward artificial neural network controller to provide obstacle avoidance behaviour to a simulated robot. The best solution discerned from simulation is transferred to the real robot for operation in the real world. Currently, the performance of the robot in the real world does not provide any measure of utility back into the system but does affect the spatial distribution of the real robots. In these experiments, the real robots operate within an enclosed circular arena with a diameter of 600mm free from obstacles. In these experiments, sets of 2, 4, 6 and 8 robots were initialised simultaneously and allowed to evolve over a period of 1 hour. The results are presented in Section III. Further experiment details are provided throughout this section.

### A. Robots

The experiments presented in this paper are performed using the e-puck robots (see figure 1), which are two wheel differential robots featuring a variety of sensors measuring 70mm in diameter [12]. For these experiments, only 6 infra-red proximity sensors positioned on the forward facing portion of the robot are used for proximity detection. The robots also use the Linux Board Extension (LBE) that has been designed at the Bristol Robotics Laboratory by Dr. W. Liu to provide compuational parallelism. The LBE provides the Linux based operating system and superior computing power, used to execute the evolutionary algorithms by way of simulation tool. The LBE is capable of exchanging information with the native e-puck hardware over an SPI bus connection. The SPI bus connection provides a method to synchronise the operation of the e-puck with the simulation tool (see Section II-B ), providing discrete operation in 40ms time increments. A final feature used is the Infra-red (IR) Communication library authored by Alexandre Campo. IR Communcation is used to distribute genetic information between robots (see Section II-D) and has the relatively short range of transmission of two body lengths of the e-pucks, maintaining locality in communication.

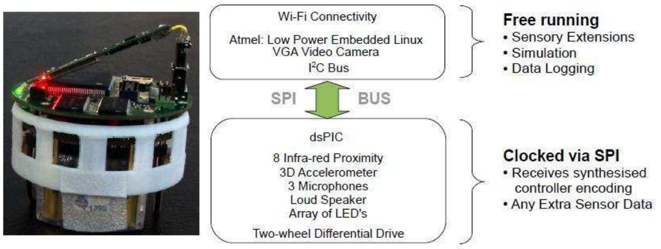

Fig. 1. Photograph of an e-puck robot fitted with the Linux Board Extension and a Wi-Fi card. The annotations to the right of the image illustrate the capabilities of the configuration.

### B. Simulation

A minimal simulation has been written in the C programming language in order to execute on board the LBE as quickly as possible. The minimal simulation is based on the assumption that e-pucks are light weight and have relatively high motor torque, negating the need for accurate modelling of physical properties such as mass, inertia and momentum. The simulation of robots and objects in the environment are represented on a 2d plane as single points with a radius (in effect, circles), reducing the representation to simple trigonometry. The simulated robots are moved on the basis of the two-wheel differential kinematics equations from [21] and the specifics of the e-puck robots. The infra-red sensors are represented with a triangular field of view. Raw sensor data has been captured from the real robot sensors to provide a look-up table emulation of the physical sensors, to which uniform noise is added.

Whilst not strictly necessary in these experiments, the simulation instantiates four robots within the same simulated environment. This feature is towards future work which will specifically investigate interactions between robots, but for the time being facilitates parallel evaluation in a more dynamic environment. The simulation is advanced in 40ms time intervals.

### C. Artificial Neural Network

The robotics controller is a discrete time feed-forward artificial neural network. The network is simplistic, not using any thresholding or transfer functions, relying on weighted connections between nodes $w_{ij}$. Two output nodes, $y_1$ and $y_2$, are fully connected to the two hidden nodes, $h_1$ and $h_2$. This connectivity creates four weight values in this layer.

$$y_i = \sum_j w_{ij} h_j \tag{1}$$

Hidden node $h_1$ takes weighted input from the three infra-red proximity sensors on the left side of the robot body ($x_{1..3}$), whilst hidden node $h_2$ takes weighted input from the three proximity sensors on the right side of the robot body ($x_{4..6}$). This connectivity creates 6 weight values in this layer.

$$h_1 = \sum_{j=1}^{3} w_{1j} x_j \tag{2}$$

$$h_2 = \sum_{j=4}^{6} w_{2j} x_j \tag{3}$$

This network is therefore shaped by the 10 weight values of the connections between nodes, in the range [-1.0,+1.0]. The output nodes are used as the control signal for the two motor outputs, limited to a value in the range [-100,+100] percent of motor speed. The output nodes also each receive a fixed bias value representing +30% motor speed to force forward motion of the robot.

### D. Accelerated Distributed Evolutionary Algorithm

The ADEA in these experiments uses a steady-state genetic algorithm (GA) to search for solutions to the obstacle avoidance problem. The GA population genomes are each composed of 10 genes, representing the connection weights through the ANN (see section II-C). Selection for reproduction is rank based and elitist, where 50% of the population is used to overwrite the lower ranking percentage. During the overwrite process, each gene of the child genome is subjected to probabilistic mutation in the range of [-1.0,+1.0] with a 14% chance of occurrence. The GA features mutation as the only mechanism to introduce variation.

The GA executes exclusively onboard the LBE of each robot in the experiment. The GA maintains a population of 10 solutions evaluating each genome sequentially. An evaluation in simulation constitutes the cloning of the genome onto the 4 simulated robots which then operate concurrently in the same simulated environment (see section II-B). Upon the initialisation of each new evaluation, the position and orientation of the robots is randomly determined.

The simulated evaluation period is 1500 steps, the equivalent to 60 seconds of real time in simulation, although evaluation is terminated prematurely if any of the four robots collides with either another robot or an obstruction. Each of the four robots in simulation is assessed for utility, and at the end of the evaluation the lowest score is taken to represent the utility of the genome. The fitness metric is as originally used by [4], given by:

$$F = V \cdot (1 - \sqrt{\Delta v}) \cdot (1 - p)$$
$$0 \leq V \leq 1, \quad 0 \leq \Delta v \leq 1, \quad 0 \leq p \leq 1$$

where $V$ is the average absolute wheel speed between both motors, $\Delta V$ is the average difference in motor speeds, and $p$ is the average activation value of the most active proximity sensor. The combination of these elements promote higher motor speed, consistent directionality and to avoid obstructions occurring on the proximity sensors.

At the end of a GA generation, defined by each genome of the population being evaluated once, the solution from simulated evaluation with the highest utility is transfered and instantiated on to the e-puck robot. When this has occurred once, the robot is henceforth able to collect genomic information from neighbouring e-pucks. Each e-puck therefore continually transmits the currently instantiated solution via IR Communication in all directions. A single transmission is composed of the gene value and the corresponding gene position index (along the length of the complete genome). Therefore, an e-puck transmits each gene sequentially.

An e-puck maintains only one representation of collected genetic information in which the latest received genes overwrite any previous data. It is important to note therefore, that the genetic information collected may be composed from several neighbouring e-puck broadcasts. When a subsequent update occurs from the LBE with a new simulated solution to instantiate on the e-puck, the collected genetic information is uploaded into the GA onboard the LBE, directly replacing the solution in the population with the current lowest utility. This integrates the collected genetic information into the simulation tool to be evaluated in the next generation of genomes.

## III. Results

In all experiments the robots are capable of finding a correct set of weights for the ANN topology to provide consistent obstacle avoidance movement through the arena, avoiding the arena walls and other robots. An interesting behavioural observation in the earlier stages of these experiments was the oscillatory behaviour in response to obstacles in the experiment due to incorrect weight assignments, corresponding to poor sensory-motor mapping. The functionality of the embedded simulator allows for human-observable adaptation of the ANN through the increasing consistency in movement, which is also represented by the improving fitness function values in figure 2.

Figure 2 represents the average fitness progression over 30 generations from 10 experiment runs for each of the sized groups of robots (2, 4, 6 & 8). From each experiment run using a group of robots, one participating robot was selected at random to represent the group. The results show that there is a distinction between group sizes and the rate of fitness progression. In all cases, the rate of fitness progression increased with group size, which can be attributed to the distribution of the search process across the number of robots participating.

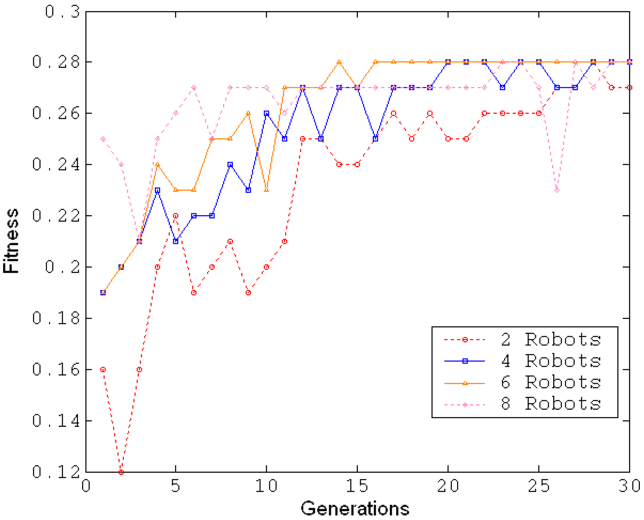

Fig. 2. The average fitness progression for each sized group of robots taken across 10 experiments for each group size.

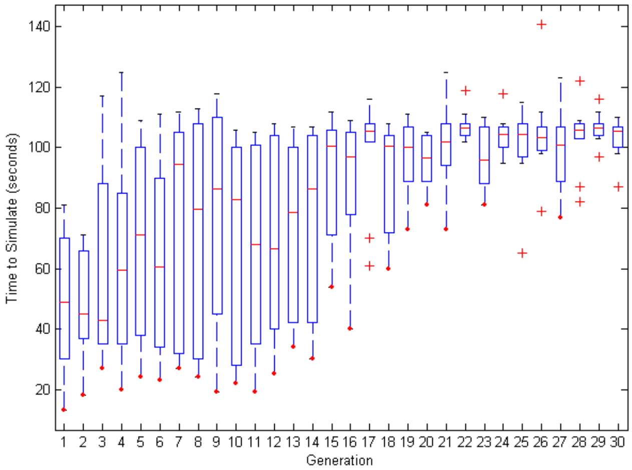

Fig. 3. The time taken to simulate each generation through evolutionary development. The results are based on the 10 experiments conducted for the group size of 2 robots. The average time to evolve 30 generations is 41 minutes.

A secondary interesting result is the time taken to conduct the evolutionary evaluation using the embedded simulation tool. Figure 3 demonstrates that as evolution progresses the time taken to simulate takes longer, with variance decreasing with respect to the progression of evolutionary generations. This is occurs for two reasons. First, the evaluation of initial solutions are aborted prematurely when any of the four simulated robots collide with an obstruction. Given that the initial solutions are randomised, this occurs frequently. The reduction of variance is also due to the population of solutions gaining a convergence on more successful solutions. Towards the later generations, simulation time stabilises at approximately 100 seconds per generation.

## IV. Conclusion & Future Work

This paper presented a methodology for accelerating a distributed and embedded evolutionary algorithm. The principle finding from these experiments is that accelerated distributed evolution is possible in real time, implemented entirely using physical robots. Furthermore, the experiment also demonstrates that the ADEA is sufficient to provide life-time adaptable behaviour to robots. The development of obstacle avoidance occurred on average within 41 minutes which is within the operable life-time of the e-puck robot. In the worst approximate case of 100 real-time seconds per generational evaluation, four simulated robots interact for each GA population member, representing a total of 600 seconds of simulated time. An interesting consequence of this methodology is that the embedded simulation tool allows for the exploration of potentially unstable behaviours whilst the robot can continue to operate with the last best known stable behaviour.

However, the scenario constructed for this proof of concept is very simplistic and requires further evaluation. In this experiment, the parameters of the genetic algorithm were arbitrarily selected and should be varied to test the sensitivity of the results. It would be interesting to conduct an analysis of the efficiency of movement by the physical robots through the course of experiment runs. Through observation, oscillatory behaviour is reduced as the experiment transpires and empirical proof could provide a strong point of further analysis. Furthermore, communication transactions between physical robots is currently not traceable. If transactions could be monitored, it might be possible to form an analysis on the progression of evolution based on the spatial distribution of robots and the frequency that meaningful transactions occur. Currently, the sources and quality of transactions between robots is unknown. As well as the specific transactions, it should be possible to trace the state of genetic representation between physical robots to analyse points of evolutionary convergence and how this may relate to spatial and temporal distribution of physical robots. These elements of future work acknowledge that there is currently a lack of implementation in the physical domain of the algorithm.

A major current deficiency is the lack of utility feedback from the physical activity of the robots. This means that the simulation tool is not grounded, and the actual performance of simulated solutions in the real world is unknown to the robot. The exact method to incorporate physical performance feedback presents an interesting problem. If the simulated evolution is paused for periods of evaluation in the real world, the acceleration gained would be lost and the behaviour of the robot would be punctuated. If simulated evolution occurs continuously, the rate of generational advancement may outstrip the evaluation conducted in the real world, causing any measure of utility gained to be out of date.

The experiment presented in this paper is just a starting point for further research. Beyond a more rigorous analysis of the current experiment scenario, the intention is to apply pressure to the algorithm to ascertain whether it holds benefits towards adaptable behaviour for the swarm as a whole, rather than just the individual. An example scenario would be foraging, where the energy efficiency of the swarm could be analysed against a variable distribution of collectable food. A definitive direction of future research is to increase the state space of the robots, such as higher sensor modality and more complex tasks, and whether or not a minimal simulation remains a viable solution for life-time adaptable behaviour.

A distributed algorithm also has interesting avenues of research towards the flow of information across the swarm population. Further research could investigate spatial segregation and the possibility of divergence within the same swarm. This also alludes to the potential for foreign members of the swarm (either those separated for large amounts of time, or newly introduced) to utilise the distributed nature of the algorithm to accelerate their own behavioural synthesis toward the current state of the collective.

Finally, there are interesting specifics of the embedded simulation tool developed for this research that offers wider potential. The current ability of the embedded simulator allows multiple simulated robots to interact and for this to be observed from any perspective (a global perspective, or local to any one of the simulated robots) by the physical hosting robot. This could provide a mechanism by which observed behaviour of other robots in the real world could be reformulated within the embedded simulation - in effect a brute force search - to find a best fit model of what occurred. The ability to change the simulated environment and view multiple perspectives may allow a robot to translate observed behaviour into motor actuation for itself, as a form of imitation.

## References

[1] J Bongard. Exploiting multiple robots to accelerate self-modeling. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, pages 214–221, New York, NY, USA, 2007. ACM.

[2] J. Bongard, V. Zykov, and H. Lipson. Resilient machines through continuous self-modeling. *Science*, 314:1118–1121, 2006.

[3] Scott Camazine, Nigel R. Franks, James Sneyd, Eric Bonabeau, Jean-Louis Deneubourg, and Guy Theraula. *Self-Organization in Biological Systems*. Princeton University Press, Princeton, NJ, USA, 2001.

[4] Dario Floreano and Francesco Mondada. Evolution of homing navigation in a real mobile robot. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 26:396–407, 1996.

[5] Aukejan Ijspeert, Alcherio Martinoli, Aude Billard, and Lucamaria Gambardella. Collaboration through the exploitation of local interactions in autonomous collective robotics: The stick pulling experiment. *Autonomous Robots*, 11(2):149–171, 2001.

[6] Nick Jakobi. Half-baked, ad-hoc and noisy: Minimal simulations for evolutionary robotics. In *Fourth European Conference on Artificial Life*, pages 348–357. MIT Press, 1993.

[7] Nick Jakobi, Phil Husbands, and Inman Harvey. Noise and the reality gap: The use of simulation in evolutionary robotics. In *Advances in Artificial Life: Proc. 3rd European Conference on Artificial Life*, pages 704–720. Springer-Verlag, 1995.

[8] Chris Jones and Maja J Mataric. Adaptive division of labor in large-scale minimalist multi-robot systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 27–31, 2003.

[9] Ian Macinnes and Ezequiel Di Paolo. Crawling out of the simulation: Evolving real robot morphologies using cheap reusable modules. In *In Artificial Life IX: Proc. Ninth Intl. Conf. on the Simulation and Synthesis of Life*, pages 94–99. MIT Press, 2004.

[10] Maja Mataric and Dave Cliff. Challenges in evolving controllers for physical robots. *Robots and Autonomous Systems, Special Issue on Evolutional Robotics*, pages 67–83, 1996.

[11] Chris Melhuish, Owen Holland, and Steve Hoddell. Collective sorting and segregation in robots with minimal sensing. In *Proceedings of the fifth international conference on simulation of adaptive behavior on From animals to animats 5*, pages 465–470, Cambridge, MA, USA, 1998. MIT Press.

[12] Bonani M. Raemy X. Pugh J. Cianci C. Klaptocz A. Magnenat S. Zufferey J.-C. Floreano D. Mondada, F. and A. Martinoli. The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, pages 59–65, 2009.

[13] S. Nolfi and D. Floreano. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. Cambridge, MA: MIT Press/Bradford Books, 2000.

[14] S. Nouyan and M Dorigo. Chain based path formation in swarms of robots. In *Proceedings of ANTS 2006*, pages 120–131, Berlin, Germany, 2006. Springer Verlag.

[15] Lynne E. Parker, Claude Touzet, and David Jung. Learning and adaptation in multi-robot teams. In *Proc. 18th Symp. on Energy Engineering Sciences*, pages 177–185, 2000.

[16] Erol Sahin. Swarm robotics: From sources of inspiration to domains of application. *Swarm Robotics*, pages 10–20, 2005.

[17] M. Schwager, J. McLurkin, J. J. E. Slotine, and D. Rus. From theory to practice: Distributed coverage control experiments with groups of robots. In *Proceedings of International Symposium on Experimental Robotics*, Athens, Greece, 2008.

[18] Thomas D. Seeley. When is self-organization used in biological systems? *Biological Bulletin*, 202(3):314–318, June 2002.

[19] Thomas D. Seeley. Consensus building during nest-site selection in honey bee swarms: the expiration of dissent. *Behavioral Ecology and Sociobiology*, 53(6):417–424, May 2003.

[20] Amanda J. C. Sharkey. Swarm robotics and minimalism. *Connect. Sci*, 19(3):245–260, 2007.

[21] Roland Siegwart and Illah R. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. Bradford Book, 2004.

[22] Vito Trianni. *On the Evolution of Self-Organising Behaviours in a Swarm of Autonomous Robots*. PhD thesis, Faculty of Applied Sciences, Universite Libre de Bruxelles, Brussels, Belgium, 2006.

[23] R. Watson, S. Ficici, and J. Pollack. Embodied evolution: Distributing an evolutionary algorithm in a population of robots. *Robotics and Autonomous Systems*, 39(1):1–18, 2002.

[24] Matt Wilson, Chris Melhuish, Ana B. Sendova-Franks, and Samuel Scholes. Algorithms for building annular structures with minimalist robots inspired by brood sorting in ant colonies. *Autonomous Robots*, 17(2-3):115–136, 2004.