An Evolutionary Approach to Formation Control with Mobile Robots

Jane Holland, Josephine Griffith and Colm O'Riordan
College of Engineering and Informatics, National University of Ireland, Galway, Ireland

Keywords: Evolutionary Robotics, Swarm Robotics, Self-organisation, Collective Behaviours, Kilobots.

Abstract: The field of swarm robotics studies multi-robot systems, emphasising decentralised and self-organising behaviours that deal with limited individual abilities, local sensing and local communication. A robotic system needs to be flexible to environmental changes, robust to failure and scalable to large groups. These desired features can be achieved through collective behaviours such as aggregation, synchronisation, coordination and exploration. We aim to analyse these emerging behaviours by applying an evolutionary approach to a specific robotic system, called the Kilobot, in order to learn behaviours. If successful, not only would the cost and computation time for evolutionary computation in mobile robotics decrease, but the reality-gap could also narrow.

1 INTRODUCTION

Swarm robotics is a developing field within collective robotics, which focuses on groups of robots that interact and cooperate with one another in order to reach a common goal. This form of robotics can be compared to a class of natural systems such as social insects – ants, bees and termites – that can accomplish intricate tasks by means of interaction. In a decentralised, problem-solving system such as this, insects have very limited capabilities. However, by working together as a group, the overall performance can be improved through self-organisation: a process whereby a system transitions from a disordered state to an ordered state by exploiting local interactions between robots and between robots and their environment (Dorigo et al., 2004; Trianni and Dorigo, 2006; Baldassarre et al., 2007).

Formation control is an emergent collective behaviour of self-organisation inspired by nature, which traditionally demonstrates a 'follow-the-leader' approach in a swarm of robots through coordination, synchronisation and communication. These abilities allow robots to dynamically interact with one another in order to organise into a complex system so as to work together effectively. Furthermore, these abilities can result in an organised set of actions, such as choosing a common sense of direction when in a group formation. This is essential for a team of robots to work as a whole entity and it can also establish a basic building block for the design of more complex behavioural strategies (Trianni et al., 2008). In order for the coordination of behaviours to be successful, robots must be able to define communication strategies and protocols among the individuals of the group. Here, simple forms of communication can be enough to accomplish the coordination of group activities and can also be easily scaled up with the number of individuals involved.

The motivation of the work described here is to use evolutionary techniques to create parallels with the biodiversity seen in natural systems; that is, to automate self-organised, collective behaviours using decentralised control and local communication. Evolutionary algorithms are noted for their simplicity; the Darwinian theory of survival of the fittest is a focal example: only the fitter individuals of a population are allowed to reproduce. In evolutionary robotics, a population of genotypes is initialised and then each genotype is encoded and mapped into each robot controller. The process of evolution then evaluates the performance of the robots as individuals or as teams. The robots that perform above average are allowed to reproduce and their genetic material can be altered by means of mutation and recombination. This method is repeated several times until one or more controllers are found that meets the requirements of an evaluation, or fitness function. The efficiency of the algorithm is dependent on the variance and selection applied to the population and so, the algorithm can be tailored to many different approaches and multi-modal problems. Not only are evolutionary algorithms relatively simple, but they can produce solutions that are robust and flexible to change.

Holland, J., Griffith, J. and O'Riordan, C.
An Evolutionary Approach to Formation Control with Mobile Robots.
DOI: 10.5220/0006068602250230
In *Proceedings of the 8th International Joint Conference on Computational Intelligence (IJCCI 2016) - Volume 1: ECTA*, pages 225-230
ISBN: 978-989-758-201-1

These abilities to adapt are crucial for practical problem solving as they significantly decrease the cost, time required and can be evolved on the fly if unexpected anomalies occur.

This position paper aims to outline the advantages of applying evolutionary computation to swarms of potentially heterogeneous, mobile robots where each robot has limited functionality. Furthermore, a genetic algorithm is proposed which we believe is capable of evolving simple swarms of mobile robots to carry out the collective behaviour of formation control. We focus on formation control because this behaviour not only demonstrates self-organisation, coordination and synchronisation, but it is also a key component for more complex behavioural strategies. Although evolutionary computation has been practised on Khepera robots (Mondada et al., 1994) and S-Bots (Mondada et al., 2002), there has been no research carried out on robots with a high level of hardware and software restraints.

The outline of the paper is as follows: initially, we outline our motivations and challenges in section 2; these include system design, cost, time and the reality-gap. Next, an overview is given of the robots which will be used in the work (Kilobots). In section 4, we discuss related work in the areas of self-assembly and decision making, both of which demonstrate the process of self-organisation. In section 5, we put forward a methodology, describing our formation control experiment in detail, making reference to our chosen evolutionary algorithm and fitness function evaluations. Lastly, conclusions and future work are discussed in section 6.

2 MOTIVATIONS AND CHALLENGES

Designing a control system for robots is a challenging and complex task as the system needs to be decomposed into two phases: the behaviours of the individual and the behaviour of the system. The global behaviour is a result of dynamical interactions among its components, be it interactions between individuals or interactions between individuals and the environment (Trianni and Dorigo, 2006). As these dynamical interactions are hard to foresee, it is a good idea to use an evolutionary technique to avoid decomposition at both levels. Such an approach relies on the evaluation of the system as a whole, particularly on the emergence of the desired global behaviour starting from the definition of the individual behaviours (Trianni et al., 2008). By using this approach, the evolutionary process eliminates unsought behaviours and selects only the desirable behaviours based on an evaluation function.

Other challenges that need to be taken into account are the costs and time required. For instance, although the evolutionary approach bypasses decomposition, artificial evolution can take a long time to compute on a physical robot. Furthermore, if robots were to be damaged during this evolution period, costs would increase as hardware would need to be replaced. It is for these reasons that simple, mobile robots need to be used. As Kilobots have a low cost price of around $14 for parts and only take 5 minutes to assemble, they can be easily produced in large numbers (Rubenstein et al., 2012). The evolution of their controllers can also be done through simulation and then the most successful individuals can be run on the physical robots in order to speed up the process. However, evolving robot behaviours through the use of simulation can present some other problems such as a reality-gap: programs may work well on simulated robots, but they can fail on real robots due to the different actuation and sensing abilities as it can be very difficult to simulate the actual dynamics of the real world (Brooks, 1992).

3 KILOBOT

The Kilobot (Rubenstein et al., 2012), shown in Figure 1, is a low-cost robot designed to make testing collective algorithms on a large number of robots easily manageable. The Kilobot has an Atmega328 processor controller, running at 8MHz with 32K of memory. The controller has two main functions; it is used to run a user-defined program as well as operating as an interface between electronics such as motors, power circuitry and LEDs. It can regulate the speed of the vibration motors by using two pulse width modulation channels and can measure the intensity of the incoming infra-red signals through 10-bit analog-to-digital converters.

This robot does not have traditional characteristics as it uses vibration motors for movement, and reflective infra-red light and distance sensing in order to communicate with other robots. The Kilobot uses an infra-red LED transmitter and receiver that allows the robot to receive messages from every direction. When the transmitter is in use, a neighbouring robot can receive light emitted by the transmitting robot once it has been reflected off the surface on which the Kilobot has been placed. The locomotion of these motors prevent against travelling long distances by providing noisy locomotion without position feedback. However, by using measured dis-

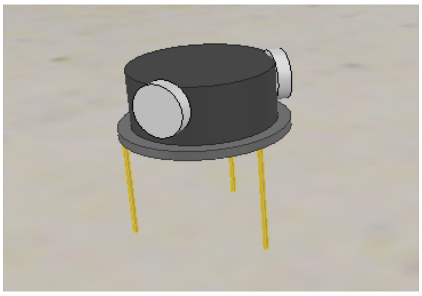


Figure 1: Graphical visualisation of the Kilobot robot in V-Rep simulation software.

tances between neighbours as feedback, the robot's movement can be corrected, as well as being used to promote the use of collective behaviours (Rubenstein et al., 2012).

As previously mentioned, the limited range and lack of bearing systems in the Kilobot make coordinated navigation difficult to implement. Kilobots can only perceive the distance between each other, but not the direction their neighbours are travelling in. To combat this, robots need to continuously communicate and measure the distances between themselves and their neighbours in order to calculate their neighbour's pose, that is their orientation and position.

4 RELATED WORK

Much of the previous work in this area has dealt with collective algorithms that demonstrate basic functionality in Kilobots, such as the ability to move within an environment and communicate with neighbouring robots (Rubenstein et al., 2012). More recently, Rubenstein et al. (2014) have developed a self-assembly algorithm which demonstrates edge following, gradient formation, localisation and collective transport. The aim of this work was to create artificial swarms with the capabilities of natural ones. Furthermore, decision making within a self-organised system has also been studied by Valentini et al. (2015), which examines the trade-off between speed and accuracy when making collective decisions as a swarm. The decision strategy used was successfully implemented on 100 robots, proving feasibility in large swarms and robustness to robot failures.

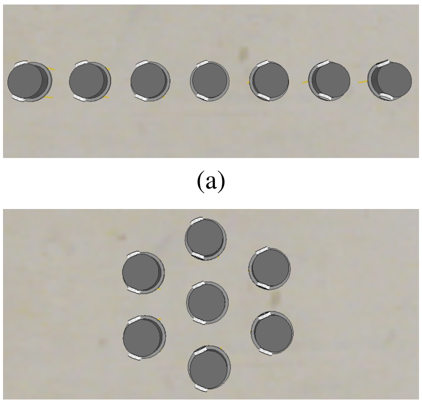
This type of research tends to gravitate towards more physics and mathematical based models, rather than evolutionary developments. Although the no-free-lunch theorem (Wolpert and Macready, 1997) states that there cannot be an algorithm that is capable of solving all problems, evolutionary algorithms can surpass other methods in respect to solving problems where heuristics are either not readily available, or do not give a good performance. For instance, evolutionary algorithms may not outrival traditional methods with simple issues, but in real-world function optimisation, where problems can have non-linear constraints, non-stationary conditions or even noisy environments, classic optimization techniques cannot compare (Fogel, 1997).

Although there has been research carried out that uses an evolutionary approach to produce emerging behaviours in robot populations, there has been nothing conducted specifically on Kilobots. The goal is to have a deeper understanding of, and new evolutionary algorithmic insights, into the emergence of collective behaviours in these limited individuals and how we can improve their robustness, flexibility and scalability. These insights could lead to real-world potential applications in areas that are too dangerous for humans, such as search and rescue and detection of environmental hazards. Furthermore, swarm robotics could be used in the nano-medical domain for early diagnosis of disease or to fight tumour cells (Mavroidis and Ferreira, 2013).

5 METHODOLOGY

Formation control experiments traditionally have robots lined up facing a leader that moves in a forward direction while all the following robots trail along behind in the same path (Figure 2a). This is a typical finite state machine: the model requires robots to either wait or go. For instance, *robotA* will stay in the wait state until *robotB* and *robotC* are also in the wait state, then *robotA* will switch to the go state, whereby it continuously measures its distance to the robot in front of it, trying to reduce it. When the distance is reduced, *robotA* will switch back to the wait stage and the process is repeated.

In contrast to this method, we will use a decentralised algorithm which does not require any states. In a decentralised problem-solving system composed of simple interacting entities, such as an ant colony, there is no leader that determines the activities of the group, nor are there individuals informed of a global pattern to be executed. Therefore, this type of algorithm can be seen as more robust and flexible compared to others as the system's behaviours can arise through the interaction of individual robots. Furthermore, we will change the overall formation structure from a linear to a hexagon shape (Figure 2b). By using a hexagon shape, robots will need to continuously measure their distance between their surrounding neighbours rather than just the robots in front and



(a)

(b)

Figure 2: A graphic visualisation of a) the traditional formation control structure and b) our proposed formation control structure.

behind. Furthermore, a hexagon shape promotes a high density; we penalise robots that detract from the density of the group.

In this experiment, formation behaviour is integrated with a navigational behaviour to enable a team of robots to reach a goal while avoiding obstacles and simultaneously remaining in formation. Teams are rewarded when the average distance of all the Kilobots from the group centre of mass is minimised. The robots are randomly placed in an enclosed square arena with cylindrical obstacles (Figure 3), whereby they need to reach a set of coordinates on the grid. Through the use of proximity sensors, individuals can measure their distance between neighbouring robots and obstacles. The Kilobot controller uses the readings from the proximity sensors as input nodes and two output nodes control the robot's motors. As the Kilobot has a limited amount of memory and processing power, we will carry out simulations using V-Rep (Rohmer et al., 2013), a 3D world simulation tool specifically developed by Coppelia Robotics for designing and evaluating control algorithms. The best controllers will then be downloaded onto real Kilobots and our hypotheses will be further tested.

5.1 The Evolutionary Algorithm

We use a genetic algorithm for the synthesis of the robot controllers; this will represent the way in which the controller will interact with the environment. To delineate the robot controller, we use a look-up table (LUT) with two columns: the set of circumstances in which the robot can find itself and the actions that correlate to each circumstance (Table 1). The input from the Kilobot's sensors are used to look-up the specific situation and find the corresponding action. A simple example of a situation could refer to obstacles in front, left or right of the Kilobot; 1 representing an obstacle in a specific direction and 0 illustrating an obstacle free path. The actions within the table are linked to the robot's actuators.

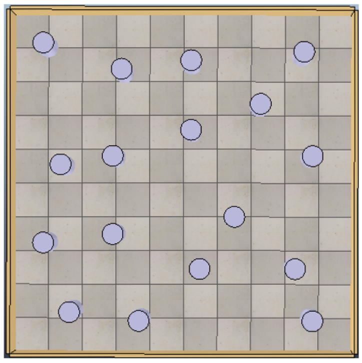


Figure 3: The proposed experimental arena containing 17 randomly placed obstacles.

The initial population is composed of 100 genotypes, all of which are binary encoded and are mapped into the controller of each Kilobot. The robots are homogeneous by design and genetically similar, but as they carry out a random action at the beginning of each run and respond to external stimuli, robots can exhibit heterogeneous behaviours. Each run of the experiment lasts a fixed number of generations and the population in subsequent generations is produced by a combination of tournament selection and crossover. In every generation, we evaluate the genotypes of the population by means of a fitness function. The best performing genotypes of every population are allowed to reproduce in order to create new offspring where are all subject to a low level of mutation.

Table 1: A simple example of a LUT.

Situation	Action
000	Forward
001	Forward
010	Forward
011	Forward
100	Left
101	Left
110	Right
111	Stop

5.2 The Fitness Evaluation

The fitness of the genotype is computed by measuring the performance of the corresponding robot in a group. The function F is calculated by evaluating the behaviour of the group of Kilobots for a number of trials and then averaging the obtained values. This fitness function is designed to favour exploration, fast reaction to obstacles and coordinated motion. For simplicity's sake, we have split the function into two parts F_{e1} and F_{e2}. F_{e1} is a weighted average of the components F_c (collision) and F_x (exploration).

$$F_c = \frac{T_c}{T}, \tag{1a}$$

where T_c is the number of cycles prior to the occurrence of a collision and T is the total number of cycles. Here, the Kilobots evolve to avoid collision by using the LUT. When the robot finds the correct action for their situation, their fitness is increased.

The second component rewards Kilobots that explore the arena:

$$F_x = \frac{z(T_c)}{Z(T_c)}, \tag{1b}$$

where $z(T_c)$ is the number of zones visited by cycle T_c and $Z(T_c)$ is the maximum number of zones that can be visited in T_c cycles.

F_{e2} is calculated as the average density of the robot group throughout the simulation:

$$F_{e2} = \frac{\sum_{i=1}^{T} density(t_i)}{T}, \tag{2a}$$

where there are T intervals.

The density of the group at time t_i is calculated as the average Euclidean distance of the n robots to the centroid of the group:

$$density(t_i) = \frac{\sum_{j=1}^{n} dist(j, centroid)}{n} \tag{2b}$$

We evolve the Kilobots by averaging the fitness functions F_{e1} and F_{e2}. As the robots exhibit potentially heterogeneous behaviours, their individual fitness values in F_{e2} are dependent on the other robots in the current population, which introduces a further complexity to the problem. That is, if one robot performs poorly in the formation, the other robots' fitnesses will also be affected.

6 CONCLUSIONS

In this position paper we outlined the advantages of using evolutionary computation on individuals with limited capabilities. We also proposed an evolutionary algorithm which we believe is simple, yet robust and flexible enough to evolve a swarm of potentially heterogeneous, mobile robots to carry out collective behaviours, in particular, formation control. Furthermore, by evolving robots with limited sensing abilities, we contend that the reality-gap can be more easily overcome as there are less parameters that need to be validated in comparison to other robots that have a more complicated set of sensors and motors. Despite this, there is still a potential chance of a reality-gap, so simulations need to be carefully modelled to retain as many features of the robot as possible.

In future work, we intend to examine the system's behaviour and the individuals' behaviours with the evolutionary approach outlined in this paper by addressing two main issues: composition and a method for selection. Robots must either have the same rules or employ different ones (genetically homogeneous or heterogeneous) and they will be evaluated using a new fitness that combines individual fitnesses and team fitnesses. Developing upon this, we will then use the best controller on the hardware to test our hypotheses further. As discussed in this position paper, this may be easier to accomplish by using simple swarms of robots, thereby reducing cost and time as well as diminishing the reality-gap.

REFERENCES

Baldassarre, G., Trianni, V., Bonani, M., Mondada, F., Dorigo, M. and Nolfi, S. (2007). Self-organized coordinated motion in groups of physically connected robots. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(1):224–239.

Brooks, R. A. (1992). Artifical life and real robots. In *Toward a practice of autonomous systems: Proceedings of the First European Conference on Artificial Life*, page 3. MIT press.

Dorigo, M., Trianni, V., Şahin, E., Groß, R., Labella, T. H., Baldassarre, G., Nolfi, S., Deneubourg, J.-L., Mondada, F., Floreano, D., and Gambardella, L. M. (2004). Evolving self-organizing behaviors for a swarm-bot. *Autonomous Robots*, 17(2-3):223–245.

Fogel, D. B. (1997). The advantages of evolutionary computation. In *BCEC*, pages 1–11.

Mavroidis, C. and Ferreira, A. (2013). *Nanorobotics: current approaches and techniques*. Springer Science & Business Media.

Mondada, F., Franzi, E. and Ienne, P. (1994). Mobile robot miniaturisation: A tool for investigation in control algorithms. In *Experimental robotics III*, pages 501–513. Springer.

Mondada, F., Pettinaro, G. C., Kwee, I., Guignard, A., Floreano, D., Gambardella, L., Nolfi, S., Deneubourg, J.-L. and Dorigo, M. (2002). Swarm-bot: A swarm of autonomous mobile robots with self-assembling capabilities. In *Proceedings of the Workshop on Self-Organisation and Evolution of Social Behaviour*, number LIS-CONF-2002-007, pages 11–22.

Rohmer, E., Singh, S. P. and Freese, M. (2013). V-rep: A versatile and scalable robot simulation framework. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1321–1326. IEEE.

Rubenstein, M., Ahler, C. and Nagpal, R. (2012). Kilobot: A low cost scalable robot system for collective behaviors. In *Robotics and Automation (ICRA), 2012 IEEE International Conference*, pages 3293–3298. IEEE.

Rubenstein, M., Cornejo, A. and Nagpal, R. (2014). Programmable self-assembly in a thousand-robot swarm. In *Science*, pages 795–799.

Trianni, V. and Dorigo, M. (2006). Self-organisation and communication in groups of simulated and physical robots. *Biological cybernetics*, 95(3):213–231.

Trianni, V., Nolfi, S. and Dorigo, M. (2008). Evolution, self-organization and swarm robotics. In *Swarm Intelligence*, pages 163–191. Springer.

Valentini, G., Hamann, H. and Dorigo, M. (2015). Efficient decision-making in a self-organizing robot swarm: On the speed versus accuracy trade-off. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 1305–1314.

Wolpert, D. H. and Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82.