Evolutionary-learning framework: improving automatic swarm robotics design

Faqihza Mukhlish and John Page
School of Mechanical and Manufacturing Engineering, University of New South Wales, Sydney, Australia, and Michael Bain
School of Computer Science and Engineering, University of New South Wales, Sydney, Australia

Abstract

Purpose – The purpose of this paper is to review the current state of proceedings in the research area of automatic swarm design and discusses possible solutions to advance swarm robotics research.

Design/methodology/approach – First, this paper begins by reviewing the current state of proceedings in the field of automatic swarm design to provide a basic understanding of the field. This should lead to the identification of which issues need to be resolved in order to move forward swarm robotics research. Then, some possible solutions to the challenges are discussed to identify future directions and how the proposed idea of incorporating learning mechanism could benefit swarm robotics design. Lastly, a novel evolutionary-learning framework for swarms based on epigenetic function is proposed with a discussion of its merits and suggestions for future research directions.

Findings – The discussion shows that main challenge which is needed to be resolved is the presence of dynamic environment which is mainly caused by agent-to-agent and agent-to-environment interactions. A possible solution to tackle the challenge is by incorporating learning capability to the swarm to tackle dynamic environment.

Originality/value – This paper gives a new perspective on how to improve automatic swarm design in order to move forward swarm robotics research. Along with the discussion, this paper also proposes a novel framework to incorporate learning mechanism into evolutionary swarm using epigenetic function.

Keywords Reinforcement learning, Evolutionary algorithm, Epigenetics, Multi-agent learning, Swarm robotics

Paper type Literature review

1. Introduction

Various strategies for programming intelligent robots to fulfil a mission in a harsh environment usually carried out by humans have been widely investigated. However, less research incorporating harsh environments has been carried out investigating a group of simple robots, namely, swarm robotics. Swarm robotics is currently earning a lot of attention because of its flexibility, scalability and robustness. Instead of using a complex robot which may not be cost effective due to the high possibility of failure in a harsh and unpredictable environment, swarm robotics has been explored in the past few years to investigate its ability to handle failure resulting in performance degradation in addition to issues such as the effect of an unpredictable environment. In recent years, inspired by collective behaviours of social animals, the idea of distributed low-level intelligence as a decentralised method to control a group of agents has been studied and developed (Brambilla et al., 2013; Bayindir, 2016). However, problems arise when a self-organising swarm tries to tackle tasks in a dynamic environment. Additionally, building a strategy to deal with such an environment is not a trivial task. Hence, an approach to decentralised control, strategy and cooperation between agents is necessary for the swarm to cope with environmental changes or an individual agent’s failure.

A common design strategy to achieve distributed low-level intelligence is using behaviour-based design inspired by social animals (ants, bees, birds, termites and fish).
Since many behaviour-based models are specifically linked to specific tasks and behaviour, these models can be applied to swarm robotics quite easily (Bogue, 2008). A seminal work by Reynolds (1987) on a distributed behavioural model showed that complex behaviour can emerge from aggregation of individual simple actions. However, a behaviour-based model may solve only a specific problem. As the complexity of problem increases, behaviour-based design requires more effort in defining a mathematical model which is adaptable to an unpredictable environment. Hence, a more effective method of achieving the collective behaviour of a swarm is required. This problem can be easily resolved using an automatic design method to achieve collective behaviour without having to define the model of the problem explicitly.

One such promising swarm design concept that has been studied in our previous work (Chi et al., 2014) is the use of evolutionary algorithms (EAs) which can be considered as an automatic design method. However, the solution that resulted from incorporating EA to swarm robotics, called an evolutionary swarm, often gives unpredictable results due to random mutation and a changing environment (Yi et al., 2017). This shortcoming is due, in part, to the lack of consideration of the dynamics of the environment caused by agent-to-agent and agent-to-environment interactions. Hence, for an evolutionary swarm to operate in a dynamic environment, information on the stimulus from the environment must be collected and used later as “knowledge” for the next generation of the EA to formulate better actions or behaviours: Improving an action based on external stimulus is commonly defined as a learning mechanism (Mitchell, 1997). However, since EAs were mainly inspired by Darwinian evolution (Darwin, 1872), there is no direct mechanism for external knowledge perceived by individuals of one generation to be inherited by those of next generation (sometimes known as the Lamarckian paradigm). In other words, heritable learning does not play any significant role in the EA. Thus, an approach to incorporating a learning framework into an evolutionary swarm is required in order to widen the perspective on how to improve swarm capability in a dynamic environment.

To generate a comprehensive discussion, this paper begins by reviewing the current state of proceedings in the field of swarm robotics to provide a global perspective of swarm design and automatic design in general. The fundamentals of swarm design are discussed to provide a basic understanding in order to stimulate discussion with regard to foundation of swarm robotics. There are several recent articles available which review swarm robotics from different perspectives (see, Brambilla et al., 2013; Bayindir, 2016). This should lead to the identification of which issues need to be resolved in order to move forward swarm robotics research. Then, viable solutions are presented to identify future directions and how the proposed idea of incorporating learning mechanism could benefit swarm robotics design. In the last section of this paper, we propose a novel evolutionary-learning framework for swarms and explore how to promote their development.

2. Fundamental of swarm robotics design
Swarm robotics is a method of controlling a group of collaborative simple agents towards completing tasks. Swarm robotics has been defined as:

A novel approach to the coordination of large numbers of robots [...] [and] a study of how 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 environment. (Şahin, 2005, p. 3)

The source of inspiration for swarm robotics comes from collective behaviour of social animals. When simple individuals cooperate in groups, they can perform complex behaviours as demonstrated by ants, bees, birds, termites and fish. Distributed behavioural models were first investigated in a seminal work by Reynolds (1987). The author showed
that complex behaviour, in this case flocking behaviour, can emerge naturally from the
aggregation of individual simple rules, namely collision avoidance, velocity matching and
flock centring. Furthermore, many researches on social animals have shown that there is
indeed a group intelligence that emerges from individual behaviour (Ame et al., 2006;
Camazine, 2001; Couzin et al., 2005). The main characteristics necessary for self-organising
swarm to be collectively functioning are the following:

- a self-organised swarm is at least partially autonomous;
- a self-organised swarm is situated in dynamic environment caused by agent-to-agent
  and agents-to-environment interaction;
- a self-organised swarm is only able to sense on a local scale and interact with nearby
  agents within the group;
- a swarm control is decentralised, and each agent does not have access to the swarm’s
  global behaviour;
- all agents inside a self-organised swarm cooperate to tackle a given task collaboratively; and
- a self-organised swarm can access and communicate the knowledge gained such as
data monitoring to a base.

Developments of collective behaviour of self-organised swarm robotics have been studied
extensively in tackling a given task, applying decentralised control and adapting them to
survive in a particular environment (Bayindir, 2016). In the past decade, methods to design a
collective behaviour of swarm robotics can be grouped into two approaches: behaviour-
based and automatic design. The latter is the way of designing swarm strategy
automatically without explicitly modelling the problem and behaviour of the swarm. The
benefits of this method are flexibility, scalability and robustness. On the other hand,
behaviour-based design is using predefined mathematical models or strategies to direct the
swarm behaviour, usually known as top-down design. Collective behaviour is decomposed
into individual behaviours. It is, in fact, more predictable than applying automatic design
since particular behaviours are already known (Bogue, 2008). However, behaviour-based
design may solve only a specific problem in a fixed environment. In general, automatic
design is more flexible than behaviour-based design and it provides the swarm with the
agility to cope with dynamic problems.

2.1 Behaviour-based design
Developing a swarm robotics system using behaviour-based design has often been
inspired by the observation of the collective behaviours of social animals. The design
process is comprised a cycle of modelling the behaviour, implementation on swarm
system, evaluation and improvement of the current model. Through this process, many
swarm behaviour-based designs have been proposed in recent decade. For example, an
object retrieval task using a group of robots was accomplished (Labella et al. 2006) by
applying behaviour of foraging and the division of labour of ants (Deneubourg et al., 1987).
Adopting behavioural models based on social animals may ease the swarm design
process, since the mathematical model can be derived easily. Moreover, there are two
forms of modelling paradigms that commonly used, namely, probability finite state
machine (PFSM) and virtual physics-based approach.

2.1.1 Probability finite state machine. PFSM was first coined by Minsky (1967) and the
main idea of PFSM is the transition between states depends on a probability value. In
swarm design, PFSM is often used as representation of class of behaviours and transition
between behaviours. Implementation of PFSM in swarm design can be seen in several
developments of collective behaviours, such as aggregation (Soysal and Şahin, 2005; Correll and Martinoli, 2011; Garnier et al., 2009), chain formation (Nouyan et al., 2008) and task allocation (Labella et al., 2004, 2006; Liu et al., 2007).

The transition threshold between states can be either constant or can vary. A finite state machine that uses constant threshold value can be found in the aggregation model studied by Soysal and Şahin (2005). Specifically, the agent changes behaviour to another behaviour dependent on the probability value associated between these two behaviours as depicted in Figure 1. On the other hand, the dynamic transition threshold value is defined through a mathematical function of parameters associated to available states of the agent as shown in Figure 2. In study conducted by Garnier et al. (2009), the transition function depends on number of robots and maximum capacity in the current state.

2.1.2 Virtual physics-based design. In virtual physics-based design, each robot is considered as particle that applies virtual potential forces to another robot. Usually, the virtual force is a function of position, direction and distance between the entities (robots, targets and obstacles). More specifically, repulsion and attraction values are also used to differentiate force’s direction. The function of virtual force generally allows each robot to sense and distinguish neighbouring robots, targets and obstacles.

Maintaining formation of the swarm, whilst traversing a terrain, is a common collective behaviour that utilise virtual physics-based design. One of the seminal works using virtual physics-based design was by Khatib (1986) which considered another vehicle as a moving obstacle in a field of artificial forces. Similar works also can be found in the works of Reynolds (1987) which formulate a general flocking behavioural model comprises of collision avoidance, velocity matching and flock centring using attraction and repulsion forces. Later, the design methods considered as virtual physics-based design also can be

---

**Figure 1.**
PFSM with constant probability transition value

**Figure 2.**
PFSM with dynamic probability transition value
found in several frameworks, such as physicomimetics frameworks proposed by Spears et al. (2004) and a nonlocal kinetic model proposed by Fetecau (2011).

In a very specific behaviour such as flocking and foraging which require medium to share knowledge and information, stigmergic approach inspired by how social animal share their information is commonly used in swarm robotics design. In social animal such as ants, they share information using a trail of pheromones allowing them to create paths to the food sources. Using this approach along with two models above, several studies accomplished to mimic animals’ behaviours such as foraging (Labella et al., 2006; Ranjbar-Sahraei et al., 2012).

Because of their generality, these two models are often used as a foundation model and later combined with optimisation or search algorithm to automatically tune model’s parameters.

3. Automatic swarm robotics design

Automatic swarm design is aimed at achieving collective behaviour without any reference to predefined explicit model. Two automatic design methods commonly used to tackle swarm robotics problems are evolutionary computation and reinforcement learning (RL). How swarm robotics design utilises these two methods will be discussed in following subsections. The principles of both methods are discussed and current challenges of applying each method is investigated.

3.1 Evolutionary swarm robotics

In robotics, evolutionary computation (Goldberg, 1989) can be used to encode characteristics of control strategies into artificial chromosomes (Holland, 1992). Each chromosome represents a particular characteristic of each strategy and its fitness value (performance) is evaluated using fitness functions. Chromosomes (a set of strategies) with high fitness value are allowed to breed through genetic operators such as recombination, random mutation and selection. Progenies with higher fitness ratings (new strategies) will replace those with least fitness rated strategies in the previous generation. This development is repeated until the fitness value of the new generation meets the designated criterion. This method of generating control strategy in robotics is defined as evolutionary robotics (ER) as depicted in Figure 3 (Nolfi et al., 2016).

Evolutionary computation has been proven to be successful in solving single-agent problems (Eiben and Smith, 2015). In multi-agent system (MAS) such as swarm robotics, evolutionary computing is also beneficial to the swarm in generating collective behaviour despite the nonlinearities caused by the number of agents and the given tasks. For example,
many studies show that evolutionary computation can be used to acquire collective behaviour such as foraging (Francesca et al., 2012; Gauci, Chen, Dodd and Groß, 2014; Trianni et al., 2003), flocking (Baldassarre et al., 2003), path formation (Kuyucu et al., 2012), clustering (Gauci, Chen, Li, Dodd and Gross, 2014; Hartmann, 2005), collective object transport (Groß and Dorigo, 2004, 2009) and task allocations (Tuci et al., 2008).

3.2 Multi-agent reinforcement learning (MARL)

MARL is the research field that studies RL techniques on MAS and studies the design of algorithms to create such adaptive agents. MARL, like ER, has been used to find a solution undefined beforehand which has behaviour considered optimal within a situation. Related to swarm robotics, a particular learning mechanism that is commonly used is RL (Brambilla et al., 2013; Busoniu et al., 2008). RL on MAS allows the individual to learn a behaviour through trial-and-error interactions with environment and other agents (Kaelbling et al., 1996; Sen and Weiss, 1999; Sutton and Barto, 1998), see Figure 4. In each interaction, the action which is taken by each agent in a corresponding state of the environment is rewarded based on its performance. Using the sum of rewards, action and state are paired into a set of action-states as a rule of swarm behave in a certain state of an environment.

RL in swarm robotics takes the advantages of multi-agent’s settings. First, multi-agents setting allows the learning process to be computed in parallel way. Second, knowledge perceived by an agent from trial-and-error interaction with the environment can be shared with other agents with similar tasks. Teaching and imitation are examples of sharing medium within the group. Lastly, redundancy also can be applied when one or more agents fail in accomplishing tasks. When the failure occurs, the remaining agents can take over and complete the tasks. Additionally, RL has been acknowledged to perform quite well in MAS such as a swarm (Busoniu et al., 2008). Several learning methods considered as MARL have been implemented into, for example, swarm of odour localisation robots (Hayes et al., 2003), stick pulling robots (Li, Martinoli et al., 2004) and RoboCup Soccer (Kalyanakrishnan and Stone, 2007; Riedmiller et al., 2009; Stone et al., 2005). All in all, summary of the current approaches in automatic swarm robotics design can be seen in Figure 5 below.

3.3 Challenges

Despite of the advantages of evolutionary computation and RL in automatic swarm design, there are several challenges in utilising automatic design to achieve collective behaviour in a swarm.

3.3.1 Deception. In solving a problem, both evolutionary computation and machine learning commonly utilise fitness function (also known as objective function). However, the

![Figure 4. Multi-agent reinforcement learning](image-url)
computation process is often misguided by the fitness function and converge on local optima which is deceiving. The fitness function acts as natural pressure for the computation process to improve its performance (Holland, 1992). This function is defined beforehand and derived based on the tasks to be solved which is challenging as complexity increases. Hence, in a dynamic environment, the computation is prone to deceitful solution. Furthermore, Lehman and Stanley (2011) concluded that performance-based fitness function can be misleading, especially for dynamic and complex problems.

3.3.2 Exploration and exploitation dilemma. Another problem arises when a fitness function is being used. The swarm will exploit an action that maximises their performance and easily dismiss any inept behaviour. This problem is tightly related to the deception problem discussed above. To counter that, exploration techniques is used to suppress the exploitation. Exploration is beneficial to seeking an alternative behaviour which may be lead to better performance in the future. However, adding exploration technique is not without problems. The collective behaviour of the swarm will be unstable if too much exploration is applied. Here is where the problem of the exploration and exploitation dilemma arises. This problem is even more complicated in multi-agent learning system because of the presence of co-learning.

3.3.3 Non-stationary behaviour. Automatic design allows each agent to evolve and learn simultaneously. However, the non-stationary behaviours of the multi-agent learning problem arises because of all agents in the system are learning simultaneously. This generates a situation in which each agent is trying to develop its own best behaviour as the other agents’ best behaviour changes. Each agent is, therefore, faced with a moving-target learning problem. This causes the computation process of achieving collective behaviour automatically to become more challenging. The main impact of non-stationary problem in co-learning setting is the risk of unstable behaviours.

3.3.4 The curse of dimensionality. MASs that utilise learning mechanism such as RL, which maps the state into best behaviour suffers from the “curse of dimensionality”. The knowledge of possible a discrete state of the environment which is resulted from MARL grows exponentially as the number of agents increases. As the estimation of possible discrete state or state-action pair becomes bigger, the computation process to choose the best behaviour or policy for the current state becomes more complex over time and requires extra computing time.
4. Improving automatic-based design

4.1 Maintaining diversity

Fitness or objective function is a tool to measure how good current behaviour is in the current situation. Evolutionary computation and learning mechanism are used to find better behaviour that maximise the fitness value. Behaviours with the least good performance will be removed although these maybe beneficial in determining true direction to global optima. This objective-based search paradigm does not necessarily measure how good the intermediate decision is in finding a direction that leads to the objective. Hence, several options to overcome deception will be discussed in following section.

4.1.1 Sustaining behavioural diversity. In fitness-based computation, all available strategies in system’s repertoire converge asymptotically towards a set of behaviours that may be maximising the fitness. The result typically is within a certain range of values and is applied to all agents. This is an issue, because of alternative solutions, which may be leading to a better behaviour, maybe dismissed in the computation process. This problem occurred because of the computation had a preference for exploitation rather than exploration. Hence, the diversity of the solutions should be maintained giving alternative direction for the swarm to explore better behaviours.

Sustaining behavioural diversity is one of many ways to improve the efficiency of fitness-based evolutionary computation in empirical exploration (Goldberg, 1989; Mahfoud, 1997; Sareni and Krahenbuhl, 1998) and also analytically (Friedrich et al., 2008). The diverseness can be achieved by defining a distance between behaviours maintained in the computation process. Hamming distance is a tool that is typically used to quantify the diverseness. Then, neighbouring behaviours will be clustered into a behaviour-set in which the evolutionary computation is carried out. In multi-agent problem, such as swarm, this seems natural method to generate a potential improvement in behaviour exploration and in overcoming the deception.

4.1.2 Novelty search. Other than maintaining the diversity of agent’s behaviour which is an extended version of fitness-based computation, there is another method to overcome deception known as novelty search suggested by Lehman and Stanley (2011). This approach quantifies the novelty of new behaviour achieved in the exploration process using a novelty metric. The metric measure how far away (novel) the new behaviour is from the rest of the available behaviours. Then, all unique or novel behaviours are placed in the behavioural space along with the novelty value. The value of novel behaviours is characterised by the sparseness of each behaviour at any point in the behavioural space. Using the sparseness value in the behavioural space, each behaviour achieved in each step of computation is rewarded a value based on its novelty. The sparseness value $\rho$ is an average distance to the $k$-nearest neighbours $\mu$ at a point $x$ as can be seen below:

$$\rho(x) = \frac{1}{k} \sum_{i=0}^{k} \text{dist}(x, \mu_i).$$

(1)

Using novelty search, the exploration is expected to be moving towards new behaviours and avoiding deception. A new behaviour is considered as a novel behaviour if its located in the sparser regions and then received more novelty rewards. On the other hand, a less novel behaviour will be rewarded less than a novel one and grouped into a group of similar behaviour. The novelty metric allows the computation process to identify where the exploration has occurred and where it currently exploring. By maximising the novelty metric, the search direction is simply moving towards what is new. Thus, because no objective function is involved, novelty search can be used to overcome deception.
4.2 Balancing exploration-exploitation

Through exploration, automatic-based design discovers new behaviours in respective states. Then, the best behaviour is selected through exploitation which maximise agent’s performance. Exploration and exploitation in a learning system should be in balance. Taking too much exploration will not lead the system to stability and may use resources to explore unnecessary space which not beneficial for the task. On the other hand, exploiting current best behaviour can also lead to deception as discussed in previous section. In fact, using the same principle as overcoming deception, the exploitation problem can be solved by maintaining diversity and alternating emphasis between exploration and exploitation based on a certain function or probability threshold.

Particular to exploration–exploitation dilemma in multi-agent learning, one way to proceed is by utilising an ε-greedy exploration method. In this method, most of the time, with probability of \((1-\epsilon)\), the algorithm exploits current best behaviour but once in a while explores randomly to another behaviour with a small probability \(\epsilon\). Another alternative is by using Boltzmann distribution which is a “SoftMax” approach. Exploitation of current best behaviour and exploration of alternative behaviour is based on a parameter \(\tau\) to balance exploration and exploitation. The probability \(P_i\) of a chosen behaviour \(\mu_i\) among available behaviours (\(\mu_1, \mu_2, \mu_3, \ldots, \mu_j\)) in a state \(S\) denoted as:

\[
P_i = \frac{e^{f(S, \mu_i)/\tau}}{\sum_j e^{f(S, \mu_j)/\tau}}.
\]

4.3 Achieving Nash equilibrium

The non-stationary behaviours of the multi-agent learning problem arises because of all agents in the system are learning simultaneously. Each agent is, therefore, faced with a moving-target learning problem: the best behaviour changes as the other agents’ behaviour change. Moreover, in fully cooperative tasks, a non-stationary behaviour is the main problem in achieving stability and adaptation of the learning system. To counter this, convergence to a stationary behaviour is needed because it reduces the non-stationary behaviours problem.

Achieving a stable behaviour in a co-learning setting is not a trivial task, especially if each agent is oblivious to other’s behaviour. However, even if each agent has a knowledge of others behaviour, it is still difficult to select best behaviour given other agent’s behaviour. To counter this condition, a set of learned behaviours has to follow a Nash equilibrium criterion (Nash, 1950) which is a condition where a best behaviour given other agents’ behaviours is exist in each agent’s behaviours repertoire. (Busoniu et al., 2008; Tuyls and Weiss, 2012). However, achieving the Nash equilibrium in a co-learning process is not without a problem, because each learning process is dynamic and which learning direction is taken by an agent is difficult to observe or invisible to others.

To meet the Nash criteria in a co-learning setting, Bowling and Veloso (2001) proposed that there are two properties of multi-agent learning algorithm have to be followed which are:

1. **Property 1. (Rationality)** If other agents’ behaviours converge to stationary behaviours then the learning algorithm will converge to a behaviour that is a best-behaviour to their behaviours.

2. **Property 2. (Convergence)** The learner will necessarily converge to a stationary behaviour. This property will usually be conditioned on the other agents using an algorithm from some class of learning algorithms.

The first property requires the agent to learn the best behaviour when other agents are applying stationary behaviours. This will restrict a learning agent to learn some behaviour that
is independent to other agent’s behaviour. Moreover, this property will give an ability for an agent to learn rationally. The latter property can be achieved only if an agent is learning with respect to rational agents or is learning with respect to agents with stationary behaviours.

In combination, these two properties guarantee that the learner will converge to a stationary behaviour given the behaviour of the other agents. There is also a connection between these two properties with Nash equilibrium. When all agents are rational and convergence, then a Nash equilibrium exists since all the behaviour learned by each agent is the best response to other agent’s behaviours.

4.4 Adaptation
Adaptation to a dynamic environment is a major challenge in automatic swarm design. As discussed above, swarm design using evolutionary computation is not inherently adapting to the condition of the environment but rather follow the direction of the fitness function. On the other hand, RL which is a common mechanism used in MASs also has many challenges in mapping behaviours to states of a dynamic environment. Despite improving both methods separately, combining evolutionary computation and learning mechanism into one framework using Lamarckian principle has shown a significant increase in performance in a dynamic environment (Giraud-Carrier, 2002; Le et al., 2009). Extended combination of evolutionary computation and learning will be discussed in following section as a new perspective for designing a swarm automatically.

5. Evolutionary-learning framework
While the evolutionary computation is commonly used to define a swarm behaviour automatically as reviewed in a previous section, multi-agent learning methods are found to have flexibility to a dynamic environment and other agents’ behaviour. Combination of these two methods would be beneficial to incorporating heuristic search and adaptation into system control (Mukhlish et al., 2018).

Evolutionary computation is primarily encouraged by Darwinian evolution (Goldberg, 1989; Holland, 1992). However, this theory does not utilise parameters from the environment in shaping an individual’s behaviour nor passing it on to the next generation. Besides this evolution theory, there is an evolution theory proposed by Lamarck which is beneficial in taking parameters from environment that will be used later to improve the next progenies. Specifically, the generation’s performance is enhanced using parental experience of nutrition, contaminants, nurturing behaviours and social stress or fear (Lim and Brunet, 2013). This principle is analogous to a learning mechanism such as RL. Moreover, studies conducted by Waddington in 1984 (Waddington, 2012) showed that there is a layer above a gene, called Epigenetic in Biology, that regulate its expression dependant to external stimulus from environment. The stimulus and experiences are somehow stored in a form of Epigenetic layer which is acting as a regulatory function and is inherited by the next generation (Wang et al., 2017).

5.1 Epigenetics algorithm
Epigenetics layer is regarded as a tool for an agent to respond to environmental stimulus by modifying its phenotypic expression. This means that an agent has some type of regulatory structures that receive an input from the environment (external stimulus) and then use it to regulate genotypes as a form of expression regulation. Moreover, based on works in Biology, these regulatory structures and their function are inherited between generations (Wang et al., 2017). Inspired by these mechanisms, an algorithm which uses such principle was proposed, known as epigenetic algorithm (EpiAL), by Sousa and Costa (2011). In their proposed model, interaction between the agent-environment is depicted in Figure 6.
EpiAL model is composed by two fundamental entities: agent and environment. External stimulus from environment is received by the agent through its sensors. The stimulus is then passed onto epigenetic layer which is acting as regulatory structures. After that, appropriate genetic codes are selected and regulated given the received stimulus. Then, the selected genetic codes are expressed modifying current behaviour of the agent. Each cycle of EpiAL algorithm, the performance of selected behaviour is measured to calculate the relation between stimulus and the genetic codes a methylation value. This methylation value is used to evaluate weights of the epigenetic algorithm.

Using this model, the authors were able to represent the regulatory function of epigenetic layer into a mathematical model to respond to a dynamic stimulus from environment. Similar works in other studies also demonstrate that the utilisation of regulatory structures is a key to handling a dynamic environment (La Cava and Spector, 2015; Tanev and Yuta, 2008; Periyasamy et al., 2008).

5.2 Behavioural space and environment state
The epigenetic layer translates the environment state to behavioural space which contains of expressions composed of selected genes by epigenetic layer (see Figure 7). Additionally,
this is analogous to function of epigenetic which has been studied in epigenome research area as a product of adaptation to environment. This perspective leads to the key idea on regulatory function of epigenetic layer.

### 5.3 Regulatory function

The regulatory function of epigenetic layer can be derived from a knowledge of external stimulus from environment. One common method to obtain the function is through trial-and-error interaction with the environment. This interaction is beneficial to get temporal and spatial knowledge of the dynamic environment. Selected behaviour in each interaction phase is evaluated and will be rewarded based on its performance in the current environmental state. The received rewards are then used to form a relation between individual’s behaviours and environmental states. This mechanism results in a regulatory function which will select a set of genetic codes forming a behaviour that gives a maximum reward given an environmental state. The model of this mechanism is depicted in Figure 8.

Moreover, the received reward is analogous to methylation process in Biology. In other words, behaviour is regulated based on the methyl value of epigenetic layer. A gene which is covered with high methylation value given the external stimulus is more likely to be expressed. On the other hand, a gene with low methylation value will likely to be silenced. Furthermore, rewarding system in learning machine can be utilised to methylate (strengthen) or demethylate (weaken) the methyl value. These methylation marks are inherited by next generation through evolutionary operator, namely selection, recombination, mutation and regeneration.

### 5.4 Sensory component – state observer

However, in swarm robotics, knowledge of the external stimulus from environment can only be measured using available sensors on board, which is of course limited. Generally, utilised sensors can only quantify simple specific units such as temperature. Knowledge of the dynamic of the environment is thus necessary to select appropriate behaviour although only a limited sensory input is usually available. Thus, to get the quality of the environment a combination of sensors’ values is necessary to recognise the current state. However, since

---

**Figure 8.**
Learning mechanism for epigenetic layer
the dynamics of the environment is known as deterministically chaotic which was summarised by Edward Lorenz (1963) as “Chaos: when the present determines the future, but the approximate present does not approximately determine the future”, taking a priori model of observer function which maps sensors’ values to environmental state is difficult. It is even more difficult to craft an appropriate function if the complexity of the environment increases. To open the possibility in automatic manner, the conceptual of epigenetic layer should be improved by incorporating a learning mechanism, which is found useful to predict and observe chaotic systems (Pathak et al., 2017), to determine the dynamics of the environment. This can be done by embedding a state observer into epigenetic layer. Then, to be applicable to automatic design, the state recognition is also obtained from trial-and-error interaction with environment as shown in Figure 9. This complete framework will be investigated in our future research.

The utilisation of epigenetic layer gives ability to apply knowledge of an environment and related behaviours into the swarm strategy. The combination of selected genes represents the phenotype. Distance between behaviours can be calculated using metric as given by Equation (1). This metric is beneficial to limit duplication of similar regulatory function in epigenetic layer and prevent the agent relearning similar behaviours.

5.5 Collective learning and evolution
Evolutionary computation provides medium of information sharing between agents through selection, recombination, mutation and regeneration. Along with corresponding genetics value, epigenetics is also inherited to next generation through evolutionary process. Collective learning occurs in this process through recombination. All behaviours with similar value are clustered into groups based on novelty metric discussed above. And then, behaviours in the same group will be evolved together and a new behaviour that constructing significantly different value (novel) is then be inserted into a behaviour to be explored. This method acts as selective pressure for behaviour-set in the behavioural space to overcome the deceptive issue. Another challenge to be resolved is exploration–exploitation dilemma. To balance exploration and exploitation, an $\epsilon$-greedy approach can be utilised as an option. By combining exploitation–exploitation balancing with the novelty metric, the exploration is expected to move towards new behaviours.

![Evolutionary-learning framework](image-url)
6. Future work

This approach offers promising way to improve swarm capability using an automatic design approach, especially extending evolutionary swarms and learning. Since learning capability is the main key for a swarm to tackle dynamic environment, it is important to investigate learning mechanisms for swarms (Mukhlish et al., 2018). Once the learning mechanism is identified, future implementation of swarm learning could be explored in swarm robotics research. In order to move forward in this direction, several steps need to be achieved.

Evolutionary-learning framework formulation: incorporating a learning mechanism into evolutionary computation can be achieved by extending EpiAL model by Sousa and Costa (2011). The main goal of this framework will be to respond to stimuli from the environment based on experience and inherit the behaviour through an evolutionary process. The response is chosen from among available responses (actions) based on the performance at the current state. The response will be built based on interaction with environment as a learning process. Then, these behaviours will be inherited to the next generation through an evolutionary process. Specifically, this framework will be applied and analysed on an individual agent.

Evolutionary swarm learning framework formulation: Once the evolutionary-learning framework on an individual agent is achieved, a parallel framework will be formulated. The method of adapting mechanisms of single learning entities into multi-agent learning will be done using collaborative and competitive learning. Each agent will learn competitively as an individual to achieve the best response to collaboratively improve swarm adaptation to the environment.

Validate the proposed framework: when the required framework and algorithms are mathematically proven and implemented, it will be possible to extensively simulate the framework in order to measure its performance. The framework has to be evaluated in a dynamic environment testbed. One way to achieve dynamic environment is using a simulation (Page and Mukhlish, 2017). The simulation will be designed as an unknown environment and evaluate the capability of the swarm to survive and thrive.

The potential case that will be used in the future work is search and rescue mission on the ocean using swarm of unmanned aerial vehicles (UAVs) which has been studied in our previous works (Chi, 2014; de Crespigny, 2015; Wright, 2015). Search and rescue mission on the ocean is known to be very challenging and require number of complex actions that require material and human resources (Vidan et al., 2016). However, it is an essentially two dimensions problem that also avoids blanking that occurs in land-based search and rescue. The dynamic of environmental setup in the simulation would be dependent on wind direction, sun position, sea state and drift of the victims. Sensory components available on the UAVs would be a camera to capture sea surface and detect floating victims and proximity sensor to detect nearby UAVs. To sense the dynamics of the environment, for example, the image quality captured by the camera depends on environmental properties such as the sun reflection off the sea surface. All these setups are considerably adequate to provide environmental dynamics for validating the evolutionary-learning framework as discussed by Helleboogh et al. (2006). Moreover, the authors stated that simulated dynamic environment should be changing its dynamics in ways beyond the agent’s actions. Dynamism in which each agent experiences in a dynamic environment depends on other autonomous agent’s acts and changing environmental properties over time.

7. Conclusion

Swarm robotics is currently earning a lot of attentions because of its flexibility, scalability and robustness. Both approaches to design, namely, behaviour-based and automatic design, of designing swarm robotics are discussed in this paper. Challenges related to current developments in both design methods and directions that could be taken for further development and improvement are also discussed. Development of behaviour-based and
automatic design is necessary to provide options for designing a swarm robotics behaviour. Especially with behaviour-based design, further development should strengthen the foundation model of swarm behaviour and provides a useful validation platform for social animal behavioural model for biological studies.

At the time of this paper was written, the development of automatic design is facing several challenges that are inherently related to computation problem such as deception, exploration–exploitation dilemma, non-stationary behaviour and the curse of dimensionality. All these challenges are mainly caused by the multi-agents rule setting and the dynamic environment. Although automatic swarm robotics designs either using evolutionary or learning mechanisms have shown promising results in many researches reports, research into tackling challenges stated above are required for future implementation of swarm robotics in real applications within dynamic environment.

In general, building a strategy to achieve an advanced self-organising swarm in a dynamic environment is not a trivial task because of its complexity. Although behaviour-based methods inspired by animal behaviour are mature enough, these may solve only a specific problem with a stationary environment. This has created a necessity for automatic-based methods to generate a more flexible and robust swarm design. As an advancement in automatic design methods, incorporating learning mechanism into an evolutionary swarm is a possible method to insert a dynamic of environment into design process.

In this paper, a novel evolutionary swarm learning method using epigenetic inheritance is proposed and a possible learning mechanism is also identified as a response to a dynamic environment. Simulation, consisting of agent-based modelling and dynamic environments, should be utilised to validate and analyse the performance of the proposed framework initially. Finally, as result, the research is expected to contribute to automatic-design methods for swarm robotics by adding learning capability and environment-aware behaviour.

References


Crespigny, A.C. de. (2015), “Swarm learning techniques”, Bachelor of Engineering, Kensington, School of Mechanical and Manufacturing, Faculty of Engineering, The University of New South Wales, Sydney.


Wright, A.J. (2015), “Clonal selection for the evolution of heterogeneous unmanned aerial vehicle swarms”, Bachelor of Engineering, School of Mechanical and Manufacturing, Faculty of Engineering, The University of New South Wales, Kensington and Sydney.


Corresponding author
Faqihza Mukhlish can be contacted at: f.mukhlish@student.unsw.edu.au

For instructions on how to order reprints of this article, please visit our website: www.emeraldgrouppublishing.com/licensing/reprints.htm
Or contact us for further details: permissions@emeraldinsight.com