Adaptive Methods for Swarm Intelligence

Literature Review

Sihle Calana
CLNSIH001@myuct.ac.za

ABSTRACT

This paper reviews research based on evolutionary adaption algorithms with regards to Swarm Robotics. The paper primarily focusses on the minimal Environment Driven Evolutionary Algorithm and Novelty Search and how both optimise task completion.

1 INTRODUCTION

Evolutionary robotics is an approach to robot development in the absence of human interference in such a way that machines are capable of learning in a decentralised manner [19] [8]. This means that there is no command centre to control all the robots in a swarm. Instead the behaviour of each robot is influenced by how they respond with their environment and each other [11]. This aids in designing flexible, robust and scalable collective behaviour patterns to coordinate a large number of robots to solve complex tasks [18]. This is useful for automatically generating artificial brains and simulated worlds that imitate real life, allowing for unconstrained exploration and testing of scientific hypotheses of biological processes [7] .

The way in which a collective group of robots work together to complete a task emulates how a colony of ants cooperate when foraging or a how a flock of birds collaborate when navigating and exploring the skies or how a school of fish communicate with one another.

This could be extremely useful in experimentation with regards to answering questions relating to evolution in the natural world, not just in artificial life [7]. This also greatly assists in completing hazardous tasks that may be harmful to human beings or difficult to reach such as search and rescue, construction underwater & space or humanitarian demining to name a few [18].

In this paper we shall review existing research done on different swarm intelligence adaptive methods, examples of their uses, discuss and compare the methods used in the research as well as their credibility. This is done with the goal of hybridizing different evolutionary algorithms and help evaluate the best way to optimise the efficiency of swarm robotics with regards to completing tasks.

2 SWARM INTELLIGENCE

Evolutionary computing is a section of computer science that uses biologically inspired natural evolution techniques to solve problems. When referencing biological evolution we are referring to an environment where the fittest individuals in the population are the ones that are most successful at surviving and passing their genes on to their offspring through reproduction. In the context of evolutionary algorithms, this would be called crossover [6].

In the evolutionary computing context, the fitness of robotic agents (individuals) would be determined by the fitness function which measures the progress towards the objective in a search space. This includes objectives such as to multiply and survive in an environment or to maximise exploration of the current environment. The evolution and adaptation of a robotic controller is dependent on evolutionary algorithms which mirrors natural selection [14] [20].

Living organisms are comprised of physical and behavioural features that determine how they interact with others and their surroundings. These attributes, called phenotypes, are determined by genes and the alleles attributed to those genes. Much like living organisms, robotic agents have invisible code (like genotypes) that influence their observable actions (similar to phenotypes) [6].

Now we can better understand Swarm Intelligence. Swarm Intelligence is a set of computing algorithms comprised of several multi-agent systems that mimic swarms in nature such as a swarm of bees [].

2.1 Methods for Adaptive Behaviour

There are several research papers on the numerous methods of adapting swarm behaviour such as using Map-Elites, Novelty Search or the minimal Environment-driven Distribution Evolutionary Adaptation (mEDEA) algorithm as examples. However this paper will mainly be focusing on mEDEA and on the Novelty Search approach.

2.1.1 The Medea Algorithm

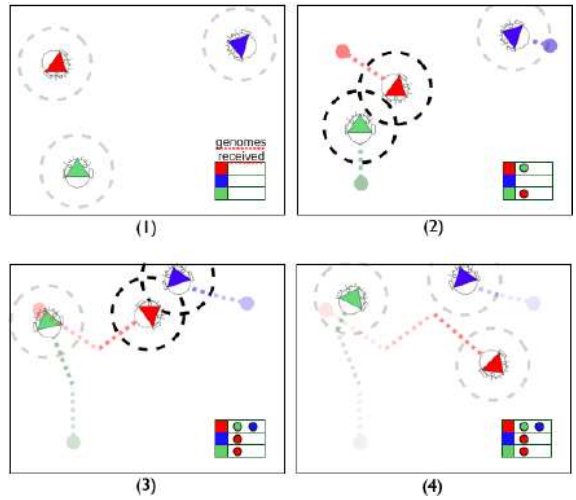
The mEDEA algorithm was developed to address the Environment Driven Evolutionary Algorithm (EDEA) shortfalls regarding the possible conflict in motivations with regards to the fitness function. mEDEA focuses on finding a balance between maximising the chances of mating (fitness function intrinsic motivation) and ensuring survival efficiency as well as energetic autonomy (fitness function extrinsic motivation). This is to ensure the strongest genome is passed on from generation to generation. The mEDEA algorithm is as follows:

The Medea Algorithm:

```
genome.randomInitialize()
while forever do
    if genome.notEmpty() then
        agent.load(genome)
    end if
    for iteration = 0 to lifetime do
        if agent.energy > 0 and genome.notEmpty() then
            agent.move()
            Broadcast(genome)
        end if
    end for
    genome.empty()
    if genomeList.size > 0 then
        genome = applyVariation(select_random(genomeList))
    end if
    genomeList.empty()
end while
```

This algorithm is distributed over a population of agents, so is within all robots in the given population. Though all robotic agents run the same algorithm, they carry different genomes (collection of genes that comprise the robot).[4]

The selection operator randomly chooses a genome from the list of imported genomes (through crossover with other robots). There is no explicit selection bias since its random, however overtime and several generations, the more common a genome is the better the likelihood of selection. The variation operator mutates the genome and the replacement operator replaces a robots current genome with the mutated one (derived from the fittest robots), thus creating new individuals every generation. The figure below and the caption illustrate and explain how the code is applied to simulated robotic swarm.



In (1) at the start of the generation the genomeList is empty. Each robot only contains their active genome which determines its behaviour. As they traverse the environment and come in contact with one another they exchange genomes when they're in close proximity. The red agent genomeList includes the green and blue agents genome from (2) and (3) respectively, whereas the other two agents both contain the red genome from their import list by the end of the generation in (4). The red genome spread more so has a greater likelihood of survival (randomly selected for mutation and passing onto the next generation). Green and blue only have a 50% chance probability of being mutated and carried on or being deleted.

One paper [4] experimented with the mEDEA algorithm using 100 simulated e-puck robots in two environment set ups. One set up in this paper was the "free ride" setup in which environmental pressure was limited so as to focus on the intrinsic motivation of the fitness function, which is to optimise mating. In the "energy" setup the environment was filled with energy resources for the robotic agents to consume and the robots were each given a set lifespan which could be influenced by the environment resources foraged and consumed. This is similar to how living organisms rely on food to survive in the natural world. This also created competition among agents over the limited energy resources in the environment. Agents the didn't forage would have their energy level depleted - and die - faster than the ones that did harvest food from the environment. This would measure how well mEDEA handles the trade-off between mating and surviving and finding equilibrium between the two objectives. The free ride setup was used for the first 75 generations before switching to the energy setup for the next 75 generations.

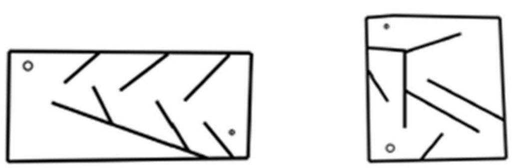
Another paper experimented with 20 physical e-puck robots (all running mEDEA) in an attempt to close the reality gap between the simulations and the real world.

This helped in identifying several issues not otherwise experience in simulations such as unreliable proximity sensors, slow speed of execution and inconsistent exchange of genomes when neighbouring agents collide. This was as a result of a number of reasons to do with hardware used for the experiment. Regardless of the problems listed, mEDEA proved to be rather robust as it displayed behaviours that thrived in the environment. This also demonstrated that smaller population samples rarely converged towards behavioural consensus unlike larger populations [5].

2.1.2 Novelty Search

Novelty Search rewards robotic agents in a swarm for performing a new, unique or unusual behaviour instead of based on an objective fitness functions [2]. This is especially useful when solving problems that have deceptively clear solutions. For example the goal of the Chinese finger trap is to free your fingers. It is ones natural instinct to just want to pull your fingers away from each other (the objective), however this can further entrap your fingers. Only when attempting something uncommon and original (like pushing your fingers closer and further apart) is one able to actually get closer to the true objective. When it comes to working with robotic agents in an swarm intelligence, novelty search aids in providing an additional method of evaluating evolutionary algorithm solutions besides the objective fitness function. It also improves functional diversity in a robotic swarm by encouraging more exploration of the environment [3].

In one paper an experiment was conducted using maze navigation where agents were trained to reach a destination whilst avoiding obstacles. The maze navigation experiment involved using robots to make their way out of the maze to the final destination. Each robot had a built in radar to determine the distance to the closest obstacles, which acted as a compass by guiding the agent towards the goal [1].



(a) Medium Map (b) Hard Map

In each of the figures above, the smaller circle represents the agent and the larger one represents it's destination. Using a straight forward, direct approach to the destination often led the agent into corners and cul-de-sacs preventing it from leaving the maze and reaching the destination [1].

2.2 Examples of Adaptive Swarm Behaviour

Dssds

3 DISCUSSION

In the simulated mEDEA experiment mentioned in the previous chapter, the number of active agents and the average number of imported genomes per generation where used as indicators of minimising deaths whilst optimising mating opportunities. The number of food items stored was also used as a performance indicator in the energy setup. Both the free ride and energy setups showed an increase in mating opportunities and survival rate. Switching setups initially caused a drop in performance indicators, but this was followed by a prompt recovery due to evolutionary adaption. There were some extinctions that occurred in the energy set up for three out of one hundred runs of the experiment. In both setups genomes from later generations displayed a better probability of survival compared to genomes from earlier generations [4].

Analysing the results from the mEDEA experiment involving (physical) e-puck robots, we make a few observations relating to evolving homogenous behaviour, robustness to environmental changes, learning team-work driven behaviour and specialisation through speciation. A concord toward a particular behaviour may be evolved and distributed among the population as both population size and mating opportunities increase. The mEDEA algorithm's adaptability was assessed by changing the swarms environment towards harsher conditions. Even though this negatively affected the number of active agents, we noted that robots running the mEDEA algorithm were able to quickly recover from this by evolving traits and actions tailored to the new environment without human input. mEDEA naturally adapts compassionate behavioural traits in aggressive environments, globally converging towards a balance between generous and selfish actions (like how one may feel charitable when their own needs are taken care of) [5].

With regards to the novelty search maze experiment, in the case of the medium map, novelty search took about a three times faster than the fitness based algorithms. Novelty search required 18.3 thousand evaluations to find a solution, whereas the NEAT objective fitness function required 56.3 thousand evaluations to solve the maze. NEAT with random selection performed the worst, only finding solutions for 21 out 40. For the case of the hard difficulty map, using a fitness-based NEAT only yielded three solutions out of forty runs while NEAT with random selection did very slightly better by finding success four out of forty times. On the other hand, NEAT with novelty search solved the maze in 39 out of the 40 runs. It is clear that Novelty search exhibits a more even distribution of points throughout both mazes Fitness-based NEAT shows areas of density around the local optima. The experiment also compared the fitness and novelty search algorithms in a borderless maze environment. This is a map with no walls. Novelty based Neat only found solutions 5 times out of a hundred runs, pretty much demonstrating that constraining the space of achievable actions in some domains is necessary for the efficient use of novelty search. Regardless, fitness-based NEAT found solutions to the borderless hard difficulty map twice out of 100 computations. This emphasizes the fact that fitness-based objective search doesn't become a more viable alternative just because novelty search is not suitable [1].

4 CRITICAL COMPARISON OF PRIOR WORK

For [5], the physical experiment essentially failed due to several hardware constraints. Despite the author continued to make assumptions about the real world applications of mEDEA based on the simulated experiment results.

The paper [1] exposes some drawbacks of Novelty search such as difficulty defining novel behaviour in relation to completing and achieving a specific task.

5 CONCLUSIONS

This paper reviewed and compared the research done on various adaptive methods for swarm robotics, including the experiments done and observations that were made based on the experiment. Going forward, we aim to hybridise evolutionary algorithms to see if it will optimise the speed and efficiency for a particular task. These tasks would include foraging and collective construction.

REFERENCES

[1] Lehman, J. and Stanley, K., 2011. Abandoning Objectives: Evolution Through the Search for Novelty Alone. Evolutionary Computation, 19(2), pp.189-223.

[2] Gomes, J., Urbano, P. and Christensen, A., 2013. Evolution of swarm robotics systems with novelty search. Swarm Intelligence, 7(2-3), pp.115-144.

[3] Fister, I. et al., 2019. Novelty search for global optimization. Applied Mathematics and Computation, 347, pp.865-881.

[4] Bredeche, N., Montanier, J., Liu, W. and Winfield, A., 2012. Environment-driven distributed evolutionary adaptation in a population of autonomous robotic agents. Mathematical and Computer Modelling of Dynamical Systems, 18(1), pp.101-129.

[5] Nicolas Bredeche and Jean-Marc Montanier. 2012. Environment-driven Open-ended Evolution with a Population of Autonomous Robots. "Evolving Physical Systems Workshop (2012)."

[6] Eiben, A. and Smith, J., 2015. Introduction to evolutionary computing, Berlin [etc.]: Springer.

[7] Floreano, D., Husbands, P. and Nolfi, S., 2008. Evolutionary Robotics. Springer Handbook of Robotics, pp.1423-1451.

[8] Boumaza, A., 2022. Seeking Specialization Through Novelty in Distributed Online Collective Robotics. Applications of Evolutionary Computation, pp.635-650.

[9] Montanier, J., 2013. Environment-driven Distributed Evolutionary Adaptation for Collective Robotic Systems. PhD. Paris Sud University - Paris XI.

[10] Didi, S. and Nitschke, G., 2016. Hybridizing novelty search for transfer learning. 2016 IEEE Symposium Series on Computational Intelligence (SSCI).

[11] Haasdijk, E., Bredeche, N. and Eiben, A., 2014. Combining Environment-Driven Adaptation and Task-Driven Optimisation in Evolutionary Robotics. PLoS ONE, 9(6), p.e98466.

[12] Evolution of a Functionally Diverse Swarm via a Novel Decentralised Quality-Diversity Algorithm

[13] Hyper-Learning Algorithms for Online Evolution of Robot Controllers

[14] Illuminating search spaces by mapping elites

[15] Quality Diversity: A New Frontier for Evolutionary Computation

[16] Swarm intelligence: new techniques for adaptive systems to provide learning support

[17] Swarm Robotic Behaviours and Current Applications

[18] Swarm robotics: a review from the swarm engineering perspective

[19] Task partitioning in swarms of robots: an adaptive method for strategy selection.

[20] Evolution of a functionally diverse swarm via a novel decentralised quality-diversity algorithm

[21] POET: open-ended coevolution of environments and their optimized solution