UNIVERSITY OF CAPE TOWN

DEPARTMENT OF COMPUTER SCIENCE

CS/IT Honours Project
Final Paper 2022

Title: Evaluation of Evolutionary Algorithms for Collective Gathering Tasks in Swarm Robotics

Author: Kevaalin Rapeti

Project Abbreviation: SWARM

Supervisor(s): Prof. Geoff Nitschke and Prof. Deshen Moodley

Category	*Min*	*Max*	Chosen
Requirement Analysis and Design	*0*	*20*	0
Theoretical Analysis	*0*	*25*	0
Experiment Design and Execution	*0*	*20*	20
System Development and Implementation	*0*	*20*	10
Results, Findings and Conclusions	*10*	*20*	20
Aim Formulation and Background Work	*10*	*15*	10
Quality of Paper Writing and Presentation	*10*		10
Quality of Deliverables	*10*		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	*0*	*10*	0
Total marks		**80**	

Evaluation of Evolutionary Algorithms for Collective Gathering Tasks in Swarm Robotics

Kevaalin Rapeti
Department of Computer Science
University of Cape Town
Cape Town, South Africa
RPTKEV001@myuct.ac.za

ABSTRACT

Functional diversity within groups of animals in the real-world has shown to create species that can perform a wider variety of tasks better, allowing for the groups to perform efficiently and ensure a level of adaptability that allows them to overcome new challenges. As such, this is a natural phenomenon that we endeavor to mimic in evolutionary algorithms in swarm-robotics, creating collections of robots that have quality diversity rather than being specialized in a singular task. An evolutionary algorithm will be used to generate new robotic swarms, which will be better adapted to completing their tasks in their environment, compared to the earlier generations, whilst still maintaining behavioural diversity. In order to build this algorithm initially, we will use an evolutionary algorithm called the minimal environment driven evolutionary algorithm (mEDEA) hybridized with a multi-behavior characteristic (multi-BC) multi-dimensional archive of phenotypic elites (MAP-elites), to allow for diversity in during robot phenotype generation.

CCS CONCEPTS

• Computing methodologies

KEYWORDS

Evolutionary robotics, Evolutionary algorithms, MAP-Elites, Swarm robotics, Adaptive robotics

1 Introduction

In nature there exists multitudes of different species, which have developed to survive using a group dynamic. This can be seen very prominently in the insect world as well as in human society, such that sub-groups of individuals learn and specialize at completing certain tasks in order for the group to become more adaptable to changing environments as described by Hamann and Schmickl [1]. Focusing on this concept in insect groups and creating a parallel to it in the world of robotics, we come to a definition of swarm robotics. As described by Brambill, et al [2] swarm robotics is an approach to collective robotics that takes inspiration from the self-organized behaviours of social animals.

It is due to these behaviours that swarm robotics stands apart from standard robotic practices and why we have taken an interest in the possible capabilities of swarm robotics. As stated by Spezzano [3], one of the objectives of swarm robotics, is to create a group of organisms, where the group exhibits group behavior which allows for it to complete tasks collectively which would be impossible to accomplish for any of its individual members. An additional benefit of using swarm robotics is based on their ability to be specialized for functional diversity, allowing the swarm to accomplish more diverse tasks, allowing for more robust usage.

An important aspect of swarm robotics is swarm intelligence which as stated by Kennedy [4] can be described as the problem-solving ability derived from the interactions of simple information-processing units but is also described by Dorigo and Birattari [5] as the discipline that deals with natural and artificial systems composed of many individuals that coordinate using decentralized control and self-organization. Based on these two definitions we can see that the key aspects of swarm intelligence depend on the interactivity held between robots in the swarm, done by interactions between individuals which can range from simple to complex. Such as communication about the environment between robots, validation of an entity in the swarm of message passing.

A swarm can be heterogenous as well as homogenous, allowing for the swarm itself to have a higher level of complexity as described by Hamann [6], but with a higher level of complexity comes the challenge of proper functionality of the swarm to achieve its goals. A balance between complexity of the robot swarm and reliability of its functionality needs to be established. In order to achieve this level of functional diversity and swarm complexity we will need to use evolutionary algorithms, to generate new swarms based on data from previous generations, allowing us to eventually create a generation that is better fitted to its environment and the objectives found in it.

Because of this, swarms need to be designed and implemented with optimization algorithms in mind for the swarm, so that the swarm intelligence of future generations is more functional, efficient and better adapted to its environment than that of previous generations. We can refer to the impact of swarm intelligence-based optimization algorithms from the work done by Yang, et al, [7] and the impact of swarm intelligence-based optimization algorithms and self-organization which is examined by Yang, et al [8]. From this we can determine how best to handle swarm intelligence optimization algorithms in order to ensure that as we generate new generations of the swarm we will be able to converge on collective behaviours which will be best suited for their environment.

This proper handling of swarm intelligence is what can help us to design the behavioural characteristics of the individual robots as well as the swarm. This design characteristic for the overall swarm can also be called their collective behaviour, in the past there has been collective behaviours designed to handle already know environments where there was isolated reproduction of each new generation as described by Hart, et al [9]. This helped create a generation of robots suited to the environment, however, required too much external influence and is not suitable to implement in real-world scenarios where such influence may not be possible. So, we look then to our evolutionary algorithm, namely a hybridized version of the minimal environment driven evolution algorithm (mEDEA) and the multi-behaviour characterization (multi-BC) Map-Elites algorithms. In order to let the robots, become an efficient and adaptive swarm.

In the attempt to create this efficient and adaptive swarm, we will need to look into adaptive collective behaviour (ACB) to see how it can positively influence the functionality of a robotic swarm along with its quality diversity. ACB has been exhibited in the natural world many times, found in swarm animals and insects alike as described by Feng, et al [10]. They also state that with regards to emergent behaviour of the swarm, these become easily apparent at a population level, but not so prevalent at an individual level. What we will need to take away from this fact, is that when designing how the swarm will act as an entity, the controller for individual robots will be designed with this in mind, and we will need to ensure that good controller design can allow for the emergent behaviour to appear at the population level.

Although the field of swarm robotics is still relatively young, there is a growth of interest in the real-world application of swarm robotics, due to some of the capabilities of it mentioned before. There are many different collections of swarm robotics research platforms, projects and product as mentioned by Schranz, et al [11], from this we can find terrestrial, aquatic, aerial and outer space applications for swarm robotics. An example of these applications include the work done by Kettler, et al [12], in which their work with the Wanda robot framework, which could be deployed in environmental clean-up projects, as stated by Schranz, et al [11]. Another examples of these applications could be in mining, where a functionally diverse swarm would be needed to manage the different tasks found in a mining operation, such as digging, exploring and collecting the mined resources, Pilania and Chakravarty[12].

1.1 Research Objectives and Contributions

Based on these findings, on the collective behaviours of swarm robotics and their ability to complete collective tasks, we want to test the effectiveness of different ways of evolving collective behaviour in robot swarms

- Objective 1: To test the effectiveness of using a hybridised MEDEA and MAP-Elites evolutionary algorithm versus MEDEA in evolving the collective behaviour of a robotic swarm, evaluated over collective behaviour tasks of increasing complexity.
- Hypothesis 1: Robotic swarms evolved using the hybridised MEDEA and MAP-Elites algorithm will produce higher performance compared to robotic swarms evolved using a base MEDEA algorithm.

The reason selecting these evolutionary algorithms and techniques is due to previous studies done, one which we base some of the work of this project. Namely, the work done by Bredeche and Montainer [13] in developing the MEDEA algorithm which give us a way to test and adapt robotic swarms to unknown environments and the work

done by Hart, et al [9] which focused on developing a novel quality-diversity algorithm using MAP-Elites hybridised with the MEDEA algorithm. From both of these papers, we will extend on them by testing the collective behaviour of a swarm in task environments of increasing complexity.
By testing this objective, we will be able to draw conclusions on whether it is worth investing more research into behavioural characteristic MAP-Elites in the field of swarm robotics.

2 Related Works

2.1 Literature Review

In developing this research paper, previous works and their conclusions have been researched, to determine some assumptions and uncover which aspects are important to consider when designing the experiment and when the experiment is carried out, such as:

A hybridised MEDEA and MAP-Elites algorithm is more effective than a base MEDEA algorithm at quality diversity. The paper by Hart, et al [9], in which we can look into creating a functionally diverse swarm of robots. In the paper, they explored the use of different variations of a hybridised MEDEA and MAP-Elites algorithm, testing how well the algorithm would allow for quality diversity at an individual level whilst developing different behaviours across the swarm. They compared four variations of this algorithm to a base MEDEA algorithm, and concluded that the hybridised algorithm, in all cases, does produce more varied quality diversity in the swarm. From this conclusion, it has been decided to further investigate the use of this quality diversity algorithm to more complicated task environments. From this paper, the algorithm variant which will be used is the EDQD-M3 variant, which was shown to generate a high number of varied genomes within the swarm.

A more complex task environment is needed. The paper by Bredeche and Montainer [13], this is the original paper which looked into developing the MEDEA algorithm. In their experiments they tested the swarm in a task environment which focused on “Energy Collection”, individuals were only tasked with collecting resources to allow them to continue “living” in their environment, without enough energy they would deactivate. In the works of Hart, et al[9] they used a similar task environment to test their findings, showing their results stand for a simple task, but not for more complicated ones, that is why a more complicated task environment needs to built to test the effectiveness of the algorithm at higher complexity levels.

Well-designed metrics for the MAP-Elites. The paper by Mouret and Clune[14], discusses how MAP-Elites can be used to identify different high-performing solutions in a search space. Taking a look at the MAP-Elites algorithm from this paper, we can see that it is designed to two values, the first being a performance score, which is determined by the user who is designing the algorithm, the second is an N-dimensional value which is called the *feature space*. In the feature space the user will determine which variations of interest they want to be recorded by the algorithm. Once is has been designed and implemented properly, the MAP-Elites algorithm will search for high performing solutions, based on first value stored, across many different feature spaces, the second value stored. By doing this, it allows for the user to pick high-performing solutions that have significant variation between them, allowing for higher degrees of variation in the final solution of an evolutionary algorithm. This paper will be used to understand how to design proper MAP-Elites algorithms and then it shall be implemented into the hybridised algorithm used in the experiment.

Good Collective behaviours. The collective behaviour of a swarm is very important, since the ability of the individuals being able to work cohesively for the overall objective is the determining factor of a good swarm. As seen in Figure 1, we can see a list of different collective swarm behaviours which will be needed in designing the swarm for the experiment. From this list we can see four main sub-sections: Spatial Organization, Navigation, Decision-Making and Miscellaneous. For this project, the Miscellaneous category shall not be focused on, however the three other categories do focus on behaviours that are important. Such as *Collective Transport* and *Coordinated Motion* under the Navigation sub-category, since for this experiment the performance of the robots does depend heavily on their ability to transport the resources in the environment both individually and collectively.

2.2 Conclusions on Literature Review

From these research papers we are able to draw on some conclusions which will help determine how the experiment should be designed, what characteristics we are looking for in the integrated solution and what aspects are missing from previous works that should be addressed in this paper.

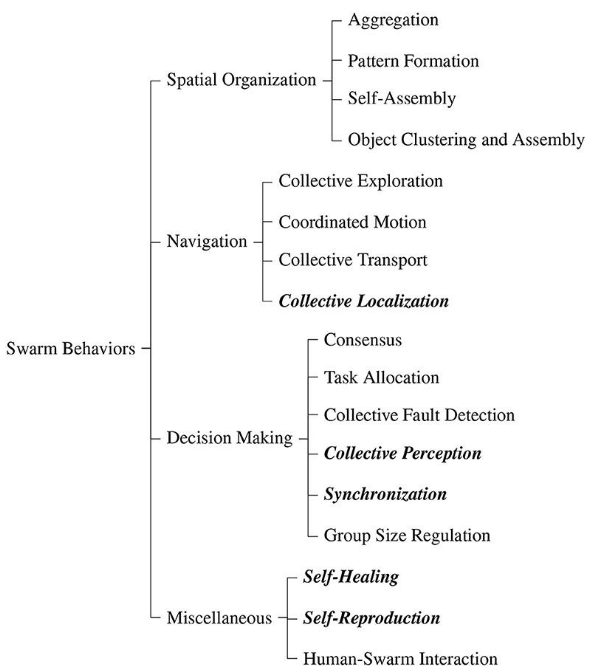

Figure 1: A list of swarm behaviours and their sub-categories. Schranz, et al [11]

From the paper by Hart, et al [9], that a hybridized MEDEA and MAP-Elites algorithm does produce higher variation in robots later in future generations of a swarm. However, in their paper, they had the robots conducting simple tasks which did not test their collaborative abilities and rather focused on creating solutions of well-performing individual robots instead of well-performing collaborative robots. This paper will compare how well those higher variation solutions perform in collaborative tasks.

There is also high emphasis on the collective behaviour of the individuals in the swarm and so this will need to be properly managed in the experiment, such that the collective behaviour of the swarm will allow the experiment to produce results which will be able to show the benefits of having chosen swarm robotics to complete the task as opposed to more traditional means of robotics.

3 Methods

Based on the related works and research objective laid out in order to uncover the results of the experiment the following methods shall be followed.

3.1 Approach

To carry out this experiment an existing 2-D robot simulator called roborobo created and maintained by nekonaute [15], with it two different robot controllers will be designed and used to run the experiments. The first controller is designed to carry out the MEDEA algorithm, with the second algorithm being the hybridized algorithm.

When designing the algorithm, as mentioned the variant used to generate new maps at the start of each robot lifetime is the EDQD-M3 variant, the procedure it uses to generate new maps is by taking the list of ReceivedMaps and merging it with its own LocalMap to form a SelectMap, a genome from this map is then selected at random, which is then used to update the LocalMap of that robot, Hart, et al[9].

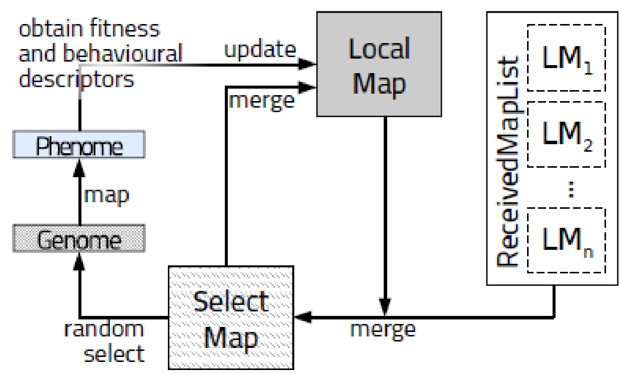

Figure 2: A visual representation of how the EDQD-M3 algorithm work, Hart, et al[9]

During a simulation, each robot will start with a basic genome which will be randomly allocated to each robot during its construction phase. Each simulation will consist of multiple generations of the swarm. Between each generation is when the controller of each robot will be updated.

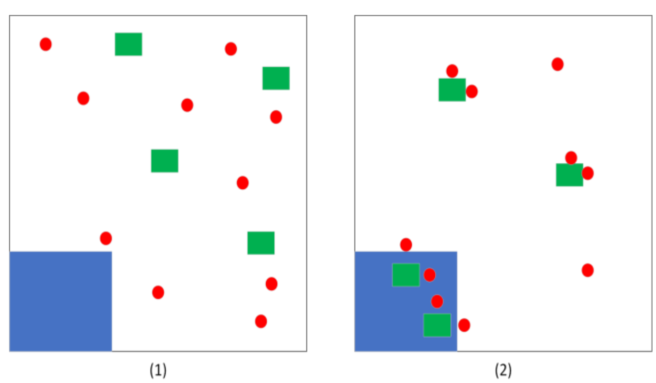

Figure 3: Collective foraging represented by agents (red) searching the environment for resources (green) and

depositing them in the target area (blue). Figure by Kevaalin Rapeti

3.2 Robot Controller Design

First, the MEDEA controller will be designed and tested to make sure it is able to complete the collecting task. The controller will be configured to identify different objects in its environment, these objects could be either borders, other robots, or the resources they need to collect. The controller will explore the environment attempting to locate an uncollected resource, once found it will attempt to transfer the resource to the collection zone. In cases where more than one robot is required for the task, refer to Table 1, the controller will attempt to wait for additional robots to arrive and move the resource. If another robot does not arrive soon, the controller will tell the robot to abandon the current resource and try to find another one so it does not waste time stuck to a resource and reduce the efficiency of the swarm.

	# of Resource A	# of Resource B	# of Resource C
Test Environment 1	75	0	0
Test Environment 2	50	25	0
Test Environment 3	30	30	15
Number of robots to move resource	1	2	3

Table 1: Table showing the composition of the different resource types across the three different environmental configurations

The way in which the controller handles being stuck on a particular resource is by waiting a predetermined number of steps in the simulation before following a "abandon" heuristic. The robot will check its current position against its position in the previous step and if this value is the same for 50 steps in the simulator the robot will leave its current assigned task, whether that is exploration or collection and move itself away from its current position. This was robots that are trapped trying to move a resource that cannot move or being trapped by other obstacles are able to reinitialize and then carry out its normal functions. This is most apparent in the case of a robot trying to move a resource of type B or C, which require multiple robots to move. Instead of a robot being stuck trying to move this object by itself, it will instead abandon the current resource if no other robot comes to help it so that it can use its time efficiently in the simulation.

3.3 Fitness Function

In order to determine which robots are performing well in the simulation, and thus which genotypes to prioritize and pass on to future generations of the swarm, a fitness value will need to be calculated in each robot during its lifetime. The process of calculating the fitness value of the robot will be based on different metrics which complement the robots resource collection ability. Namely, the *robot displacement from its starting point and the distance it has moved a resource*. These two metrics are chosen because they promote both the ability of a robot to explore its environment and the ability it has to transfer resources across this environment. The fitness function f(x), is defined below:

$$f(x) = \frac{TD}{MD} * 0.4 + \frac{RD}{TD} * 0.6, \qquad (1)$$

Where TD is the *total displacement* of a robot from its starting position, MD is the *maximum displacement* a robot can move based on the size of the environment and RD is the *resource distance* which states how far a robot has travelled whilst pushing a resource. This will produce a value between 0 and 1, where the higher the value of the fitness function the better performing the robot is. The function has been weighted such it values transporting resources more than general exploration, as shown by the higher weighting scores.

4 Experiment

In the appendix is a table discussing the breakdown of the different levels of complexity found in the task environment. For the collective gathering task, we have broken down the tasks into three levels of increasing complexity, based on the resources that can be found in the simulation. Resources are divided into three categories, levels A, B and C. Where depending on the category, it determines the number of robots required to collect the resource. Resource A requires a single robot to move, Resource B requires a minimum of 2 robots and Resource C requires a minimum of 3 robots to move the resource.

4.1 Experiment Design

The experiment shall test the performance of the MEDEA algorithm against the hybrid algorithm for collect gathering tasks of increasing complexity. The experiment will test the capabilities of the swarm at locating and transferring resources in the environment to a predefined target area, as shown in figure 2. The changing complexity will be done adding resources into the environment that require multiple robots in order to move in the environment, allowing for the collaborative ability of the swarm to be tested.

Metrics that will be collected in the experiment are: average movement of each resource type by the swarm in a swarm lifetime towards the target area, which will be the main way to compare the performance of the two algorithms; the distance an individual robot has moved resources, which will be used to select which maps will be used to generate new maps between robot lifetimes, and the displacement of a robot from the current lifetime's starting position by the end of each lifetime which will be another swarm-level metric which we can use to compare the algorithms.

4.2 Environment Design

The experiment shall be done in a virtual environment, this virtual environment is created from the roborobov4 simulator, which is project done by Nekonaute[15]. The experiment shall occur in the same environmental conditions for all three complexity levels, which will be 2 dimensional, flat environment, which shall be call the testing arena. There will not be any obstacles spawned in the arena, meaning robots will get stuck on obstacles or be forced to maneuver around these obstacles. The environment has a pre-determined collection zone, in which the controllers will transport the resource to. The resources themselves are all spawned at the beginning of the simulation and are randomly distributed around the environment.

4.3 Resource Design

Resources can are designed to be one of three different categories, type A, B or C. Type A resources are designed to be moved by any number of robots, whereas type B requires 2 robots and type C requires 3 robots to move in the environment, as shown in Table 1. This design forces the robots to work collaboratively in order to move them, and has been designed as such so that robot controllers with better collaborative behaviours will emerge in future generations.

4.4 MEDEA Design

The base algorithm which runs MEDEA, will run with no form of fitness function, instead it will pass genomes to future generations based purely on the occurrence of any genome in the list of received genomes from other controllers.

4.4 MAP-Elites Design

The hybridized algorithm works by creating a MAP-Elite for each robot controller. These MAPs are designed with a performance value, which is the fitness function described in Section 3.3, and then a 2-dimensional vector which will store data on the displacement of a robot from its starting position and the distance the robot has moved resources.

Table 2: MEDEA parameters

Parameters	Values
Population Size	100
Simulation Lifetime	40200
Generation Lifetime	400
Number of Generations	100
Number of Sensors	8
Fitness Function	None

Table 3: MEDEA and MAP-Elites parameters

Parameters	Value
Population Size	100
Simulation Lifetime	40200
Generation Lifetime	400
Number of Generations	100
Number of Sensors	8
Fitness Function	Function 1 (Section 3.3)
Feature Space Values	
First dimension	Robot displacement
Second dimension	Distance resources moved

5 Results and Discussion

5.1 Experiment Results

We have tested this experiment in the three different task environments and in each task environment we ran both the MEDEA algorithm and the MEDEA-MAP Elites algorithm, in order to determine the effectiveness of each evolutionary algorithm at the collective gathering task. Each simulation was run 10 times and their results were averaged and represented below.

For the first task environment where only resource type A is used, the robots have the lowest incentive to work collaboratively and so robot variations which move more resources and traverse the environment better have higher performances than any which exhibit collaborative behaviours. For the MEDEA algorithm the average number of generations it took to collect all resources was 54 and each generation had on average moved the type A resources a distance of 39 units in the 600x600 unit arena. For the hybrid algorithm, the average number of generations taken to collect all resources was 50 with an average movement of type A resource of 44 units

In the second environment we included the second resource type. With regard to resource of type A we experienced similar results to those from task environment 1. With MEDEA performing with the average generations to collect resources being 55 and average movement per generation being 38 units, and the hybrid being done on generation 52 and a distance of 43 units. The difference for resource B being that MEDEA took much longer to collect all resource, averaging 62 generation and a distance of 25 units and the hybrid algorithm averaging 57 generations and a distance of 28 units.

In the third environment, we experienced similar results for resource type A and B to those found in task environment 2. However the significance lies with the averages recorded for resource type C. In the MEDEA algorithm, we found that on average it took approximately 90 generations for all type C resource to be collected in the target zone, andon average these resources were only move 14 units per generation. Compared to the hybrid algorithm, which took on average 75 generations to collect all type C resources and moved them on an average of 15 units per generation. However, it should be noted that these averages are affected by low recorded scores in the initial generations.

5.2 Discussion of Results

Based on the results from the experiment we can see that on average, as the task environment complexity increases, the efficiency of the solutions do decrease at the beginning. However, we can see from the results that the hybrid algorithm does perform slightly better than the MEDEA algorithm by itself.

We also can note that in the higher complexity task environments, TE3, compared to the lower complexity task environments, the swarm is able to gather the simpler resources, type A, in line with the results from those in task environment 1 and 2. It slows down performance for collecting resources type B and C, since the initial controller is not well-designed for collaborative tasks, but does eventually evolve to being better at collaborative work.

6 Conclusions

In conclusion, the hybrid MEDEA and MAP-Elites algorithm did outperform the base standard MEDEA algorithm, however, it had not outperformed it by a significant margin. This better performance can be accredited to the design of the MAP-Elites used to encourage the evolution of more variation in the swarm at later generations

I believe that there are improvements that can be made to the experiments and research I have done, changes to the robot controller design as well as more complex iterations of the MEDEA and MAP-Elites algorithm which could possible produce better results and that this future research should be conducted to explore the capabilities of it as an evolutionary algorithm for swarm robotics.

7 Acknowledgements

Simulations and experiments were run using the Centre for High Performing Computing cluster.

References

[1] Hamann, H & Schmickl, T. (2012). Modelling the swarm: Analysing biological and engineered swarm systems. Methods Tools and Applications in Engineering and Related Sciences. 1-12. DOI: https://doi.org/10.1080/13873954.2011.601426

[2] Brambilla, M., Ferrante, E., Birattari, M., & Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 1-41. DOI: https://doi-org.ezproxy.uct.ac.za/10.1007/s11721-012-0075-2

[3] Giandomenico, S. (2019). Swarm Robotics. Rende. Italy. DOI: 10.3390/books978-3-03897-923-4

[4] Hamann, H. (2018). Swarm Robotics: A Formal Approach (1st ed.). Brussels. Belgium DOI: https://doi-org.ezproxy.uct.ac.za/10.1007/978-3-319-74528-2

[5] Kennedy, J. (2006). Swarm Intelligence. *Handbook of Nature-Inspired and Innovative Computing*, 187-219. DOI: https://doi.org/10.1007/0-387-27705-6

[6] Dorigo, M & Birattari, M. (2007) Swarm Intelligence. Scholarpedia. DOI: doi:10.4249/scholarpedia.1462

[7] Yang, X; Cui, Z; Xiao, R; Gandomi, A H; Karamanoglu, M. (2013). Swarm Intelligence and Bio-inspired Computation: Theory and Applications. Newnes

[8] Yang, X.-S., Deb, S., Zhao, Y.-X., Fong, S., and He, X. (2017). Swarm intelligence: past, present and future. *Soft Comput.* 22, 5923–5933. doi: 10.1007/s00500-017-2810-5

[9] Hart, E., Steyven, A. S., & Paechter, B. (2018). *Evolution of a Functionally Diverse Swarm via a Novel Decentralised Quality-Diversity Algorithm.* Kyoto: GECCO. DOI: https://doi.org/10.1145/3205455.3205481

[10] Feng, C; Gribaudo, M & Hillston, J. (2015). Performance Analysis of Collective Adaptive Behaviour in Time and Space. Electronic Notes in Theoretical Computer Science. 53-68. DOI: https://doi.org/10.1016/j.entcs.2015.10.019

[11] Schranz, M., Umlauft, M., Sende, M., & Elmenreich, W. (2020). Swarm Robotic Behaviors and. *Frontiers in Robotics and AI,* 1-20. DOI: https://doi.org/10.3389/frobt.2020.00036

[12] Pilania, V. K and Chakravarty. (2008). Application of Wireless Visual Sensor for Semi-Autonomous Mine Navigation System in World Academy of Science, Engineering and Technology. DOI: doi.org/10.5281/zenodo.1071398

[13] Bredeche, N., & Montanier, J.-M. (2010). Environment-driven Embodied Evolution. *Parallel Problem Solving From Nature*, 1-28. DOI:

[14] Mouret, J.-B., & Clune, J. (2015). Illuminating search spaces by mapping elites. *arXiv*, 1-15.

[15] Nekomaute (2022). Roborobo4. URL: https://github.com/nekonaute/roborobo4

Code

Code Repository: https://github.com/Kevaalin/SWARM