Evolving Problem-Solving Collective Behaviour and Functional Diversity in Swarm Robotics

Project Proposal

Bailey Green
University of Cape Town
GRNBAI001@myuct.ac.za

Sihle Calana
University of Cape Town
CLNSIH001@myuct.ac.za

Kevaalin Rapeti
University of Cape Town
RPTKEV001@myuct.ac.za

1 PROJECT DESCRIPTION

Swarm robotics is a complex and constantly evolving field. It is an application of swarm intelligence which follows the behaviours of a group of interacting agents. Swarm intelligence often takes inspiration from nature and applies it to artificial systems [8]. Swarm robotics looks at the collective behaviour of a group of robots and tries to optimize the behaviours and interactions between robots as well as between robots and the environment to perform some task. These robots can either be physical robots in a real environment or simulated robots in a virtual environment. For this project we will be using virtual robots in a simulated environment [13, 14]. This project will investigate various methods for evolving problem-solving collective behaviour and functional diversity in swarm robotics. We will attempt to hybridize existing and previously researched techniques, which will hopefully be able to add to the field of swarm robotics and possibly find new methods for solving tasks using swarm robotics in an effective way where other techniques may be lacking.

2 PROBLEM STATEMENT

Solving complex problems using swarm robotics is a difficult task and there is still lots of ongoing research in this field. Depending on the problem you are trying to solve, there may be different solutions that best fit that problem. The three big areas when looking at methods for implementing swarm robotic systems are, being adaptable to unknown or changing environments, being able to achieve a high max fitness value to be able to solve the task effectively and having a diverse swarm. There are many methods each with their own strengths and weaknesses which can be used to achieve these goals. In this project we aim to find hybridized methods that attempt to address these three goals and further improve upon existing solutions.

Being able to adapt or react to unknown environments is important for swarm robotic systems. There are many techniques to deal with this and we will be using environment-driven distributed evolutionary adaptation (EDEA) [2]. The algorithm for this is known as the mEDEA algorithm and is discussed in section 3.2. mEDEA will act as the baseline for the three hybridized methods we will be testing.

Being able to solve the task in an efficient and effective way is the next part of the problem we will look at. Objective-based functions are a common way of achieving this. They work by having a fitness function which defines how well an agent is achieving a certain task. The higher the fitness of an agent the better the solution to the problem. The fitness function is then used to select and reproduce agents the agents with the highest fitness at the end of a generation [9]. While this method works to obtain high fitness values it has the common problem of getting stuck in a local maximum as seen in [5]. This is where the algorithm will too quickly converge on a particular solution that may be good but is not the overall best solution. This is due to objective-based methods generally producing a low diversity of solutions.

We are aiming to not only find solutions with high fitness values but also a high functional diversity. This is where behaviour-based methods come in. Behavior-based search uses the behaviors of the robots in a swarm as a control rather than an objective. A behavior could simply be obstacle avoidance or exploration. Behaviour-based methods produce a higher functional diversity and solve the problem of getting stuck in a local maximum. The two main behaviour-based methods we will be using are MAP-Elites, section 3.2.1, and Novelty Search, section 3.2.2. Additionally novelty search will be integrated with local competition to see if competition among robots of a similar type has a positive impact on the results, section 3.2.3.

Using mEDEA as a base environment-driven algorithm we will then be implementing three hybridized methods using the three behaviour-based methods mentioned earlier in this section. These three hybridized methods are our research objectives, shown in section 2.1. With each of these methods, we aim to evolve problem-solving collective behaviour and functional diversity in the field of swarm robotics.

2.1 Research Objectives

The primary research objective for this project is determining the best method for evolving problem-solving collective behaviour and functional diversity in swarm robotics. This will be divided into three sub-tasks:

(1) To hybridize the minimal Environment Driven Evolutionary Algorithm (mEDEA) with a multi-Behavior Characterization (multi-BC) Map-Elites algorithm, section 3.2.1.

(2) To hybridize the mEDEA algorithm with Novelty Search, section 3.2.2.

(3) To hybridize the mEDEA algorithm with Novelty Search and Local Competition, section 3.2.3.

As a secondary objective, if we are able to properly implement, test and compare the data of this experiment, we will try and expand on our research goals, by making improvements to how we simulate the swarm and the complexity of the tasks we assign it. Namely, taking it from a 2-D model and represtning in a 3-D environment.

3 PROCEDURES AND METHODS

3.1 Minimal Environment Driven Evolutionary Algorithm (mEDEA)

mEDEA, a minimal environment-driven distributed evolutionary adaptation algorithm, is the base algorithm that we will be hybridizing in all three research objectives. The mEDEA algorithm provides a way of dealing with complex changing or unknown environments. As shown by the pseudo code in Algorithm 1 below, this algorithm is implemented by having a genome which is controlling an agent throughout a generation. Initially this genome is randomly initialized. Each generation the genome is loaded onto the agent and as long as that genome is not empty and the agent has energy, the agent will move around the environment and broadcast its genome to other agents in a certain proximity as shown in figure 1. At the end of each generation the current genome is emptied. There are then three operators that are then used to find a new active genome for each agent. The selection operator finds a random subset from the list of genomes the agent has received from other agents. The variation operator then applies some mutation to these genomes in order to modify them slightly. Usually, this modification is very minor. Finally, a replacement operator is used to randomly select the new active genome to control the agent for the next generation and the list of genomes the agent received is emptied [2]. While there is no explicit fitness function defined when using this algorithm alone, the number of active agents and the average number of imported genomes per generation can give an idea of the resulting performance.

Algorithm 1 The MEDEA Algorithm [2]

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

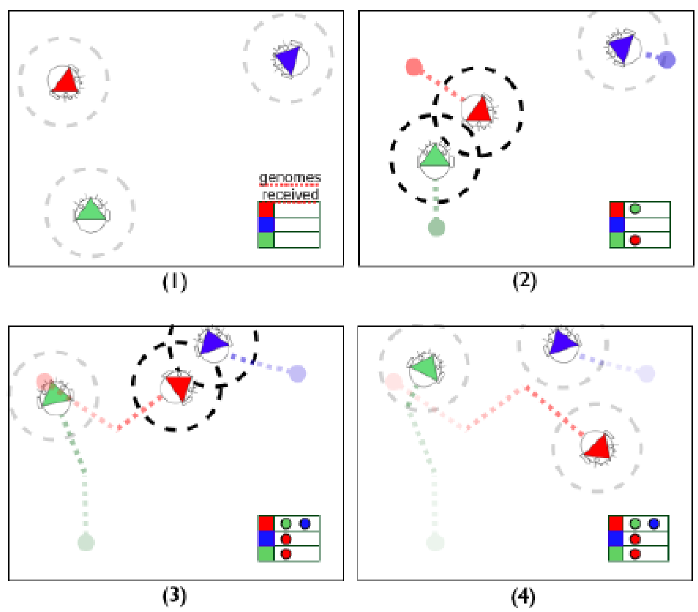

Figure 1: Graphical representation of how agents interact with their environment in the MEDEA algorithm, and how each agent stores the genome information of other agents [2]

3.2 Quality Diversity Algorithms

Although mEDEA is very well-designed for adapting a robotic swarm to unknown or changing environment, it is not as effective at adapting the swarm with regards to the quality diversity and functional diversity of the robots. In the original paper discussing mEDEA [2], the evolutionary algorithm encouraged genomes that were effective at a simple task and eventually converges with all robots in the swarm having this genome or a variation of it. For higher complexity tasks, it would be better that the swarm has a higher level of functional diversity as this will allow us to simulate more complex swarms and systems. In order to achieve this, we will be extending mEDEA, with different algorithms to improve on their quality diversity. These being the multi-behavioural characteristic (multi-BC) MAP-Elites, Novelty Search and Novelty Search with Local Competition.

3.2.1 MAP-Elites

MAP-Elites is a quality diversity algorithm, a type of algorithm that focuses on creating a diverse collection of high-quality behaviours. MAP-Elites work by the user first determining a set of performance measures by which we can determine the design of the robots. The second step required is to choose N-dimensions of interest by which they would want to evaluate the robot, this is described as the feature space. An example of how this would work is that from the N possible dimension, the first could be some physical characteristic of the robot that would change either robot to robot or generation to generation, such as the number of sensors found in the robot. The feature space could be populated either with simple features or more complex features based on other features in the feature space. These complex features will allow us to design fitness tests which will based on a robots design, determine how suitably adapted it is for its environment. And from this we can create newer generation which would be best adapted for their environments.

3.2.2 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 [5]. 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 [16].

3.2.3 Novelty Search with Local Competition

Competition is a powerful technique that is often used in artificial applications. A popular example of this is the study done in which hide and seek was played between two teams of agents, competing against each other [1]. We will be adapting this technique for use in one of our research objectives. As described in [10] there are two types of competition, local competition and global competition. In local competition only agents of a similar type compete against each other whereas in global competition all agents compete against each other. For this project local competition will be combined with novelty search as the base. The method will work by first coming up with a space of niches. This will be used to determine which individuals are nearby in niche space. Competition will then happen between these individuals nearby in niche space. This technique will mean the search and selection will become less global based and more based on the fitness per niche. Novelty search in combination with local competition should allow a large diversity of agents while still getting the best performing agents in each niche.

3.3 Method Hybridization

In order to hybridize each algorithm, first we will need to implement mEDEA in the simulator. Using the pseudocode description of the algorithm 1 from [2] as a guide on how to implement mEDEA. We will run simulations of mEDEA in our environments to see how well it was implemented in the simulator and to gather data and results as a base case to compare the hybrid algorithms.
Then we will proceed to derive the new algorithms, first designing what new features will be required for each hybrid algorithm and how to integrate them properly with both mEDEA and the RoboRobo simulator. After designing we can move on to building each algorithm and then deploying them in the simulator.

3.4 Task Environments

Each of the methods we will be hybridizing will be tested and compared to the base mEDEA algorithm using two tasks. The two tasks are a collective gathering task and a collective construction task. Both of the tasks will also have three levels of complexity. In the collective gathering task the virtual environment will be populated with objects which the robots must then move to a certain area. In the lowest complexity level there will only be one type of object which only requires one robot to move. As the complexity increases additional types of objects will be added that will require more than one robot to work together to move the object. For a comprehensive table of each experiment, their evaluation metrics and task list, look at Appendix C.

3.4.1 Collective Gathering

Collective gathering tests the ability of the robot swarm to find resources in their environment and then move these resources across the environment to pre-determined target areas to deposit the resources. The learning objectives of this case study is to test swarm behaviours of the robots, namely looking at navigation and decision-making behaviours. This case study will allow us to evaluate how well the swarm can navigate an unknown environment, locate resources, transport resource and work co-operatively to complete the task.

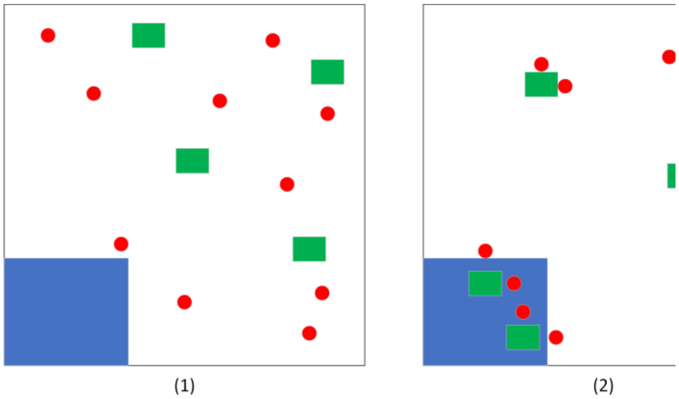

Figure 2: Collective Gathering represented by agents (red) searching the environment for resources (green) and depositing them in the target area (blue).

3.4.2 Collective Construction

The collective construction task environment is an extension of the collective gathering task environment. Instead of just gathering resources in the environment and placing them in the target area, we want to also place them in a specified orientation to form a structure. This test environment shall help us further evaluate the collective behaviours of the swarm as a higher level of navigation and collective decision-making.

It should also be stated that the collective construction task environment will only be attempted if we are able to implement the collective gathering task environment and it works properly.

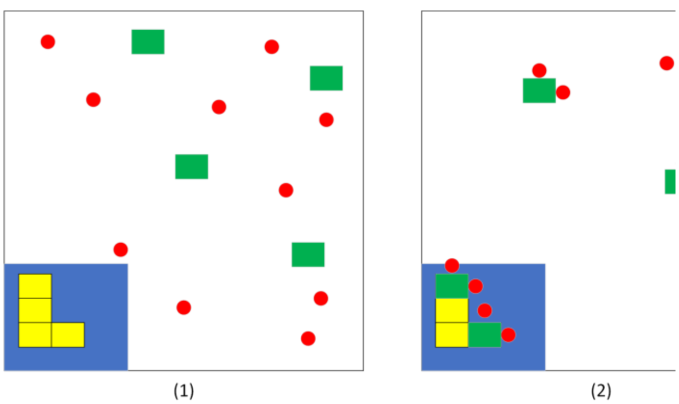

Figure 3: Collective Gathering represented by agents (red) searching the environment for resources (green) and depositing them in the target area (blue) in a specified orientation (yellow).

3.5 Evaluation Metrics

We will need to evaluate both individual capabilities of robots in the swarm and the overall capabilities of the swarm to complete its task.

3.5.1 Individual Evaluation

Individuals will be evaluated on their ability to explore the environment and how effectively they pass their genome on to other individuals in the swarm. Exploration is relevant due to robots not knowing where the resources are in the environment, being able to search efficiently and find the resources will allow for the task to be completed quicker, thus expanding fewer resources. We evaluate how effectively the robot can pass on its genome since this will determine if variant of this genome will actually be seen in future generation of the swarm.

3.5.2 Swarm Evaluation

As a collective the task solving of the swarm needs to be tested to show that the evolutionary algorithm is working in evolving a swarm that is better adapted to a new environment, as well as wanting the swarm to efficiently complete its task. We can measure this with a fitness test that will value two variables. The first being the time taken to complete the task, where time taken will have an inversed relation to the score on the fitness test, and secondly the displacement the resources have been moved to the target area from their original spawning point. This will have a positive correlation to the fitness test.

3.6 Analysis of Results

The two main metrics we will be looking at in the results are fitness values and functional diversity. To measure the fitness values a fitness function will need to be defined which will differ depending on the task done. Functional diversity could be a slightly more challenging metric to accurately measure. It will most likely also depend on the task being done as well as the method being used. We will then compare each of the three hybridized methods against the base mEDEA algorithm to see how the performance compares. An independent t-test will be done to make this comparison.

4 ETHICAL, PROFESSIONAL AND LEGAL ISSUES

This project will have no human involvement and so poses no ethical issues in this respect. We will not be using or modifying any third-party data for this project either. We will be using third-party software to implement our algorithms in a 2-D simulator, this simulator is called RoboRobo and is opensource software. As such we will adhere to the open source software guideline [6] to make use of the RoboRobo simulator in an ethical manner. As an extension of our project, only given that we achieve our primary goal sufficiently in advance of the deadlines we have placed for ourselves, we may consider using another third-party software called Robogen, to simulate the algorithms and robots in 3-D. In this case, we will follow the same guidelines we used for RoboRobo, as Robogen is also opensource software.

5 RELATED WORK

5.1 MAP-Elites

There has been research in how MAP-Elites are usable and preferred as search algorithms. Namely we can look into the work done in [12] which looks at using MAP-Elites as a better form of search algorithm. The justification of this is that MAP-Elites allows the user to determine which features or characteristics they want to focus on and then create an N-dimensional map of vectors which they could populate with values from the different features. This would then allow the user to search for a specific vector and its data in a complexity of O(1).

There has also been work done in researching the effectiveness of MAP-Elites as a method of quality diversity in evolutionary algorithms, such as in [7]. In this paper, they look at the use of MAP-Elites as the quality diversity algorithm, in order to generate high-quality behaviors in the swarm.

5.2 Novelty Search

There has been similar research into the efficiency of Novelty Search compared a more objective based approach. In one such paper an experiment was conducted using maze navigation and biped robots. In these experiments agents where trained to reach a destination or learn to walk without falling over. The maze navigation experiment involved using robots with built in sensors to reach a particular destination. In the figure 4 below, the smaller circle represented the agent and the larger one represented it's destination.

Figure 4: showing maze naviation task [15]

Using a straight forward direct approach to the destination often led the agent into corners and cul-de-sacs preventing it from reaching the goal [15]. In the case of the hard map, seemingly moving further away from the goal at certain parts of the maze actually improved the chances of reaching the objective, much lot like the Chinese finger trap mentioned before. Where the medium map is concerned, novelty search was found to be 3 times faster than the direct approach.

The paper did expose some drawbacks of Novelty search such difficulty defining novel behaviour in relation to the goal or how novelty search may struggle in some unconstrained environments such as a borderless maze. However, even in such scenarios where novelty was not as effective, fitness based search was still not a suitable alternative [15].

5.3 Novelty Search with Local Competition

While there has not been much public research on integrating novelty search with local competition the studies done in [10, 11] have explored this technique. The studies look at evolving a diversity of virtual creatures and evolving roguelike dungeons for use in virtual environments such as video games. Both studies compare this technique to a typical evolutionary algorithm objective-based technique as well as novelty search alone. The resulting fitness values and diversity were compared. The objective-based approach in the dungeon experiment was able to outperform novelty search and novelty search with local competition in the fitness value test however in the study done with the diversity of virtual creatures, the fitness values of the objective-based approach and novelty search with local competition were shown to be very similar with novelty search performing the worst. In both studies Novelty search and novelty search with local competition showed similar levels of diversity and significantly outperformed the objective-based solution. Novelty search with local competition expands on novelty search and is shown to provide the benefit of obtaining a wide diversity of solutions whilst finding the best solutions for each niche. It may not always provide the highest global fitness values that an objective-based approach can provide but provides a good balance between diversity and fitness.

6 ANTICIPATED OUTCOMES

6.1 System

The three hybridized methods will be implemented using a simulated swarm robotics environment. We will initially be doing the experiments in RoboRobo which is a fast, 2D simulator for evolutionary swarm robotics [3]. This will allow us to test our algorithms, in a simple 2-D environment, where we can focus on the performance of the swarm due to the algorithms used instead of the computational power required to simulate the environment, which would be more demanding in a 3-D space. If these experiments are successful and there is sufficient time, we will then attempt the experiments using RoboGen which is a 3D simulator.

In order to run this simulation, we will need access to the UCT high performance cluster in addition to personal computers, either our own or on campus. We will use the cluster to run multi-generational simulations to retrieve the results of the experiment.

6.2 Impact

We believe this project will be able to find hybridized methods that will be able maximize both the fitness and functional diversity of the swarm of robots. On top of adding results to the field of swarm robotics, the methods we are hybridizing are used in other fields as well and so we will also be contributing our results and methods to these fields as well. If our results match our hypothesis, we will have found hybridized methods for solving potentially complex and varying problems. If our results do not match our hypothesis, they could still provide valuable information into where to put research efforts.

6.3 Success Factors

The success of this project will be based off a number of factors:

- The successful use of the RoboRobo framework for simulating virtual environments and a swarm of robots for use in each of the method implementations.
- The successful implementation of the three-research objective hybridized methods.
- Designing the fitness test to take into consideration the evaluation metrics we will consider for both individuals and the swarm as mentioned in section 3.5.
- Having a functional diverse swarm of robots that is able to deal with the increasing complexity of the two tasks we will be using as tests for each of the method implementations.
- Extracting meaningful results from the tests which we will be able to draw conclusions from and add to the field of swarm robotics.

7 PROJECT PLAN

7.1 Risks

The risks and mitigation strategies for this project are shown in the risk matrix, Table 1 in Appendix A.

7.2 Timeline

We will start work on the project following the submission of the project proposal on the 27th of May. The project will then run until the 10th of October when the website is due. A full breakdown of the timeline and work allocated is shown in the Gantt chart, Figure 5 in Appendix B.

7.3 Resources Required

For this project we will need access to a computer, either on campus or a personal computer. We will also need access to the RoboRobo framework in order to simulate the robots in a virtual environment and access to the UCT high performance cluster in order to run these simulations.

7.4 Milestones and Deliverables

The milestones for this project are shown in the Gantt chart, Figure 5 in Appendix B.

Below is a list of our key milestones and the tasks we broke them up into in order to achieve them.

- Establishing the RoboRobo Simulator
    - Installation of the RoboRobo software
    - Familiarization of the RoboRobo framework
- Building and implementing the MEDEA Algorithm
    - Discussion of how to implement MEDEA
    - Designing MEDEA to work in RoboRobo environment
    - Code MEDEA algorithm
- Designing the Robot Genome Template
    - Discussion of how to implement the robot genome
    - Designing the robot genome to work with both RoboRobo and each hybrid algorithm
    - Implement Robot genome in RoboRobo
- Implement Hybrid Algorithms (All subtasks will be done by the respective member covering that hybrid algorithm)
    - Algorithm implementation
    - Experimentation
    - Report writing
    - Code Refactoring

7.5 Work Allocation

The work for this project will be split evenly among all three members. Each member will be working on one of the three research objectives independently and we will each produce a report by the end. Kevaalin Rapeti will be hybridizing the minimal Environment Driven Evolutionary Algorithm (mEDEA) with a multi-Behavior Characterization (multi-BC) Map-Elites algorithm. Sihle Calana will be hybridizing the mEDEA algorithm with Novelty Search. Bailey Green will be hybridizing the mEDEA algorithm with Novelty Search and Local Competition.

REFERENCES

[1] Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. 2019. Emergent Tool Use From Multi-Agent Autocurricula. (2019). DOI:https://doi.org/10.48550/arXiv.1909.07528

[2] Nicolas Bredeche and Jean-Marc Montanier. 2010. Environment-driven Embodied Evolution in a Population of Autonomous Agents. "Parallel Problem Solving From Nature (2010).

[3] Nicolas Bredeche, Jean-Marc Montainer, Berend Weel, and Evert Haasdijk. 2013. RoboRobo! a Fast Robot

Simulator for Swarm and Collective Robotics. (2013). DOI:https://doi.org/10.48550/ARXIV.1304.2888

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

[5] Jorge Gomes, Paulo Urbano, and Anders Lyhne Christensen. 2013. Evolution of swarm robotics systems with novelty search. Swarm Intelligence 7, 2–3 (2013), 115–144. DOI:https://doi.org/10.1007/s11721-013-0081-z

[6] F S Grodzinsky, Keith W. Miller, and M Wolf. 2003. Ethical issues in open sourcesoftware. J. Inf. Commun. Ethics Soc.1 (2003), 193–205. DOI:https://doi.org/10.1108/14779960380000235

[7] Emma Hart, Andreas S. W. Steyven, and Ben Paechter. 2018. Evolution of a functionally diverse swarm via a novel decentralised quality-diversity algorithm. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '18). Association for Computing Machinery, New York, NY, USA, 101–108. DOI:https://doi.org/10.1145/3205455.3205481

[8] Aboul Ella Hassanien and Eid Emary. 2016. Swarm Intelligence: Principles, Advances, and Applications (1st ed.). CRC Press.

[9] Eisuke Kita. 2011. Evolutionary Algorithms. IntechOpen.

[10] Joel Lehman and Kenneth O. Stanley. 2011. Evolving a diversity of virtual creatures through novelty search and local competition. In Proceedings of the 13th annual conference on Genetic and evolutionary computation (GECCO '11). Association for Computing Machinery, New York, NY, USA, 211–218. DOI:https://doi.org/10.1145/2001576.2001606

[11] Alexandre S. Melotti and Carlos de Moraes. 2019. Evolving Roguelike Dungeons With Deluged Novelty Search Local Competition. IEEE Transactions on Games. 11, 2 (2019), 173-182. DOI:https://doi.org/10.1109/TG.2018.2859424

[12] Jean-Baptiste Mouret and Jeff Clune. 2015. Illuminating search spaces by mapping elites. CoRR abs/1504.04909, (2015). DOI:https://doi.org/10.48550/ARXIV.1504.04909

[13] Melanie Schranz, Martina Umlauft, Micha Sende, and Wilfried Elmenreich. 2020. Swarm Robotic Behaviors and Current Applications. Frontiers in Robotics and AI 7, (2020), 36. DOI:https://doi.org/10.3389/frobt.2020.00036

[14] Giandomenico Spezzano. 2019. Swarm Robotics. MDPI - Multidisciplinary Digital Publishing Institute.

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

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

A RISK MATRIX

Risk	Probability	Impact	Mitigation
Failure to make use of Robogen Framework due to lack of knowledge or other issue	Low	High	Allocate time to familiarizing ourselves with the Robogen Framework and setting it up for use in our tests.
Failure to meet one or more deadlines	Low	Medium	Weekly meetings with supervisor and follow Gantt Chart created and adjust as necessary if falling behind.
Unable to successfully implement hybridized method due to lack of knowledge and experience in field.	Medium	High	Make sure to research and have a good understanding of methods beforehand and if need be get help from supervisor or other project members.
Not getting access or enough time to the UCT high performance cluster in order to run our tests	Low	High	Try to start testing as early as possible so there is time to fix any potential issues with the high performance cluster

Table 1: Risks and mitigation strategy

B GANTT CHART

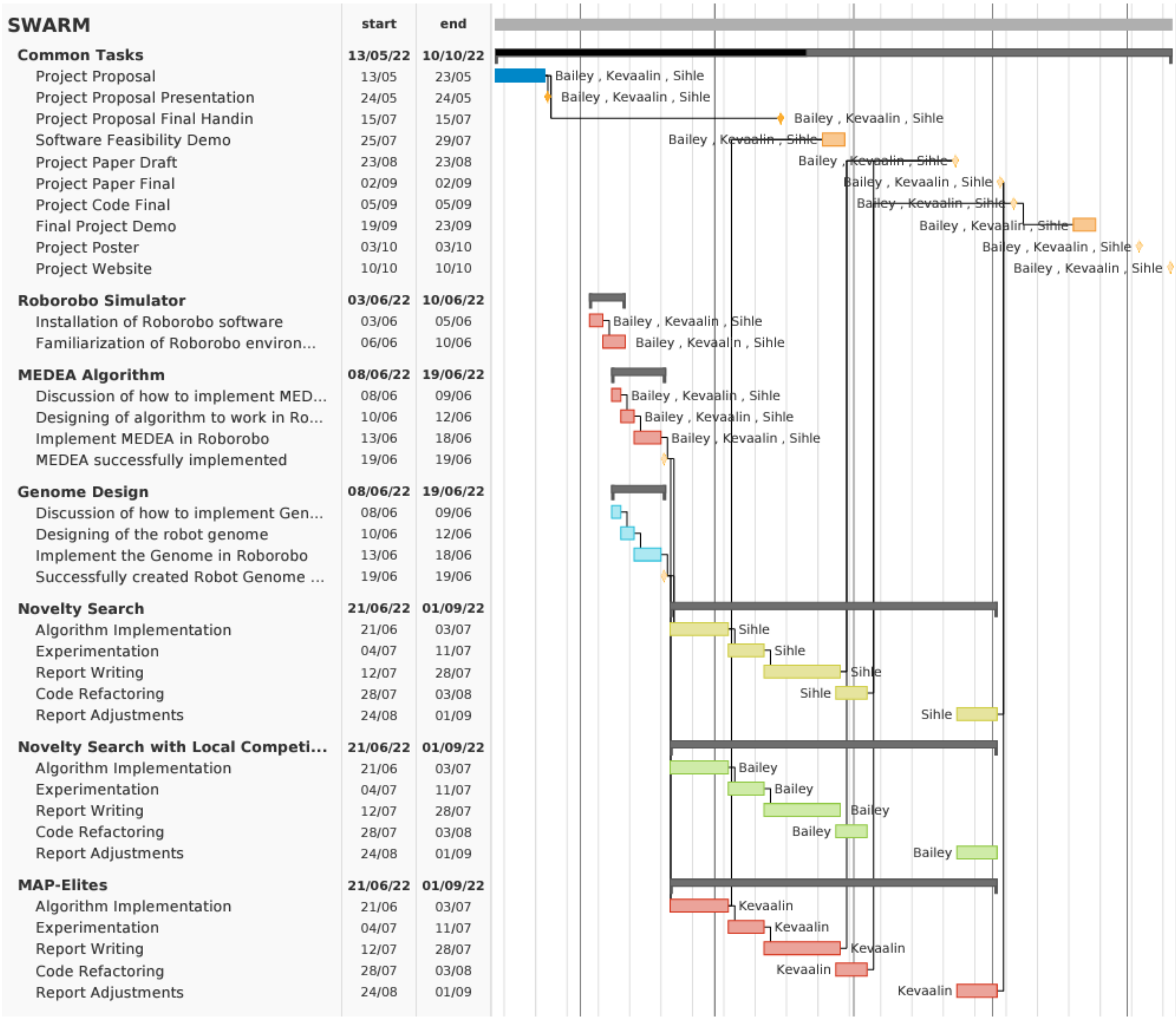

Figure 5: Gantt Chart showing work allocation and milestones

C Project Plan

Experiment Name	Evaluation Metrics	Tasks Needed
Collective Gathering	- Time required by swarm to complete task - Distance resources were moved towards target area	- Swarm needs to search the environment for any resources, for this task a smaller amount of time taken is preferred. - Once resources have been found then the swarm needs to divide the work force and decide which robots will push which resources. The swarm will need to ensure it does not overestimate or underestimate the number of robots to move each resource so that it runs efficiently
Collective Construction	- Time required by swarm to complete task - Distance resources were moved towards target area	- Swarm needs to search the environment for any resources, for this task a smaller amount of time taken is preferred. - Once resources have been found then the swarm needs to divide the work force and decide which robots will push which resources. The swarm will need to ensure it does not overestimate or underestimate the number of robots to move each resource so that it runs efficiently - Once in the target area the swarm will have to determine the shape of the resource and orientation it needs to be in to construct the structure. We will want the robots to place resources in the correct orientation on the first try, so that we reduce redundancies in the swarm.

Table 2: Experiments and their evaluation metrics with the tasks required