The University of Sussex

School of Engineering & Informatics

Masters Dissertation

Programme…MSc in Artificial intelligence and adaptive systems…......

Candidate number ………276209………

Title of Dissertation *Neuroevolutionary swarm robotics with novelty search*……………………………………………………………………………

Approximate number of words ……9200……

Supervisor(s) …………Dr. Chris Johnson…………………………………

Date submitted …22/08/2024... Time submitted ………21h30……….

**Declaration**

I certify that the information on this cover sheet is correct.
I certify that the content of this dissertation is my own work, and that my work contains no examples of misconduct such as plagiarism, collusion, or fabrication of results.

Candidate's signature …….….…  ………………...

UNIVERSITY OF SUSSEX

ARTIFICIAL INTELLIGENCE AND ADAPTIVE SYSTEMS

MASTERS DISSERTATION

# Neuroevolutionary swarm robotics with novelty search

*Author:*
Andres SALAZAR

*Supervisor:*
Dr. Chris JOHNSON

Submission date: August 22, 2024

UNIVERSITY OF SUSSEX

# Abstract

Inspired by natural self-organizing systems such as bird flocks, ant colonies, and bee swarms, swarm robotics emerges as an innovative approach to address complex, modern challenges. While a single robot might tackle a broad range of problems, certain tasks exceed the capabilities of individual robots due to their complexity or hardware limitations. In such cases, the collaboration of multiple robots becomes essential. Swarm robotics has shown promise in various applications, including aggregation, navigation, object clustering, sorting, and collaborative manipulation. Traditionally, swarm robotics controllers are trained and evaluated using objective-based search functions that measure the swarm's performance. However, this method can be inadequate for deceptive problems where the risk of converging to local optima and reaching impasses is high. In these scenarios, abandoning traditional objectives to adopt a novelty search can be crucial. This approach encourages the exploration of diverse behaviors, potentially uncovering superior solutions. This paper investigates the use of novelty search in swarm robotics for a collaborative object manipulation task, a context where traditional approaches may not be enough robust. By comparing novelty search with fitness-based and random search methods, this study not only tests the effectiveness of alternative strategies but also contributes to the broader understanding of swarm intelligence. The research leverages insights from related work and applies them to novel tasks, aiming to demonstrate the unique capabilities of homogeneous robots using uniform controllers to collaboratively maneuver large objects in intricate environments.

# Acknowledgements

I would like to thank my family for their unconditional support and to my supervisor for his time, help, patience, and active participation in my research. Without you, It would not be possible to carry out this experiment and write this document.

# Declaration

This paper is submitted as part of a requirement for a Master's degree at the University of Sussex. It is the product of my own labour except where indicated in the text. The paper may be freely copied and distributed provided the source is acknowledged.

Date: August 22, 2024

Signature:

# Content

# List of figures

# List of tables

# Chapter 1

# Introduction

## 1.1 Swarm robotics, evolutionary robotics and novelty search background

Swarm behaviour, as observed in nature, exemplifies the power of collective effort. In the animal kingdom, species such as ants and bees perform complex tasks that surpass the capabilities of individual members, thanks to their inherent collective behaviour [1] [2] [3]. Inspired by these natural observations, swarm robotics applies similar principles to engineering challenges, particularly tasks that require collaborative effort among multiple entities [4] [5] [6] [7].

A significant portion of swarm robotics employs evolutionary techniques such as genetic algorithms to evaluate the performance of an individual based on natural selection [5][6][8]. However, unlike traditional methods in robotics, which predominantly employ objective-based search algorithms, a growing subset of this field has shifted towards novelty search. This approach prioritizes a broader exploration of the solution space, thereby mitigating the risk of stagnation at local optima [9]. Researchers have highlighted the effectiveness of novelty search over objective-based methods in dealing with deceptive tasks, where conventional strategies may inadvertently lead to suboptimal solutions or dead ends [10]. The superiority of novelty search has been demonstrated in contexts such as aggregation, resource sharing and with single robots in neuro-evolution in deceptive mazes, where it not only outperformed objective-based strategies but also evolved more compact and efficient programs [11][12]. Additionally, novelty search has been noted for its capacity to explore the outcome space extensively and uniformly, an attribute critical to achieving comprehensive problem-solving in robotics [13].

The field of evolutionary robotics further enriches this landscape by integrating principles of natural selection to engineer more autonomous, collaborative and adaptive robots [14]. This approach is particularly powerful when combined with novelty search, encouraging the exploration of new behaviours rather than merely enhancing existing functionalities [15]. Such methodologies are invaluable in swarm robotics, where the synergy of small robots can tackle complex and dynamic tasks, like the adaptability and versatility seen in modular robotics systems designed to meet evolving challenges [16]. By leveraging these evolutionary principles, it becomes achievable to develop robotic systems that are not only more proficient at navigating their environments but are also capable of handling tasks that demanding and dangerous for human intervention [17].

## 1.2 Research motivation and objectives

In the last 20 years, there has been a growing integration of robotics in various fields [18] [19]. From basic automatic vacuum robots to intelligent robots in airports that effectively solve customer's requests [20] [21]. However, the incorporation of robotics in other areas as rescue operations and construction implements one robot that performs one task [22][23]. Moreover, swarm robotics have been studied as first responder in emergencies to detect areas with high probability to find survivors [24] [25].

Novelty search is particularly effective in evolving controllers for robot swarms handling complex tasks like collaborative object manipulation, which often involve deceptive challenges where straightforward objectives might lead to suboptimal solutions [12]. This approach is crucial in swarm robotics, where leveraging the collective abilities of robots can lead to innovative problem-solving strategies inspired by natural systems, such as flocking and navigating, which are essential for tasks requiring high levels of coordination and adaptability [26].

Moreover, novelty search's ability to overcome biases introduced by handcrafted behavior similarity measures in traditional evolutionary approaches is especially valuable, as it enables a more genuine exploration of potential solutions [27]. Utilizing simulation environments like Webots and Enki further enhances this research, allowing for precise control and replication of experimental conditions, facilitating the detailed study of swarm behavior under a variety of simulated physical constraints [28]. Together, these studies punctuated the synergy between novelty search, swarm robotics, and evolutionary robotics in addressing complex, deceptive tasks by promoting diversity and innovation in robotic evolution.

This paper aims to contribute to swarm evolutionary robotics using novelty search. Especially in deceptive tasks where objective-based search is not completely effective. The main research will be focused on implementing the novelty search algorithm in a deceptive task with a swarm of robots, analysing the potential gaps and improvement points in the algorithm and behaviour characterisation on each simulation. Then, the performance will be compared with fitness-based search and random search.

Additionally, explores the potential of small robots, operating in swarms, to collectively perform tasks that are not possible to perform for a single robot and hazardous for humans. Specifically, this project demonstrates that homogeneous robots using the same controller can collaboratively move large objects in complex scenarios.

## 1.3 Research questions

This paper will examine three main factors that could potentially affect neuro-evolution in swarm robotics with novelty search. The research questions are as follows:

1. How does the performance of a neural-based controller evolve through novelty search compared to evolved through fitness-based search in the context of a specific deceptive task in swarm robotics?

This question aims to evaluate the comparative performance of neural controllers evolved through novelty search versus those evolved via fitness-based search, specifically within a deceptive task context in swarm robotics.

2. What unique behavioural patterns emerge in homogeneous robot swarms when using novelty search compared to fitness-based approach?

This question seeks to identify and analyse the unique behavioural patterns that emerge in homogeneous robot swarms when neural controllers evolved using novelty search compared to traditional fitness-based method.

3. How effective is the behaviour characterisation proposed with novelty search in evolving neural controllers for collaborative tasks?

This question investigates the effectiveness of the proposed behaviour characterisation in novelty search for evolving neural controllers that enable robots to perform collaborative tasks efficiently in swarm robotics.

## 1.4 Structure of dissertation

The structure of this thesis consists of three main sections. The first part (Chapter 1) is a general introduction and background in the area. The second part (Chapter 2) is a review of the relevant literature at the time of exploring and developing methods and theories in this research. The following section (Chapter 3) describes the research methodology – i.e. designing of environmental conditions, robot's set-up and data collection from simulations. Then, the data is analysed and interpreted to solve the research questions. Finally, the last section is the conclusion (Chapter 4). Contains a summary of the results, recommendations and limitations of the research.

# Chapter 2

# Literature Review

## 2.1 Introduction and approach

This chapter investigates the existing literature, extracting theories and ideas pertinent to evolutionary robotics, swarm robotics and novelty search, especially focusing on the potential benefits of using them together.

## 2.2 Evolutionary Robotics

### 2.2.1 Why evolutionary robotics?

The implementation of evolutionary robotics (ER) in the study of swarm robotics, particularly for deceptive tasks, builds upon foundational work that has consistently demonstrated the advantages of ER in handling complex, dynamic environments. Evolutionary algorithms, in ER, have proven effective in developing autonomous robots capable of adapting to and evolving within their operational contexts without extensive human intervention. For instance, work highlighted in the ”Evolutionary modular robotics: Survey and analysis” emphasizes the effectiveness of evolutionary-based solutions in introducing novel capabilities to robots, which surpasses traditional hand-designed methods by automating the evolution of robot control systems and morphologies [16]. Similarly, [29] underscores the importance of embodied intelligence in ER, where robots are viewed as integrated systems interacting dynamically with their environments, thereby promoting a holistic approach to robot design.

Moreover, the unique approaches to ER at Sussex, as detailed in [30], emphasize the shift from traditional design methodologies to evolutionary strategies. This transition highlights the inherent limitations of human-designed systems and the potential advantages of evolutionarily derived systems. The research at Sussex has focused on evolving controllers for autonomous robots, utilizing both simulated and real robots, which underscores the practical applications and challenges of ER in real-world settings. This study aims to bridge these gaps by implementing novelty search, a technique that prioritizes behavioral diversity over fitness measures tied directly to task performance. Novelty search, as explored in [14], has shown potential in avoiding local optima common with fitness-based approaches by fostering a broader exploration of the behaviour space. This approach is particularly pertinent in swarm robotics, where the collective capabilities of robots can be significantly enhanced by diversifying the strategies they can employ to accomplish tasks potentially hazardous for humans. Furthermore, by comparing the performance of novelty search with traditional fitness and random search methods, this research not only aims to validate the efficacy of behavioural diversity in ER but also addresses the unexplored potential of homogeneous robots using shared control systems to perform complex collaborative tasks, as discussed in [17].

Figure 2.1 illustrates an example of the principal word-flow of evolutionary robotics. Consists of an evolutionary algorithm (left) and the robot and its interaction with the environment (right) [14].

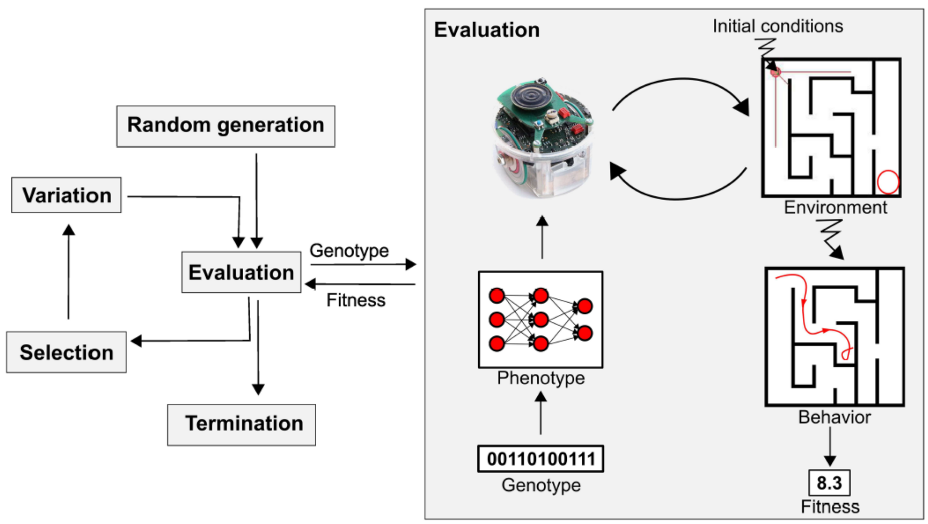



**Figure 2.1:** Evolutionary robotics work-flow example [14].

### 2.2.2 Behaviour in evolutionary robotics

The search for robust autonomous robotic systems in ER has shifted towards enhancing behavioural diversity, particularly through the implementation of novelty search algorithms. This approach is fundamentally rooted in the observation that traditional fitness-based search methods could fail to escape local optima (See Figure 2.2), particularly in complex and deceptive tasks where the evolutionary trajectory may mislead rather than guide [31]. By focusing on behavioural diversity, ER aims to explore a broader spectrum of potential solutions, essentially separating evolutionary success from specific task performance metrics and instead promoting a variety of successful strategies [32]. In addition, [33] introduces behaviour-based speciation, which maintains diversity within evolutionary populations by developing multiple effective behaviours to achieve a given task. This method contrasts traditional genetic or topological speciation approaches, highlighting its potential to promoting diverse behavioural strategies effectively within evolutionary populations.

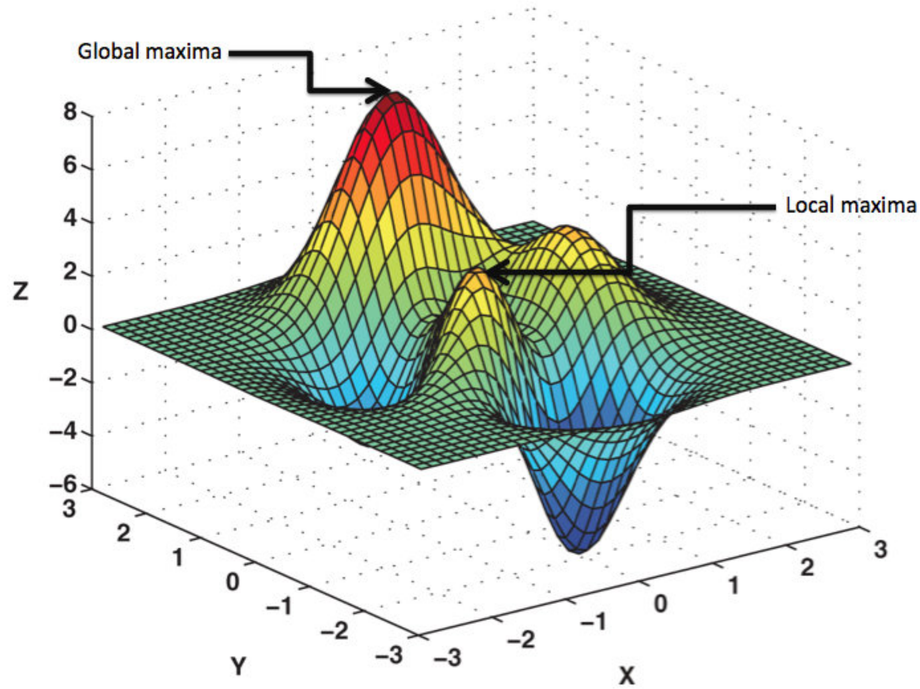



**Figure 2.2:** Local and global optima [34].

Moreover, the incorporation of systematic approaches to behaviour characterisation, such as Systematically Derived Behaviour Characterisations (SDBCs), further improve the process by reducing experimenter bias and enhancing the relevance of behavioural evaluations [35]. This systematic derivation ensures that behaviour characterisations are both relevant to the task and without arbitrary influences, providing a more objective basis for assessing evolutionary progress. Such advancements not only address the limitations inherent in fitness-based approaches but also align with the foundational principles of ER, where the imitation of natural evolutionary processes (through the encouragement of diversity and the iterative exploration of new behaviour species) promises to produce solutions unreachable by traditional design methodologies [36]. The novelty search exemplifies this by prioritizing behavioural novelty over immediate task performance, which has been shown to lead to innovative and often more robust solutions [35].

# 2.3 Swarm robotics

## 2.3.1 Characteristics of swarm robotics

Swarm robotics (SR), inspired by the self-organizing behaviours of biological systems such as ants, bees, and birds, emphasizes decentralized control and local interactions to achieve complex collective behaviours (See figure 2.3). These systems are robust and adaptable, effectively addressing complex tasks that surpass the capabilities of individual agents [37][4][5]. The importance of studying SR lies in its potential to handle dynamic and unpredictable environments, leveraging simple, minimalistic robots that collaboratively achieve sophisticated tasks without centralized supervision. This is particularly crucial in applications such as disaster response, environmental monitoring, and military operations, where traditional robotics approaches may face difficulties [4][5][6]. The literature underscores the foundational principles of swarm intelligence such as simplicity of individual units, distributed control, and local sensing. Which are not just theoretical principles but are increasingly applied in practical, real-world scenarios, demonstrating the utility of SR over single-robot systems [5][38][26].

**Figure 2.3:** Biological swarms in nature [26].

Current research in SR is often done around refining and investigating algorithms that facilitate autonomous, cooperative behaviour, focusing on robustness and operational efficiency even under partial system failures [37][26]. Despite extensive theoretical advancements, the implementation of these algorithms into practical applications remains challenging, highlighting a critical gap in the current research landscape [4][6][26]. My study addresses this gap by implementing novelty search algorithms in SR, specifically targeting deceptive tasks where traditional objective-based searches may not be the best approach. By comparing the outcomes with fitness-based and random searches, this research not only contributes to understanding algorithmic efficiency in complex tasks but also enhances the characterisation of behaviour dynamics within swarms [37][5].

### 2.3.2 Applications and challenges

SR encapsulates a multidisciplinary approach that uses the principles of swarm intelligence with the functionalities of multi-robot systems, enabling an unlimited of applications from disaster management to agriculture. Inspired by the coordinated behaviours of natural swarms, such as ants and bees, SR systems prioritize decentralized control and local interactions among robots to foster behaviours that lead to solving real-world problems [39][40][41]. For instance, SR has been key in hazardous operations, such as search and rescue missions, where traditional robotic systems may be less effective due to the complexity and unpredictability of the environment. This operational domain underscores the significance of robustness and adaptability given by the inherent features of SR due to its decentralized nature and ability to self-organize without centralized oversight [39][40][7].

However, despite its extensive potential, the deployment of SR in practical applications faces considerable challenges, primarily relating to the scalability of systems and the standardization of hardware and software. These obstacles highlight a critical gap between theoretical research and real-world application, an area my study seeks to address by investigating novel algorithms capable of enhancing the autonomy and efficiency of swarm behaviours in complex tasks as illustrated in Figure **??** [39][40][42]. Furthermore, my research aims to contribute to the standardization efforts in SR by proposing a framework that could potentially optimize the integration of SR systems into practical scenarios. This initiative not only aligns with the current research trends but also amplifies the relevance of SR in advancing autonomous systems capable of collaborative and intelligent task execution, thereby contribute to innovative solutions in all applications [41][7][42].

**(a)** Swarm of bots in square formation passing over a trough.

**(b)** Swarm of bots with a linear formation moving on rough terrain.

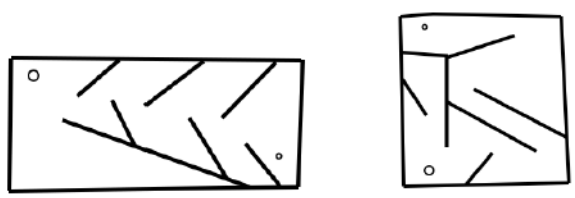
**Figure 2.4:** Swarm robotics real world applications [42].

## 2.4 Novelty search

### 2.4.1 Novelty search in swarm robotics

Novelty Search (NS) is a significant concept in evolutionary robotics, promoting the exploration of new behaviours rather than optimizing pre-defined fitness goals. NS is particularly advantageous in environments where traditional fitness landscapes are misleading or not robust enough to achieve complex solutions for complex problems in dynamic conditions. This approach has been theoretically modelled to behave like a uniform random search in the behaviour space, a complex space linked to the genotype space that traditional methods cannot directly sample [9]. This algorithm enhances exploration, that is crucial, thus preventing premature convergence on suboptimal solutions that are common in deceptive tasks like those involved in swarm robotics [9][43].

NS proves particularly effective in scenarios where objective-based searches fail. Traditional fitness functions often mislead optimization processes towards ineffective solutions, especially under deceptive or complex conditions. For instance, in tasks where the objective landscape can misdirect algorithms, such as navigating mazes (figure 2.5) or bipedal locomotion, NS has shown to outperform traditional methods by focusing solely on discovering novel behaviours without regard to specific outcomes [10]. This method's ability to overcome the pitfalls of objective-driven searches makes it suitable for complex scenarios where adaptability and exploration are more beneficial than direct optimization [10][43].



**(a)** Medium difficulty map. **(b)** High difficulty map.

**Figure 2.5:** Deceptive mazes illustration [10].

The components of NS, such as the selection of most novel candidates with difference metrics like nearest neighbours (k parameter), Levenshtein and Euclidean distance, use of archives, and mutation rates, play significant roles in determining the effectiveness of the search [43]. Adaptive behaviour characterisations (BCs) boost NS by learning from behaviours across training tasks, thus improving search efficiency and effectiveness in complex domains [44]. This approach, particularly with learned BCs, addresses the shortage of both hand-coded and generic BCs by dynamically adapting to new challenges, thereby significantly enhancing the ability of NS to navigate through deceptive landscapes [44].

### 2.4.2 Evolving with novelty search

NS inherently drives high evolvability by encouraging continuous exploration within the behaviour space, which is crucial for evolutionary algorithms, especially in bounded or constrained environments. Unlike traditional methods that focus on optimizing specific fitness functions, NS promotes the discovery of novel behaviours, providing the ability of evolutionary processes to generate diverse and effective solutions. This focus on exploration rather than exploitation helps maintain population diversity, crucial for avoiding premature convergence on local optima and ensuring sustained adaptability in dynamic environments [13][45]. The dynamic nature of the evaluation criteria in NS, which adjusts based on the evolving population, plays a critical role in promoting evolvability and diversity [13].

The minimal criteria approach in novelty search further enhances the capability of NS by integrating basic survival or performance thresholds, which must be met side by side with novelty. This method, known as Progressive Minimal Criteria Novelty Search (PMCNS), combines the benefits of fitness-based evolution with NS, focusing on exploring new regions of the behaviour space while satisfying increasingly rigorous criteria [46]. PMCNS has shown superior performance in complex tasks, balancing exploration with necessary exploitation, thus effectively navigating through deceptive evolutionary landscapes. It overcomes the limitations of minimal criteria novelty search by dynamically adjusting fitness thresholds, ensuring that the evolution process remains robust and aligned with practical objectives [47].

In the context of machine learning, novelty search presents a promising approach to addressing the limitations posed by deceptive objective functions that may interfere with reaching true solutions. By continuously promoting novelty, this approach bypasses the deceptive gradients of traditional fitness functions, which can mislead the search process. The integration of NS into genetic programming and other machine learning frameworks has demonstrated significant improvements in problem-solving efficiency, particularly in complex, deceptive tasks. This adaptability and effectiveness of NS in machine learning illustrates its potential as a broadly applicable solution in artificial life and beyond, reflecting the dynamic and open-ended nature of natural evolutionary processes [11][48].

## 2.5 Related work

SR has seen substantial advancements through the integration of evolutionary algorithms, particularly NS, as a robust alternative to traditional fitness-based methods. For example, the exploration of NS combined with NEAT (short for Neuro-Evolution of Augmenting Topologies) - NEAT optimizes both the weight parameters and the structure of artificial neural networks, this process allows the networks to develop diverse and complex solutions over time [49][50]- in swarm robotics tasks demonstrates enhanced performance in environments inclined to deception, such as task aggregation and resource sharing [12]. The effectiveness of NS in these contexts lies in its ability to reward novel behaviours rather than merely successful outcomes, thereby avoiding common pitfalls such as premature convergence and local optima [12]. This is mentioned in another study where NS, applied to online neuro-evolution, allowed robotic swarms to adapt dynamically in deceptive foraging scenarios, confirming its superiority over traditional fitness-based approaches [51]. The successful application of NS in these studies not only highlights its potential in managing complex swarm behaviours but also suggests a broader applicability for tackling similar challenges in other domains of ER.

The performance and advantages of using NS in ER are further underscored by its capacity to integrate with generic behaviour similarity measures, reducing the reliance on task-specific metrics. This integration is key in simplifying the evolutionary process, as demonstrated in tasks like aggregation and resource sharing, where generic measures are performed comparably to hand-crafted, task-specific ones [27]. The adoption of generic measures supports the hypothesis that less experimenter intervention can lead to a true diversity of evolved solutions, minimizing experimenter bias and reducing workload [27]. These findings are crucial in illustrating how NS can be effectively implemented without the complexity of designing task-specific measures, thereby promoting a more autonomous system that is adaptable across various operational contexts.

The utility of simulation environments like Webots and Enki for testing and developing SR algorithms is another critical aspect of current research. The use of Particle Swarm Optimization (PSO) to coordinate collective behaviours in swarm robots within a simulated framework highlights both the challenges and potential improvements in algorithmic design [28]. This approach not only reduces costs and mitigates risks associated with physical deployments but also provides a controlled setting to evaluate the impact of physical conditions on robot performance [28]. By identifying inefficiencies and potential algorithmic enhancements, such as adjusting iteration values and error handling, these simulations play an important role in improving the approaches used in SR, ensuring that the robots can effectively coordinate and perform complex tasks, which aligns with the broader objectives of enhancing the robustness and adaptability of swarm systems.

## 2.6 Simulation software

Platforms such as Argos, Gazebo, ISAAC Sim, Webots, and CoppeliaSim were evaluated for their respective capabilities and compatibility with various operating systems. Argos is known for its adaptability, supporting multiple physics engines and functioning on Linux and macOS, which is beneficial for simulations requiring flexibility [52]. Gazebo, developed by the Open-Source Robotics Foundation, is distinguished by its robustness and compatibility with Linux, macOS, and Windows, supported by a suite of physics engines like ODE, Bullet, and DART, making it a good choice for academic and professional tasks [53].

ISAAC Sim, a product of Nvidia, is tailored for Linux users and integrates PhysX for advanced simulations, positioning it as a preferred tool for robotics applications involving artificial intelligence [54]. On the other hand, CoppeliaSim offers extensive versatility with support for multiple programming languages and operating systems, accommodating complex, interdisciplinary simulation needs [55]. Webots stands out with its use of a custom version of the ODE physics engine, providing an effective platform for realistic simulations across Linux, macOS, and Windows, which is particularly suited for educational and research-focused projects due to its ease of use and comprehensive documentation [56].

However, for the specific requirements of my research, Enki was selected due to its operational efficiency and straightforward Python integration. Enki's design prioritizes speed and ease of use, facilitating rapid setup and execution. This efficiency allows for an increased focus on the research itself rather than on extensive software management. Enki's focused feature set and performance efficiency make it an optimal choice for my academic research, where simplicity and effective simulation are key.

**Table 2.1:** Simulation software comparison

| Software | Developer | Physics engine | Supported OS | Languages | Top companies |
|---|---|---|---|---|---|
| Argos | Swarmanoid project | Multiple physics engine | Linux, macOS | C++ | Ascens/E-Swarm Swarmix |
| Gazebo | Open-source Robotics Foundation (OSRF) | ODE/Bullet DART | Linux, macOS Windows | C++ | Skarc Technologies TerraClear Toyota Vicarious InDro Robotics |
| ISAAC Sim | Nvidia | PhysX | Linux | C++/Python | BYD electronics Siemens Teradyne Robotics Instrict |
| Enki | Stéphane Magnenat Lausanne (EPFL) | Custom | Linux | C++/Python | Academia |
| Coppelia Sim | Coppelia Robotics | MujoCo Bullet ODE Newton Vortex | Linux, macOS Windows | C++/Python Java MATLAB | Kuka/Airbus intel/Honda Kyocera/esa |
| Webots | Cyberbotics | Custom version of ODE | Linux, macOS Windows | C/C++ MATLAB Python | NASA Samsung Siemens Sony Toyota |

## 2.7 Conclusion

Evolutionary robotics, swarm robotics, and novelty search, increase their potential when combined. It is evident how novelty search, which focuses on exploring new behaviours rather than just achieving specific goals, can help overcome common challenges in robotics like getting stuck in local optima in complex environments. This review also emphasizes that using groups of simple robots can address complex tasks that would be difficult or dangerous for humans or individual robots.

This research significantly advances the field by using novelty search, to tackle complex challenges that traditional methods face with difficulty. It focuses on groups of robots working together, expanding their ability to handle real-world tasks that are complicated for just one robot, especially in dangerous environments. This new approach fills important gaps in previous research by addressing the unpredictable nature of real-world settings and greatly improving the robots' ability to adapt and solve problems together. This study not only enhances our understanding of how groups of robots can work together more effectively but also shows how they can handle larger and more complex tasks safely and efficiently.

# Chapter 3

# Methodology, analysis and discussion

## 3.1 Methods

### 3.1.1 Simulation software

Enki was chosen as the simulation platform due to several key advantages that align perfectly with the research goals. Principally, Enki offers exceptional operational efficiency and stable Python integration, which are critical for the rapid set-up and execution of simulations. The choice of Enki ensures that the focus remains on optimizing and analysing the algorithm's performance in deceptive tasks, rather than on the complexity of the simulation platform itself. This simplified approach facilitates a more effective and focused investigation, making Enki an indispensable tool for achieving the objectives of this research.

Enki set-up is fast and straightforward. The instructions to compile the Enki and use Python bindings are in the official GitHub repository [57].

### 3.1.2 Genetic algorithm

The genetic algorithm is based on Charles Darwin's natural selection theory. Darwin says "if variations useful to any organic being do occur, assuredly individuals thus characterized will have the best chance of being preserved in the struggle for life; and from the strong principle of inheritance, they will tend to produce offspring similarly characterized. This principle of preservation, I have called, for the sake of brevity, Natural Selection" [8].
The fundamentals of a genetic algorithm according to [58] are **population, individual, fitness** and **selection**.

- **Population:** Collection of candidate solutions that are considered during the course of the algorithm. Over the generations of the algorithm, new members are "born" into the population, while others "die" out of the population.

- **Individual:** Candidate in the population.

- **Fitness:** Measure of how "good" the solution is represented by the individual. Usually, the higher the fitness, the better - Depending on the problem to be solved.

- **Selection:** The survival of the fittest individual. Individuals that are selected for "breeding" – That is where the **crossover** takes place, based on the parent's fitness values where a high fitness makes more likely an individual to reproduce. Moreover, during each generation, there is a small probability for each individual to **mutate**, which changes the individual in some small proportion.

The genetic algorithm implemented was based on [58] and [59]. However, in this case fitness is the novelty of each candidate and these candidates are selected from the novelty archive. The pseudo-code is shown below:

**Algorithm 1** Genetic Algorithm

```
Create a population of random candidate solutions, number of genes, and number of
generations named pop, NoGenes, NoIndividuals and NoGenerations.
Calculate Novelty for the initial population.
while algorithm termination conditions are not met do
    Create an empty population named newPop.
    while newPop is not full do
        Select parents proportional to novelty.
        Parents are selected by tournament selection.
        Crossover parents to create children.
        Mutate children.
        Add children to newPop.
    Replace pop with newPop.
Select from the final novelty archive novelty archive the candidates with the behaviour
that satisfy the task criteria.
```

### 3.1.3 Novelty search algorithm

The structure of the experiment's novelty search archive algorithm differs from traditional methods by focusing on behavioural differences measured using either Euclidean or Levenshtein distances, without incorporating sparseness as a metric. The archive is a list of dictionaries, where each dictionary entry contains essential information about the genotype, such as genotype ID, genome, behaviour, and novelty score. Initially, the archive collects the first N behaviours, storing them directly until full capacity. Once the archive reaches its limit, any new candidate behaviour is compared against those in the archive. If the novelty of the new candidate surpasses the least novel entry in the archive, it replaces that entry. This process ensures the archive consistently retains the most novel behaviours, providing a complete collection of the most unique solutions discovered during the search.

The implementation is based on the approaches described in [12] by Gomes et al., and [10] by Lehman and Stanley. In the former implementation, sparseness is a critical factor, as novelty search focuses on exploring diverse behavioural regions to prevent premature convergence, especially in deceptive landscapes. In contrast, the proposed method does not utilize

sparseness, relying solely on behavioural differences. Similarly, Lehman and Stanley's approach leverages a dynamic measure of novelty, often integrating a minimal criteria novelty search (MCNS) to ensure behaviour meets domain-specific requirements. This archive strategy simplifies this by not enforcing such criteria, allowing for a more straightforward yet effective selection process based on the calculated novelty alone. This streamlined approach maintains the core principles of novelty search while adapting it to a context where direct behavioural comparison is adequate for distinguishing novel solutions.
The proposed novelty archive pseudo-code is shown below:

**Algorithm 2** Novelty search algorithm

```
if length of archive.archive == 0 then
    insert_entry(genotype, final_bd, 0, genotype_id)
    add_novelty_to_behaviour(0, genotype_id)
    Add 0 to population_novelty
    Update genotype_id
else
    Compute novelty, diffs = compute_novelty(final_bd)
    insert_entry(genotype, final_bd, novelty, genotype_id)
    add_novelty_to_behaviour(novelty, genotype_id)
    Add novelty to population_novelty
    Update genotype_id
```

## 3.1.4 Controllers

### 3.1.4.1 CTRNN controller

The description of the CTRNN used and why was used in the experimentation can be found in appendix .1.

### 3.1.4.2 Feed forward neural network controller

The experimentation involved the implementation of a robot controller based on a fully-connected feed-forward neural network (FNN), optimized using a GA. The FNN served as the core decision-making component, leveraging its ability to model complex relations between motors and sensors mappings and hierarchical representations of the environment. By incorporating multiple hidden layers, the FNN extracted meaningful features from sensor inputs, enabling the robot to navigate and explore effectively. The use of a GA for parameter optimization enhanced the FNN's performance by iteratively evolving the neural network's weights and biases, thus improving obstacle avoidance and exploration capabilities.
The related work experimental results have demonstrated the efficacy of the proposed approach in real-world scenarios, showcasing the robot's ability to avoid obstacles and explore unknown areas efficiently [60] [61]. The FNN robustness and generalization capabilities are features that may enable the robot to navigate complex environments with minimal human intervention.

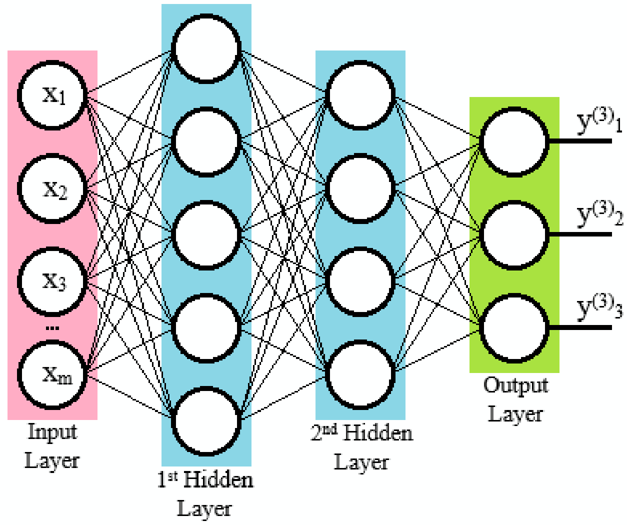



**Figure 3.1:** Fully connected feed forward neural network structure [62].

### 3.1.5 Experimental set-up

The simulation takes place within a 2-meter by 2-meter square area bordered by walls. This set-up includes four distinct environments. The environments tested in this study were a series of maps varying from "low" to "high" difficulty, each designed to incrementally increase the complexity of the tasks. The level of difficulty was determined by the variety of potential trajectories the cylinder could take during the simulation. Initially, the simpler maps were designed to allow fewer possible cylinder behaviours, facilitating a straightforward assessment of the novelty search algorithm. As the testing progressed to more complex tasks, additional local optima were introduced starting from the second environment. This increased the complexity for both novelty and fitness-based searches, systematically challenging the algorithms to adapt and perform under progressively tougher conditions.

In the simulation, two e-puck robots were used, representing the minimum number required to effectively solve collaborative tasks. The challenge was intentionally designed around the weight of the cylinder, which was set heavy enough that a single robot could not move it on its own. This requirement needs genuine collaboration between the robots, complicating the task as the evolution process must adapt the neural-based controllers to ensure the robots not only work together but also push the cylinder in the correct direction. This set-up strategically enhances the complexity of the task, testing the limits of both the robots and the evolutionary algorithms.

In the "low" difficulty environment, robots begin at the bottom of the map with a cylinder positioned centrally. In this first map, there is a funnel that make the task less challenging. The challenge involves navigating the cylinder through a gap to reach a designated endpoint. The final desired position is at the top right corner of the map. Figure 3.3 illustrates the layout of this initial environment.

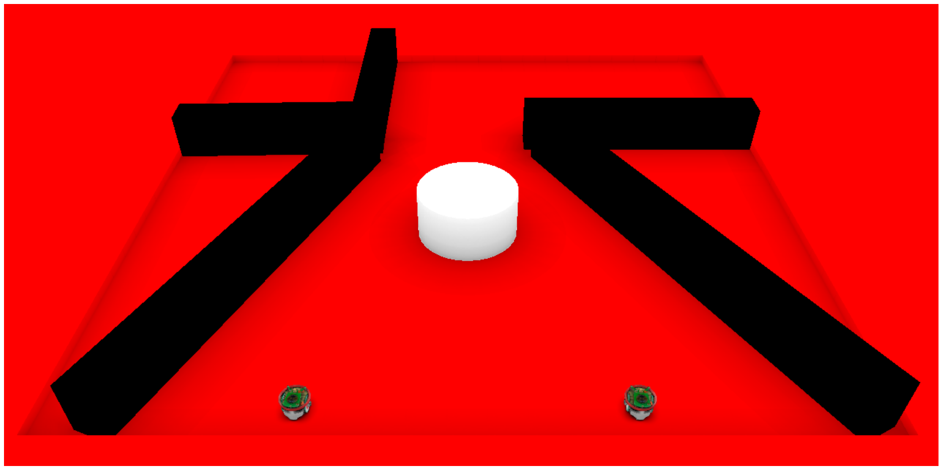


**Figure 3.2:** Low difficulty map.

On the other hand, the "medium" environment presents a slightly more complex challenge, characterized by removing the walls that worked as a funnel and adding a vertical wall to create just one "trap" at the left of the environment. Here, robots must navigate through the gap without the funnel. This map is particularly challenging because the fitness function rewards apparent progress, represented by the cylinder in the right-hand wall, which can mislead optimization efforts. Nevertheless, employing novelty search, a broader exploration of behavioural possibilities is explored, increasing the possibility to find a solution. In this scenario, the starting position of the robots significantly increases the chances of the cylinder to be pushed through the gap. However, the robots most cooperate to guide the cylinder with precision.

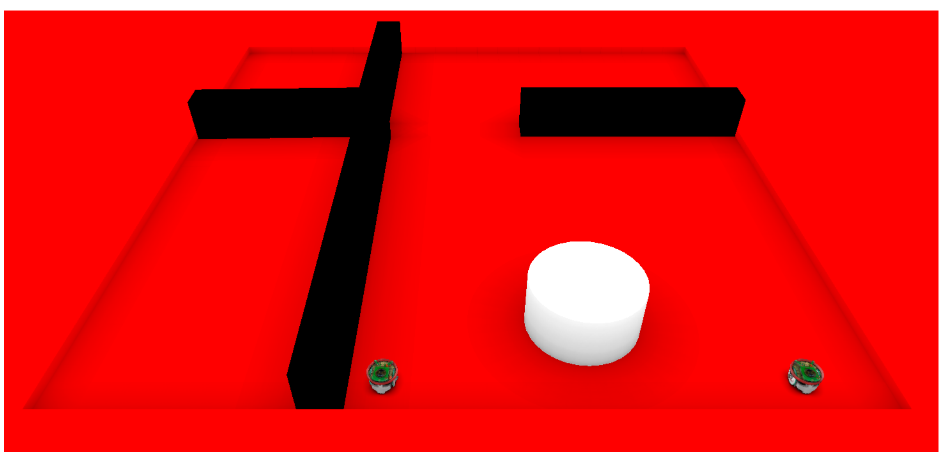


**Figure 3.3:** Medium difficulty map.

The "medium-high" environment introduce an environment with more possible behaviours. This configuration embody the "medium" environment. However, the large vertical wall is removed and the cylinder's and robot's position is modified to foster different behaviours.

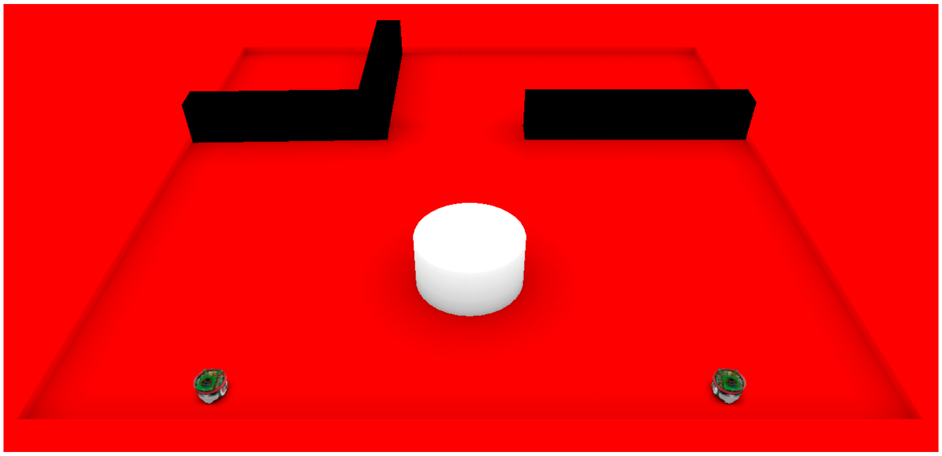


**Figure 3.4:** Medium-High difficulty map.

Finally, the "high" environment presents the most complex scenario. This setting consists of a maze of walls designed to "trap" the cylinder, adding a significant challenge for navigation. In this environment, robots are tasked with escaping the trap and manoeuvring the cylinder to a designated endpoint. The complexity arises from the fitness function, which rewards the cylinder's entrapment as a form of progress, potentially diverting optimization efforts. However, leveraging a novelty search approach enables a more expansive exploration of behavioural strategies, enhancing the probability of finding effective solutions. Additionally, the initial position of the robots directly influences the possibility of the cylinder to be pushed towards the trap initially, further complicating the task.

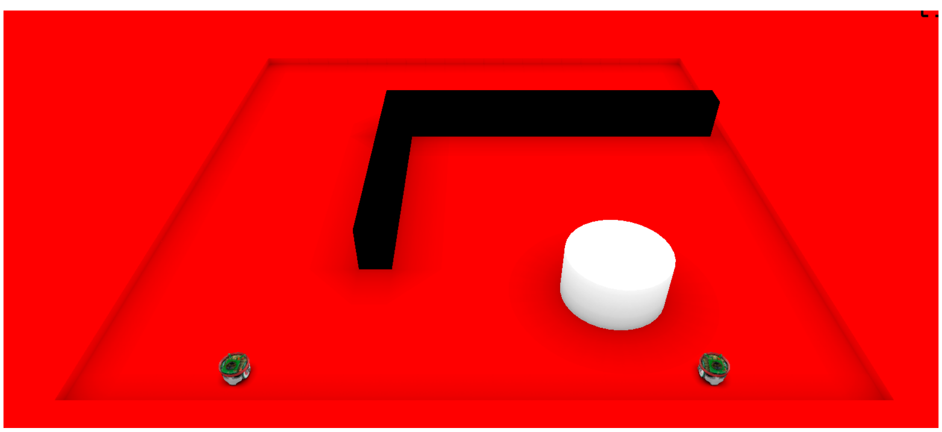


**Figure 3.5:** High difficulty map.

The robots used in the simulation are two identical e-pucks. Each robot is equipped with a camera and infra-red sensors, arranged in the same configuration as found on the standard e-puck educational robot (Figure 3.6). This set-up in principle allow transferring the evolved controllers to real robots for real world applications.

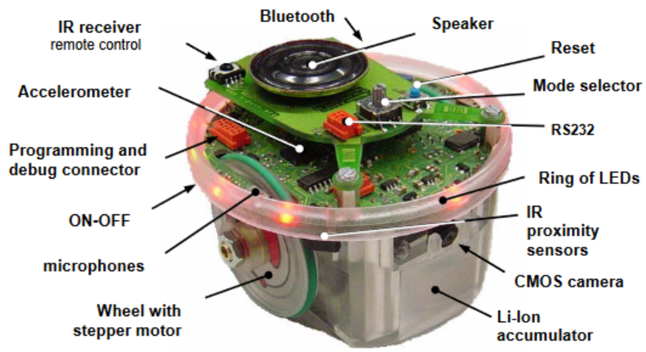



**Figure 3.6:** E-puck sensors and components [63].

Each controller was evaluated once. In 6 independent code executions. The robots always began at the same location, with the cylinder positioned approximately 25 cm away from them. Each simulation ran for 1,500 steps to ensure thorough testing. Finally, the best candidate was post-evaluated in 10 simulations, to obtain more accurate behaviour.

### 3.1.6 Fitness function

The fitness function selected for the experiment is based on the Euclidean distance between the cylinder's final position and the desired target position, both represented by x and y coordinates on the map.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{3.1}$$

This choice is driven by the simplicity and clarity with which it describes the desired outcome. The function effectively rewards solutions that move the cylinder closer to the target position while penalizing deviations. By employing this straightforward metric, the fitness function directly aligns with the objective of evaluating and overcoming the limitations of traditional objective-based search methods, particularly in deceptive tasks where novelty search is proposed as a more effective alternative. This approach ensures a clear and quantifiable measure of success, facilitating a robust comparison between fitness-based search, random search, and novelty search.

### 3.1.7 Levenshtein distance

The Levenshtein algorithm, also known as the Edit-Distance algorithm, determines the minimum number of edits required to transform one string into another. It utilizes a matrix where the cell (m,n) represents the Levenshtein Distance between the m-character prefix of one string and the n-character prefix of the other. The matrix is populated from the upper left to the lower right corner, with each horizontal or vertical move in the matrix representing an insertion or a deletion, respectively, typically assigning a cost of 1 to these operations. A diagonal move, which corresponds to a substitution, incurs a cost of one if the corresponding characters differ, or zero if they match, ensuring that each cell's value is the result of local cost minimization. The value at the bottom right corner of the matrix then indicates the total Levenshtein Distance between the two strings. There are essentially two optimal paths within the matrix that yield this minimum cost, characterized by the operations: "=" for Match, "o" for Substitution, "+" for Insertion, and "-" for Deletion [64].

**Algorithm 3** Levenshtein Distance Algorithm

$n \leftarrow$ length of $s$
$m \leftarrow$ length of $t$
**if** $n = 0$ **then**
  **return** $m$
**if** $m = 0$ **then**
  **return** $n$
Construct a matrix containing 0..m rows and 0..n columns.
Initialize the first row to 0..n.
Initialize the first column to 0..m.

Examine each character of s (i from 1 to n).
Examine each character of t (j from 1 to m).

if s[i] equals t[j], the cost is 0.
if s[i] doesn't equal t[j], the cost is 1.

Set cell d[i,j] of the matrix equal to the minimum of:
a. The cell immediately above plus 1:d[i-1,j]+1.
b. The cell immediately to the left plus 1:d[i,j-1]+1.
c. The cell diagonally above and to the left plus the cost: d[i-1, j-1]+cost.

The distance is found in cell d[n,m].

This metric was used in the behaviour characterisation, specifically assessing the cylinder's trajectory to effectively quantify the dissimilarity between different trajectory sequences. By applying the Levenshtein distance, it becomes possible to determine the novelty of each behaviour, distinguishing which trajectories shows more unique characteristics compared to others. This approach enables a precise evaluation of novelty, crucial for optimizing the swarm robotics' performance by identifying and encouraging the exploration of new, effective strategies.

### 3.1.8 Behaviour characterisation

The research evaluated three distinct behaviour characterisations for use in the novelty search archive:

1. **Cylinder's trajectory**: In this characterisation, the behaviour is represented by the set of explored squares during the simulation. Each square is identified by a unique ID, ensuring that there are no duplicates within the behaviour variable. This method captures the trajectory of the cylinder as it moves through the environment. See Figure 3.7.

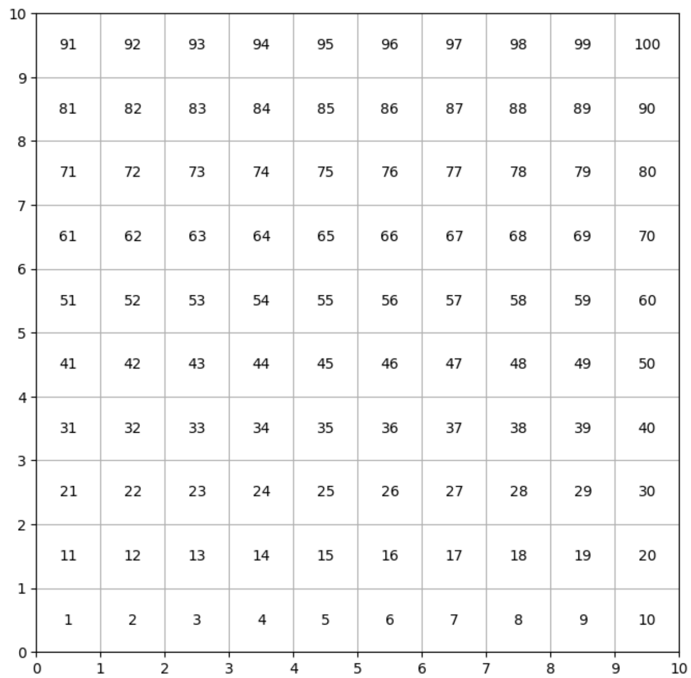



**Figure 3.7:** Map grid.

2. **Final cylinder coordinates**: Here, the behaviour is characterized by the cylinder's final coordinates. The coordinates are represented store in a tuple in order X, Y. The Euclidean distance between the candidates dictates its novelty and if it is added to the archive.

3. **Average distance between robots and cylinder's trajectory or final position**: This characterisation involves the average distance between the robots and either the cylinder's trajectory or the distance to the desired position. Distances between the robots are recorded every 50 simulation steps, resulting in a behaviour vector. The first value in this vector is the average distance between the robots throughout the simulation, while the second value represents either the trajectory of the cylinder or the normalized difference between the cylinder's final and desired positions.

These behaviour characterisations are informed by related works, specifically those by Gomes et al. [12] and Lehman and Stanley [10] and are integral to testing the effectiveness of novelty search in this research context.

### 3.1.9 Evolutionary algorithms configuration

Both fitness-based and random evolution employed the proposed Genetic Algorithm (GA). In these methods, the best candidate is determined by the highest fitness score from the final population. The experiments conducted utilized CTRNN and FNN as the controller, maintaining a consistent set-up across tests. The GA parameters for both novelty-based and fitness-based searches included 200 generations a population size of 150, and a mutation rate of 10%, following the guidelines suggested by Gomes et al. [12]. For a comprehensive comparison, all controllers — evolved through novelty-based search, fitness-based search — were evaluated using the proposed fitness function. It is crucial to note that the fitness scores did not influence the novelty search experiments, adhering to the recommendations by Gomes et al. [12].

**Table 3.1:** Experimental settings and parameters.

| | |
|---|---|
| Number of robots | 2 |
| Number of neurons | 38 |
| Number of generations | 200 |
| Population size | 150 |
| Mutation probability | 10% |
| Tournament size | 2 |
| Number of tournaments | 70 |
| Archive size | 400 |
| Simulation steps | 1500 |

# 3.2 Results

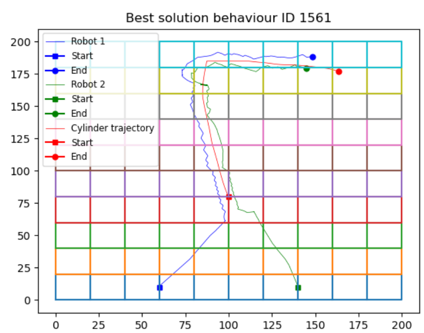
## 3.2.1 CTRNN controller results

The outcomes using the CTRNN as a controller did not meet the initial expectations. Nevertheless, the robots did successfully aggregate, demonstrating that the methods outlined in the related work are reliable. Detailed results can be found in the Appendix .1.

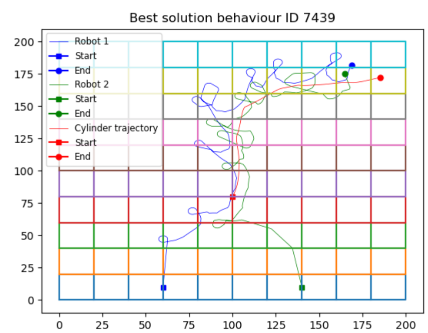
## 3.2.2 Novelty-based search

### 3.2.2.1 Low difficulty map - Final coordinates as behaviour description

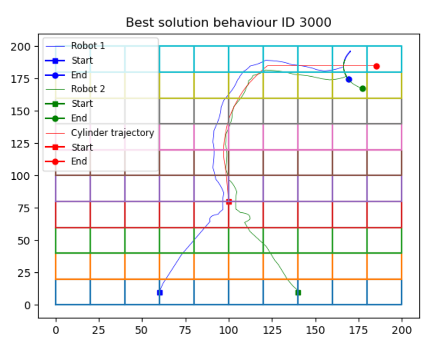
Using the final coordinates of the cylinder as a behaviour characterisation in “low” difficulty map has proven effectiveness in this environment, similar to using trajectory descriptions. However, after multiple tests, characterizing behaviour by coordinates consistently yielded better solutions in fewer generations. Around 35 to 55 generations solve the task. Some of the successful solutions identified are illustrated in Figure 3.8. These solutions were obtained in 6 separate runs of the code.



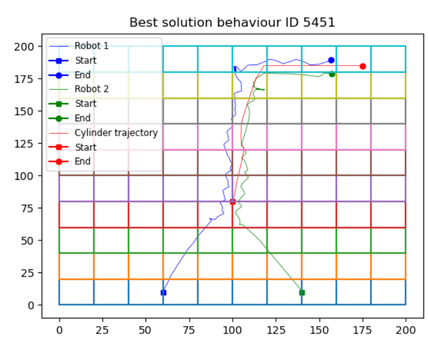

**(a)** Candidate 1561 in final novelty archive.




**(b)** Candidate 7439 in final novelty archive.




**(c)** Candidate 3000 in final novelty archive.

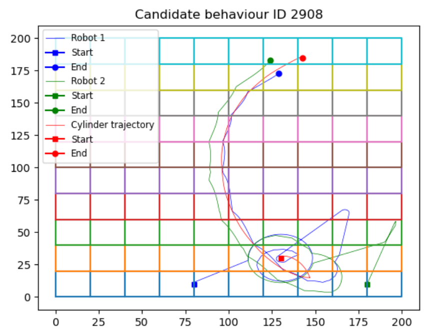



**(d)** Candidate 5451 in final novelty archive.

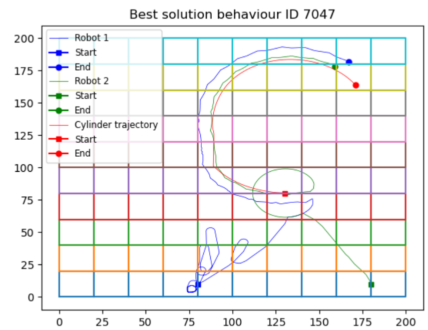
**Figure 3.8:** Best solutions found with coordinates characterisation in low difficulty map for pushing the cylinder to desired position task.

#### 3.2.2.2 Medium difficulty map - Final coordinates as behaviour description

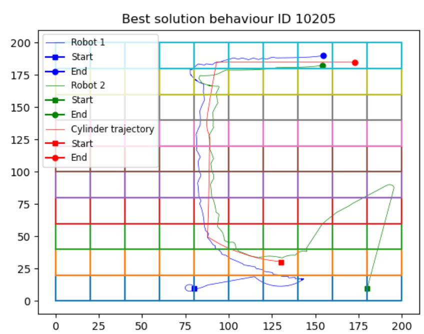
The results from the medium difficulty map illustrate that novelty search can identify multiple behaviours capable of solving the problem. Some of these behaviours are notably straightforward, as in Figure 3.9, where the cylinder typically moves directly to the desired position. During these simulations, the robots consistently push the cylinder without separating, maintaining a cohesive unit throughout.



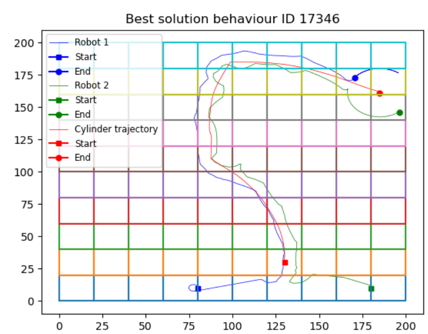

**(a)** Candidate 2908 in final novelty archive.




**(b)** Candidate 7047 in final novelty archive.



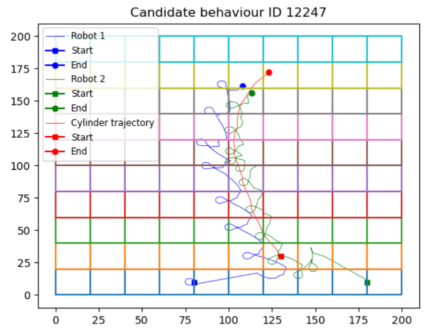

**(c)** Candidate 10205 in final novelty archive.



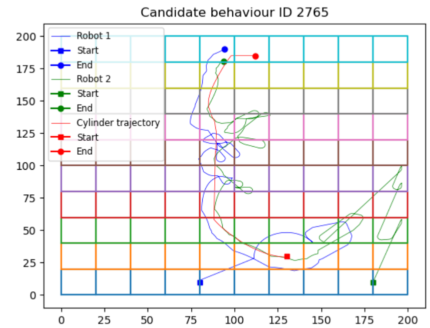

**(d)** Candidate 17346 in final novelty archive.

**Figure 3.9:** Medium difficulty map with coordinates characterisation for pushing the cylinder to desired position task - consistent push.

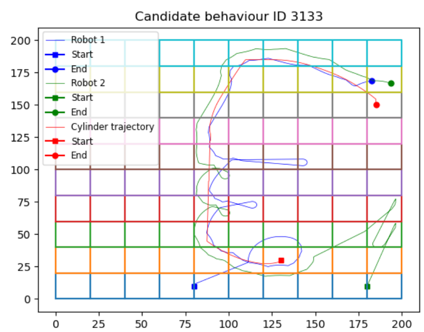
On the other hand, there are behaviours where the robots reach the final position but exhibit a pattern of pushing the cylinder and then moving away during the simulation. These actions demonstrate the robots' ability to navigate obstacles, scan the environment, and relocate the cylinder to continue pushing it toward the desired final position. Figure 3.10 illustrates some example behaviours.



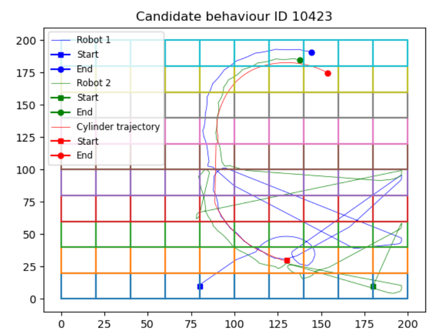

**(a)** Candidate 12247 in final novelty archive.




**(b)** Candidate 2765 in final novelty archive.




**(c)** Candidate 3133 in final novelty archive.



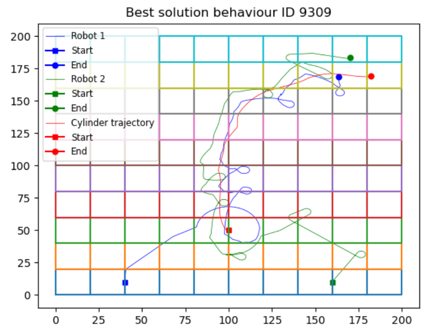

**(d)** Candidate 10423 in final novelty archive.

**Figure 3.10:** Medium difficulty map with coordinates characterisation for pushing the cylinder to desired position task - inconsistent push.

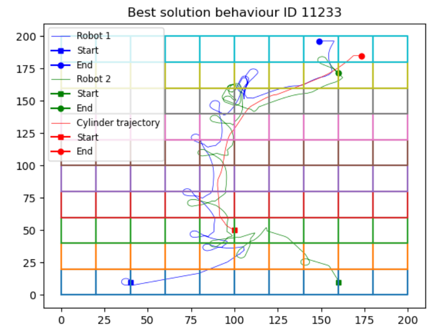
For instance, 3.10b illustrates how Robot 2 locates the cylinder on the map, initiates pushing, and then navigates the cylinder through a gap after encountering a corner. Similarly, Figure 3.10c demonstrates comparable behaviour with Robot 1 managing a corner obstacle. Additionally, Figure 3.10c showcases the robots' ability to simultaneously push the cylinder and scan the environment, ultimately guiding the cylinder to the desired final position successfully.

#### 3.2.2.3 Medium/high difficulty map - Final coordinates as behaviour description

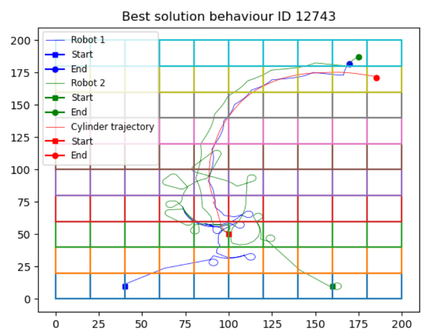
When the large vertical walls are removed from the environment, it might be expected that the algorithm would take significantly longer to find an optimal solution. However, the algorithm demonstrates its robustness by locating an optimal solution in around 150 generations. The evolution of candidates during the simulation shows extensive exploration of the solution space. Given the map's deceptive nature, the novelty search effectively rewards uncommon behaviours that lead to the task's solution. Furthermore, The evolution obtained by using novelty search guarantee that a vast extension of the solution space is explored. An example of this evolutionary process is presented in Appendix .3. Figure 3.11 illustrates the best solutions found across four different runs.



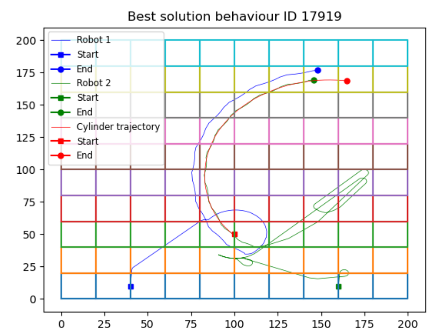

**(a)** Best solution in novelty archive - Candidate 9309.




**(b)** Best solution in novelty archive - Candidate 11233.




**(c)** Best solution in novelty archive - Candidate 12743.



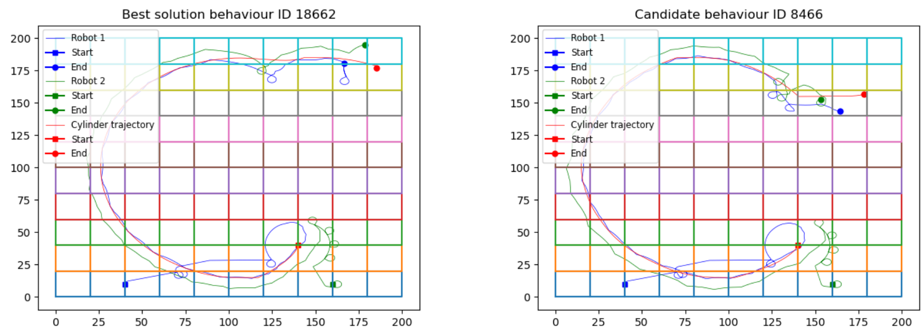

**(d)** Best solution in novelty archive - Candidate 17919.

**Figure 3.11:** Best solutions found with coordinates characterisation in medium-high difficulty map for pushing the cylinder to desired position task.

#### 3.2.2.4 High difficulty map - Final coordinates as behaviour description

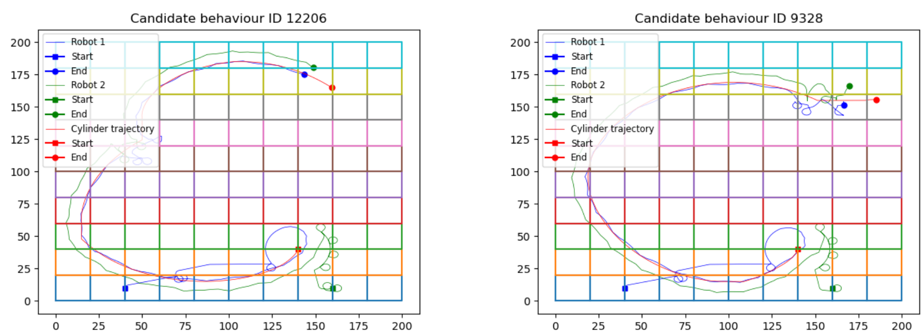
The results for the high-difficulty map are definitive, demonstrating the algorithm's ability to explore a vast portion of the solution space. Additionally, since multiple behaviours can solve the task, proves that novelty search not only drives evolution and problem-solving but also effectively tackles complex challenges. In this specific task and environment, the robots must cluster and push together to successfully move the cylinder. The robots' trajectories indicate that they are able to cluster, locate the cylinder, and coordinate their efforts to push it. However, a detailed analysis of the robots' sensor inputs and motor speeds during the simulation is necessary to fully understand their behaviour.

It is also important to highlight that both behaviour characterisations — the trajectory set and the final cylinder position coordinates — are successful in solving the problem. These characterisations enable the algorithm to effectively explore the solution space and discover various successful strategies. Detailed results and analysis of the trajectory set behaviour characterisation can be found in Appendix .2. Appendix .3 illustrates the evolution throughout generations using final cylinder coordinates as behaviour description. Figure 3.12 illustrates some of the best solutions with final cylinder coordinates as behaviour description discovered during the process.




**(a)** Best solution in novelty archive - Candidate 18662.

**(b)** Candidate 8466 in final novelty archive.




**(c)** Candidate 12206 in final novelty archive.

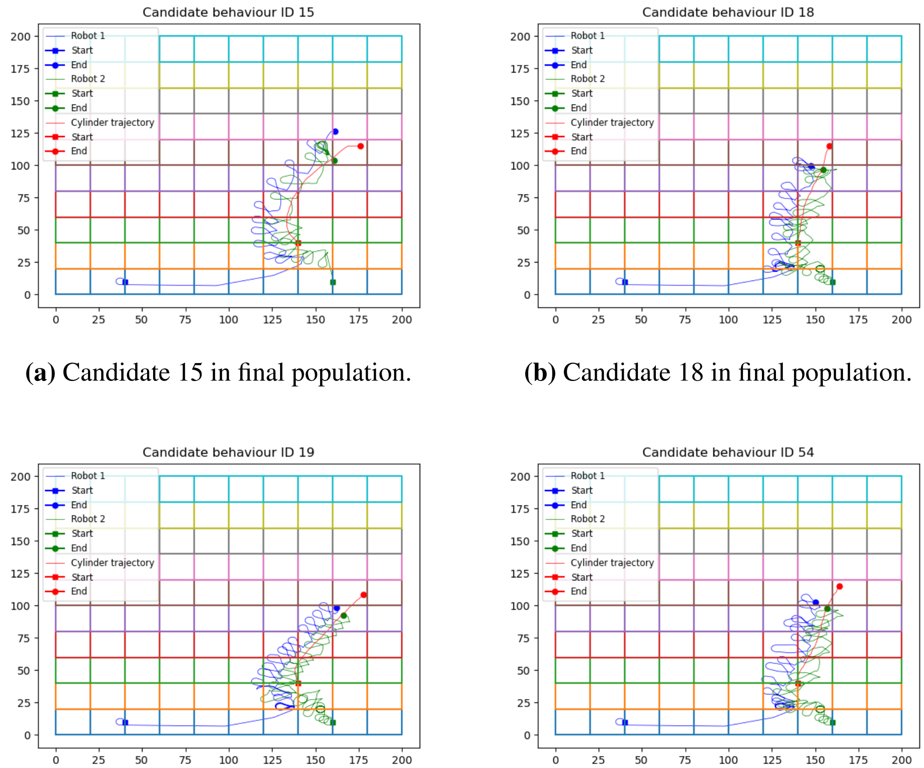
**(d)** Candidate 9328 in final novelty archive.

**Figure 3.12:** Best solutions found with coordinates characterisation in high difficulty map for pushing the cylinder to desired position task.

### 3.2.3 Fitness-based search

In testing fitness-based search on a "high" difficulty map, the results show that this approach is ineffective for deceptive tasks. The algorithm explores various potential solutions, but it tends to converge on local optima because the path to the desired position is characterized by low fitness values. As a result, even if the algorithm begins exploring in the right direction, other "solutions" with higher immediate fitness values follow different direction from the correct path.

The environment is specifically designed such that the correct path to the desired position has low fitness values until it reaches the top left corner, making it a challenging problem for fitness-based search. This design causes the algorithm to get stuck in a "trap" repeatedly rewarding local optima instead of discovering the true solution. Figure 3.13 illustrates examples of candidate solutions that prematurely converge on these predefined local optima.




**(a)** Candidate 15 in final population.

**(b)** Candidate 18 in final population.

**(c)** Candidate 19 in final population.

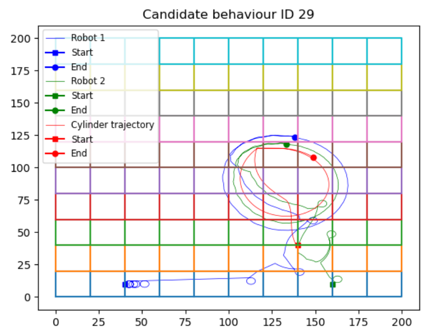
**(d)** Candidate 54 in final population.

**Figure 3.13:** High difficulty map candidates stuck in local optima in final population - Fitness-based search.

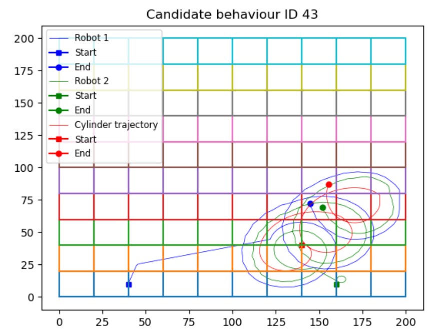
Moreover, fitness-based search demonstrates its limited effectiveness in this complex and deceptive task. As shown in Figure 3.14, the most common behaviour identified through objective-based search reveals the algorithm's struggle to optimize the solution. Even when the algorithm attempts to improve, it tends to favour selections that reinforce local optima because escaping these traps requires favouring lower fitness values, which the algorithm naturally avoids. Consequently, it selects and reproduces candidates that keep the cylinder within the "trap" reinforcing suboptimal solutions.

Alternatively, it is possible to improve the effectiveness of fitness-based search by refining the fitness function. The basic fitness function used may not be optimal for this particular task. Introducing a more sophisticated fitness function could enhance optimization performance. Other option is divide the environment into quadrants and assign higher fitness values to those that lead towards the desired position could guide the algorithm more effectively. However, this approach requires designing new quadrants for each different task, significantly increasing the effort required to configure each environment.

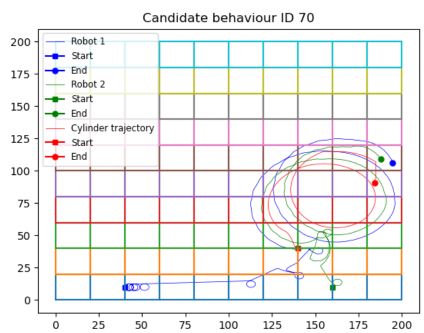
In contrast, the results and flexibility demonstrated by novelty search make it evident that the methods proposed in this research are more effective and superior to objective-based search when dealing with complex and deceptive tasks. Novelty search not only simplifies the process by avoiding the need for task-specific adjustments but also consistently leads to successful outcomes in challenging environments.



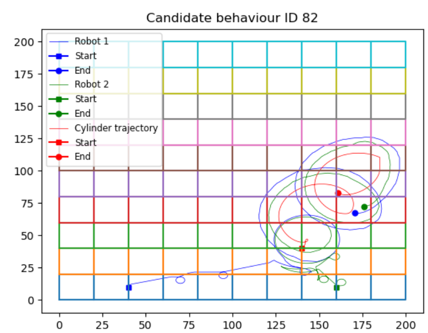

**(a)** Candidate 29 in final population.




**(b)** Candidate 43 in final population.




**(c)** Candidate 70 in final population.




**(d)** Candidate 82 in final population.

**Figure 3.14:** High difficulty map candidates stuck in local optima in final population - Fitness-based search.

### 3.2.4 Behavioural diversity

The novelty search algorithm has demonstrated its ability to effectively explore the solution space. All the candidates in the novelty archive that successfully solved the problem exhibited different robot trajectories, indicating that there are multiple solutions for each problem, and novelty search is successful in discovering them. Unlike fitness-based search, novelty search rewards behaviours that move the cylinder through gaps throughout generation, allowing the exploration of regions of the environment that fitness-based search might omit. These dismissed areas are often associated with low fitness values, causing fitness-based algorithms to overlook them and get trapped in local optima. Thus, novelty search not only solves the problem effectively but also has the potential to uncover multiple successful solutions by exploring a broader portion of the map.

### 3.2.5 Research questions and discussion

#### 3.2.5.1 How does the performance of neural-based controller evolve through novelty search compared to evolved through fitness-based search in the context of specific deceptive task in swarm robotics?

This question emerged due to the lack of research on the evolution of neural-based controllers using novelty search in deceptive tasks within swarm robotics. Previous studies, such as [12] have explored different tasks, focusing primarily on robot behaviours within a map. However, this research aimed to assess the effectiveness of novelty search in more complex and deceptive tasks, where robots must collaborate and interact with the environment to find a solution.

The results clearly show that novelty-based evolution outperforms fitness-based search. Controllers evolved through novelty search successfully found solutions to the problem, as evidenced by the generational plots showing how candidates explored the map and rewarded behaviours that, given the correct behaviour characterisation, led to successful outcomes. While novelty search was not directly compared to fitness-based search in the first three maps, the objective was to evaluate both strategies in the most complex and deceptive environment proposed. Novelty search consistently proved to be the superior choice, often finding good solutions in fewer generations than expected across all maps. Its speed and consistency in solving problems throughout the experiments were notable, with the robots exhibiting different behaviours to reach the solutions.

On the contrary, fitness-based search was applied only in the hardest map, where its performance was as expected — it failed to find any solution within the same number of runs as novelty search. The algorithm frequently became stuck in the designed local optima because it naturally dismissed candidates with low fitness and rewarded those leading to local optima. While fitness-based search could potentially be improved by using a more sophisticated fitness function or by creating a grid that assigns high fitness values to movements in the correct direction, these adjustments would require significant additional work for each environment. This makes fitness-based search more tedious and less flexible compared to novelty search.

In conclusion, the performance of neural-based controllers evolved through novelty search significantly outperforms those evolved through fitness-based search in the context of specific deceptive tasks in swarm robotics. Novelty search not only consistently finds effective solutions by encouraging exploration and rewarding unique behaviours but also does so with greater efficiency across a variety of environments. In contrast, fitness-based search struggles with deceptive tasks, often becoming trapped in local optima and requiring extensive modifications to the fitness function to approach the versatility of novelty search. Therefore, for complex and deceptive tasks that require collaboration and environmental interaction, novelty search proves to be a more robust and flexible evolutionary strategy, yielding superior results in evolving neural-based controllers.

#### 3.2.5.2 What unique behavioural patterns emerge in homogeneous robot swarms when using novelty search compared to fitness-based approach?

It can be obtained different solutions using novelty search as evidenced in [12] where novelty search provided various solutions for aggregation tasks that were as effective as fitness-based search. Therefore, in this research, it was expected to get more than one solution that achieves the goal.

In the high-difficulty map experiments, novelty search demonstrated a significant advantage over fitness-based methods by dynamically adapting to the environment's complexity and deception. Unlike fitness-based search, which strictly adheres to predefined fitness criteria often unsuitable for complex scenarios, novelty search encouraged a diverse range of strategies by rewarding unique and effective behaviours. This led to the discovery of multiple viable solutions, showcasing the algorithm's ability to handle complex tasks where traditional methods do not work as expected. The generational development under novelty search displayed a continual evolution of strategies, adjusting to environmental challenges and obstacles efficiently, a clear contrast to the static and often counter-productive approach seen in fitness-based search.

Moreover, while fitness-based search struggled in the high-difficulty environment, often getting trapped in local optima without real progress towards the task's completion, novelty search thrived under these conditions. It consistently led to the emergence of innovative problem-solving behaviours, exploiting the full potential of the robotic swarm's capabilities. This was evident in the robots ability to collaboratively navigate and manipulate objects in the environment, a complex behaviour that fitness-based search failed to develop due to its constrained exploration scope. Thus, novelty search proved not only more effective but also more efficient in fostering adaptive behaviours necessary for success in deceptive and dynamic tasks.

#### 3.2.5.3 How effective is the behaviour characterisation proposed with novelty search in evolving neural-based controller for collaborative tasks?

In the context of this research on swarm evolutionary robotics, there were explored various behaviour characterisations to determine the most effective strategy for enabling collaboration between robots in moving a heavy object — beyond the capability of a single robot. The approach is based on methodologies from Gomes et al. and Lehman and Stanley, focusing on behaviour characterisation tailored to the experimental goals [12] [10].

Three distinct behaviour characterisations were tested, each designed to optimize the collaborative task. The first approach, which tracked the average distance between robots and the cylinder's trajectory or final coordinates, proved to be the least successful. The results suggest that the relationship between robot positions and cylinder state is not sufficiently direct, indicating that alternative strategies, such as multi-objective optimization, might be necessary to leverage this type of behaviour characterisation effectively.

On the other hand, characterizing behaviour based on the cylinder's trajectory was more productive. In environments classified as "easy", "medium", "medium-hard", this method yielded optimal solutions in four out of six trials, typically within 50 to 150 generations. However, in more challenging "high" difficulty maps, the approach occasionally failed to consistently reward behaviours that facilitated the successful completion of the task, as detailed in the evolution and final novelty archive plots (See appendix .2). This inconsistency underscores potential areas for enhancement, possibly through refined or alternative behavioural characterisations.

The behaviour characterisation tracking the final coordinates of the cylinder proved to be the most effective across all environments, rapidly identifying optimal solutions within 25 to 120 generations for the "easy", "medium", and "medium-hard" maps. Particularly in the "high" difficulty setting, it outperformed the trajectory-based approach by more consistently leading candidates in the correct direction to solve the problem. Generational evolution data further confirmed that the population consistently converged towards cylinder positions that facilitated successful outcomes, as detailed in appendix .3. This method, which also straightforwardly integrates the final cylinder position into the fitness estimation of candidates, emerged as the most practical and efficient behaviour characterisation for the task.

## 3.3 Conclusion

This study demonstrates that novelty search outstrips fitness-based search in the evolution of neural-based controllers for deceptive swarm robotics tasks. Through comparative analysis, novelty search not only facilitated more dynamic and adaptive strategies but also showcased a consistent ability to overcome complex challenges and discover effective solutions. The exploration across multiple environments confirmed that novelty search is flexible and explorative nature is crucial for tasks requiring complex collaboration and environmental interaction. The robustness of novelty search in handling diverse and deceptive scenarios marks it as a superior evolutionary strategy, suited for advancing the capabilities of swarm robotics.

Furthermore, the effectiveness of behaviour characterisations using the final coordinates of the cylinder highlights the practical advantages of novelty search in real-world applications. This approach not only ensured rapid convergence towards successful outcomes but also demonstrated ease of implementation, making it an effective tool in evolutionary robotics. By enabling more accurate and efficient problem-solving methods, the findings from this research provide a solid foundation for future investigations into the optimization of collaborative robotic tasks, emphasizing the potential of novelty search in the field.

# Chapter 4

# Conclusion

## 4.1 Review of research questions and findings

This dissertation confirms that novelty search surpass fitness-based search in the evolution of neural-based controllers within deceptive swarm robotics tasks. Novelty search not only discovers diverse and effective solutions efficiently but also enhances the adaptability and innovation of robotic behaviours. This adaptability is crucial in complex environments where traditional methods may fail due to their rigid adherence to predefined fitness criteria.

Furthermore, findings highlight that the final coordinates of the cylinder as a behaviour characterisation greatly outperform other methods. This approach consistently led to quicker and more effective problem-solving across various difficulty levels, proving its efficacy in facilitating successful outcomes in collaborative robotic tasks.

Overall, the research underscores the superiority of novelty search in managing complex, deceptive tasks that require advanced interaction and collaboration among robots. These insights contribute to further exploration into advanced behaviour characterisations and their potential to optimize swarm robotics systems.

## 4.2 Recommendations

In evolutionary swarm robotics, objective-based search is commonly applied to a range of tasks. However, certain environments may present complexities and deceptions that challenge the effectiveness of fitness-based search, necessitating complex fitness functions. This novel approach, neuroevolutionary swarm robotics with novelty search, is designed specifically to tackle these complex and deceptive tasks.

The success of this approach relies on the appropriate characterisation of behaviour, which is crucial for guiding agents toward task-specific actions. If the behaviour characterisation does not align closely with the required actions within the environment, the algorithm may fail to identify and promote the desired behaviours. In the context of this study, where robots are tasked with collectively moving a heavy object that cannot be moved by a single robot, it is essential to focus on characterizing the behaviour of the cylinder. This ensures that the robots must cooperate to move the cylinder, preventing scenarios where one robot might be sufficient, thus negating the need for collaborative effort.

It is also recommended to evaluate the algorithm across various environments, from simple to complex. This strategy allows for the fine-tuning of novelty search and genetic algorithm parameters in a progressive manner, ensuring the algorithm's effectiveness across varying levels of difficulty. Successfully addressing a high-difficulty environment suggests the algorithm's capability to handle simpler scenarios, facilitating broader applicability without significant modifications to the source code.

Furthermore, the choice and implementation of sensors are critical. In this project, the utilization of each robot's camera and infra-red sensor is vital for detecting the cylinder and other robots within the environment. These sensors provide crucial data that empower the robotic controllers to make informed decisions, highlighting the importance of sensor integration in supporting context-aware robotic behaviours.

## 4.3 Limitations and further research

The iterative simulation of the novelty-based search algorithm across diverse environments and its comparison with fitness-based search on the most challenging map revealed key insights into solving complex problems in swarm robotics. A critical factor was the size of the novelty archive, with a final value of 400, the algorithm successfully resolved the most difficult challenges. This suggests that expanding the novelty archive further may potentially uncover even more solutions.

In this implementation, the novelty algorithm was used in its most basic form. Enhancing it with sparseness could likely increase its effectiveness. Moreover, although a hybrid approach combining novelty and fitness-based searches has shown promise in related works, it was not explored in this experiment due to specific research objectives. This represents a potential area for future investigation.

To gain a deeper understanding of the robots' behaviour, a detailed analysis of the interaction between motor speeds and sensor inputs is essential. Currently, without such analysis, interpretations of how robots respond to environmental stimuli remain speculative. It is advisable to conduct thorough examinations and generate detailed visualizations of sensor and motor data across various simulations. This approach will enable a more accurate and empirical description of the robots' behaviour, enhancing the interpretability and applicability of the findings.

# Bibliography


[1] M. Beekman, G. A. Sword, and S. J. Simpson, *Biological Foundations of Swarm Intelligence*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 3–41. [Online]. Available: https://doi.org/10.1007/978-3-540-74089-6_1

[2] S. Garnier, J. Gautrais, and G. Theraulaz, "The biological principles of swarm intelligence," *Swarm Intelligence*, vol. 1, no. 1, pp. 3–31, jun 2007. [Online]. Available: https://doi.org/10.1007/s11721-007-0004-y

[3] H. Ahmed and J. Glasgow, "Swarm intelligence: concepts, models and applications," *School Of Computing, Queens University Technical Report*, 2012.

[4] M. M. Shahzad, Z. Saeed, A. Akhtar, H. Munawar, M. H. Yousaf, N. K. Baloach, and F. Hussain, "A review of swarm robotics in a nutshell," *Drones*, vol. 7, no. 4, 2023. [Online]. Available: https://www.mdpi.com/2504-446X/7/4/269

[5] L. Bayındır, "A review of swarm robotics tasks," *Neurocomputing*, vol. 172, pp. 292–321, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231215010486

[6] Y. Mohan and S. G. Ponnambalam, "An extensive review of research in swarm robotics," in *2009 World Congress on Nature I& Biologically Inspired Computing (NaBIC)*, 2009, pp. 140–145.

[7] A. J. C. Sharkey, "Swarm robotics and minimalism," *Connection Science*, vol. 19, no. 3, pp. 245–260, 2007. [Online]. Available: https://doi.org/10.1080/09540090701584970

[8] C. Darwin, *On the Origin of Species in From So Simple a Beginning: The Four Great Books of Charles Darwin*, E. O. Wilson, Ed. New York: W. W. Norton & Company, 2006.

[9] S. Doncieux, A. Laflaquière, and A. Coninx, "Novelty search: a theoretical perspective," ser. GECCO '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 99–106. [Online]. Available: https://doi.org/10.1145/3321707.3321752

[10] J. Lehman and K. Stanley, "Abandoning objectives: Evolution through the search for novelty alone," *Evolutionary computation*, vol. 19, pp. 189–223, 06 2011.

[11] J. Lehman and K. O. Stanley, "Efficiently evolving programs through the search for novelty," in *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 837–844. [Online]. Available: https://doi.org/10.1145/1830483.1830638

[12] J. Gomes, P. Urbano, and A. L. Christensen, "Evolution of swarm robotics systems with novelty search," *Swarm Intelligence*, vol. 7, no. 2, pp. 115–144, sep 2013. [Online]. Available: https://doi.org/10.1007/s11721-013-0081-z

[13] S. Doncieux, G. Paolo, A. Laflaquière, and A. Coninx, "Novelty search makes evolvability inevitable," in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, ser. GECCO '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 85–93. [Online]. Available: https://doi.org/10.1145/3377930.3389840

[14] S. Doncieux, N. Bredeche, J.-B. Mouret, and A. E. G. Eiben, "Evolutionary robotics: What, why, and where to," *Frontiers in Robotics and AI*, vol. 2, 2015. [Online]. Available: https://www.frontiersin.org/journals/robotics-and-ai/articles/10.3389/frobt.2015.00004

[15] J. Gomes, P. Mariano, and A. Christensen, "Systematic derivation of behaviour characterisations in evolutionary robotics," in *Artificial Life 14: Proceedings of the Fourteenth International Conference on the Synthesis and Simulation of Living Systems*, ser. ALIFE. The MIT Press, Jul. 2014. [Online]. Available: http://dx.doi.org/10.7551/978-0-262-32621-6-ch036

[16] R. J. Alattas, S. Patel, and T. M. Sobh, "Evolutionary modular robotics: Survey and analysis," *Journal of Intelligent & Robotic Systems*, vol. 95, no. 3, pp. 815–828, sep 2019. [Online]. Available: https://doi.org/10.1007/s10846-018-0902-9

[17] D. Cliff, P. Husbands, and I. Harvey, "Explorations in evolutionary robotics," *Adaptive Behavior*, vol. 2, no. 1, pp. 73–110, 1993. [Online]. Available: https://doi.org/10.1177/105971239300200104

[18] O. Mubin, C. J. Stevens, S. Shahid, A. Al Mahmud, and J.-J. Dong, "A review of the applicability of robots in education," *Journal of Technology in Education and Learning*, vol. 1, no. 209-0015, p. 13, 2013.

[19] M. Hägele, K. Nilsson, J. N. Pires, and R. Bischoff, *Industrial Robotics*. Cham: Springer International Publishing, 2016, pp. 1385–1422. [Online]. Available: https://doi.org/10.1007/978-3-319-32552-1_54

[20] J. Fink, V. Bauwens, F. Kaplan, and P. Dillenbourg, "Living with a vacuum cleaning robot," *International Journal of Social Robotics*, vol. 5, no. 3, pp. 389–408, aug 2013. [Online]. Available: https://doi.org/10.1007/s12369-013-0190-2

[21] S. Paluch, J. Wirtz, and W. H. Kunz, *Service Robots and the Future of Services*. Wiesbaden: Springer Fachmedien Wiesbaden, 2020, pp. 423–435. [Online]. Available: https://doi.org/10.1007/978-3-658-31563-4_21

[22] V. P. Kumar, M. Balasubramanian, and S. J. Raj, "Robotics in construction industry," *Indian Journal of Science and Technology*, vol. 9, no. 23, pp. 1–12, 2016. [Online]. Available: https://d1wqtxts1xzle7.cloudfront.net/93739662/Article19.pdf?1667716024=&response-content-disposition=inline%3B+filename%3DRobotics_in_Construction_Industry.pdf&Expires=1722959871&Signature=FouIIxfaLiYxJfcxI1nupMiZ4jIkg4bG~6Ev4zfagec~CNJLlju2JnxPsfpT-0X1j-AC4iW4Hc1xxgXqES_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

[23] S. Elattar, "Automation and robotics in construction: opportunities and challenges," *Emirates journal for engineering research*, vol. 13, no. 2, pp. 21–26, 2008.

[24] D. Stormont, "Autonomous rescue robot swarms for first responders," in *CIHSPS 2005. Proceedings of the 2005 IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety, 2005.*, 2005, pp. 151–157.

[25] M. Bakhshipour, M. Jabbari Ghadi, and F. Namdari, "Swarm robotics search and rescue: A novel artificial intelligence-inspired optimization approach," *Applied Soft Computing*, vol. 57, pp. 708–726, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494617301072

[26] Y. Tan and Z. yang Zheng, "Research advance in swarm robotics," *Defence Technology*, vol. 9, no. 1, pp. 18–39, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S221491471300024X

[27] J. Gomes and A. L. Christensen, "Generic behaviour similarity measures for evolutionary swarm robotics," in *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 199–206. [Online]. Available: https://doi.org/10.1145/2463372.2463398

[28] L. TÜRKLER, T. Akkan, and L. AKKAN, "Control of swarm robotics in webots with pso," in *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2021, pp. 1–6.

[29] D. Floreano, S. Nolfi, and P. Husbands, "Handbook of robotics chapter 61 evolutionary robotics," 2007.

[30] I. Harvey, P. Husbands, D. Cliff, A. Thompson, and N. Jakobi, "Evolutionary robotics: the sussex approach," *Robotics and Autonomous Systems*, vol. 20, no. 2, pp. 205–224, 1997, practice and Future of Autonomous Agents. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092188909600067X

[31] A. L. Nelson, G. J. Barlow, and L. Doitsidis, "Fitness functions in evolutionary robotics: A survey and analysis," *Robotics and Autonomous Systems*, vol. 57,

no. 4, pp. 345–370, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889008001450

[32] S. Doncieux and J.-B. Mouret, "Behavioral diversity measures for evolutionary robotics," in *IEEE Congress on Evolutionary Computation*, 2010, pp. 1–8.

[33] L. Trujillo, G. Olague, E. Lutton, F. F. de Vega, L. Dozal, and E. Clemente, "Speciation in behavioral space for evolutionary robotics," *Journal of Intelligent and Robotic Systems*, vol. 64, no. 3, pp. 323–351, dec 2011. [Online]. Available: https://doi.org/10.1007/s10846-011-9542-z

[34] F. Venezia, "Design principles of plant photosensory networks : quantitative analysis and modelling of phytochrome dimer dynamics in arabidopsis thaliana," Ph.D. dissertation, 06 2016.

[35] J. Gomes, P. Mariano, and A. Christensen, "Systematic derivation of behaviour characterisations in evolutionary robotics," in *Artificial Life 14: Proceedings of the Fourteenth International Conference on the Synthesis and Simulation of Living Systems*, ser. ALIFE. The MIT Press, Jul. 2014. [Online]. Available: http://dx.doi.org/10.7551/978-0-262-32621-6-ch036

[36] C. Hartland and N. Bredeche, "Evolutionary robotics, anticipation and the reality gap," in *2006 IEEE International Conference on Robotics and Biomimetics*, 2006, pp. 1640–1645.

[37] L. Bayindir and E. Şahin, "A review of studies in swarm robotics," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 15, no. 2, pp. 115–147, 2007.

[38] I. Navarro and F. Matía, "An introduction to swarm robotics," *International Scholarly Research Notices*, vol. 2013, no. 1, p. 608164, 2013. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.5402/2013/608164

[39] B. Khaldi and F. Cherif, "An overview of swarm robotics: Swarm intelligence applied to multi-robotics," *International Journal of Computer Applications*, vol. 126, no. 2, 2015.

[40] N. Nedjah and L. S. Junior, "Review of methodologies and tasks in swarm robotics towards standardization," *Swarm and Evolutionary Computation*, vol. 50, p. 100565, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210650217308398

[41] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, mar 2013. [Online]. Available: https://doi.org/10.1007/s11721-012-0075-2

[42] M. Dorigo, "Swarm-bot: an experiment in swarm robotics," in *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.*, 2005, pp. 192–200.

[43] J. Gomes, P. Mariano, and A. L. Christensen, "Devising effective novelty search algorithms: A comprehensive empirical study," in *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 943–950. [Online]. Available: https://doi.org/10.1145/2739480.2754736

[44] E. Meyerson, J. Lehman, and R. Miikkulainen, "Learning behavior characterizations for novelty search," in *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, ser. GECCO '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 149–156. [Online]. Available: https://doi.org/10.1145/2908812.2908929

[45] I. Fister, A. Iglesias, A. Galvez, J. Del Ser, E. Osaba, I. Fister, M. Perc, and M. Slavinec, "Novelty search for global optimization," *Applied Mathematics and Computation*, vol. 347, pp. 865–881, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0096300318310269

[46] J. Gomes, P. Urbano, and A. L. Christensen, "Progressive minimal criteria novelty search," in *Advances in Artificial Intelligence – IBERAMIA 2012*, J. Pavón, N. D. Duque-Méndez, and R. Fuentes-Fernández, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 281–290.

[47] J. Lehman and K. O. Stanley, "Revising the evolutionary computation abstraction: minimal criteria novelty search," in *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 103–110. [Online]. Available: https://doi.org/10.1145/1830483.1830503

[48] J. Lehman, K. O. Stanley *et al.*, "Exploiting open-endedness to solve problems through the search for novelty." in *ALIFE*, 2008, pp. 329–336.

[49] K. O. Stanley, "Efficient evolution of neural networks through complexification," Ph.D. dissertation, 2004. [Online]. Available: https://sussex.idm.oclc.org/login?url=https://www.proquest.com/dissertations-theses/efficient-evolution-neural-networks-through/docview/305130627/se-2

[50] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, 2002.

[51] M. Galassi, N. Capodieci, G. Cabri, and L. Leonardi, "Evolutionary strategies for novelty-based online neuroevolution in swarm robotics," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2016, pp. 002 026–002 032.

[52] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, M. Birattari, L. M. Gambardella, and M. Dorigo, "ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems," *Swarm Intelligence*, vol. 6, no. 4, pp. 271–295, 2012.

[53] O. Robotics, "Gazebo: Robot simulator," https://gazebosim.org/features, 2024, accessed: 2024-08-06.

[54] NVIDIA, "Isaac sim: Robotics simulation and synthetic data," https://developer.nvidia.com/isaac/sim, 2024, accessed: 2024-08-06.

[55] E. Rohmer, S. P. N. Singh, t. f. V.-R. a. V. M. Freese", and S. R. S. Framework, in *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2013, www.coppeliarobotics.com".

[56] C. Ltd., "Webots: Open source robot simulator," https://cyberbotics.com/, 2024, accessed: 2024-08-06.

[57] E. Community, "Enki: A fast 2-d robot simulator," https://github.com/enki-community/enki, 2024, accessed: 2024-08-06.

[58] S. M. Thede, "An introduction to genetic algorithms," *Journal of Computing Sciences in Colleges*, vol. 20, no. 1, pp. 115–123, 2004.

[59] C. Johnson, "Evolution as algorithm," PowerPoint presentation, Brighton, 2023.

[60] J. A. Fernandez-Leon, G. G. Acosta, and M. A. Mayosky, "Behavioral control through evolutionary neurocontrollers for autonomous mobile robot navigation," *Robotics and Autonomous Systems*, vol. 57, no. 4, pp. 411–419, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889008000936

[61] J. A. F. León1, O. E. Goñi, G. G. Acosta, and M. A. Mayosky, "Neuro-controllers, scalability and adaptation," pp. 1279–1288, 2004. [Online]. Available: https://d1wqtxts1xzle7.cloudfront.net/70869659/Documento_completo-libre.pdf?1635668135=&response-content-disposition=inline%3B+filename%3DNeuro_Controllers_scalability_and_adapta.pdf&Expires=1712891479&Signature=P7kEgmND1qyOik2q9ENWu38RY3UfGv-9v1czEEg8f-49F~KQxLf50QB23be8FY5o6Sm7sMchkE _&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

[62] A. Kirillov, "Annt: Feed forward fully connected neural network," https://www.codeproject.com/Articles/1261763/ANNT-Feed-Forward-Fully-Connected-Neural-Networks, 2018, accessed: 2024-08-07.

[63] GCtronic, "Minidoc web," https://www.gctronic.com/files/miniDocWeb.pdf, accessed: 2024-08-14.

[64] D. K. Po, "Similarity based information retrieval using levenshtein distance algorithm," *Int. J. Adv. Sci. Res. Eng*, vol. 6, no. 04, pp. 06–10, 2020.

[65] R. D. Beer, "Parameter space structure of continuous-time recurrent neural networks," *Neural Computation*, vol. 18, no. 12, pp. 3009–3051, 2006.

[66] C. Johnson, "Alife_2023_l9_ dynamical systems_ctrnn," PowerPoint presentation, Brighton, 2023.

# .1 Appendix A

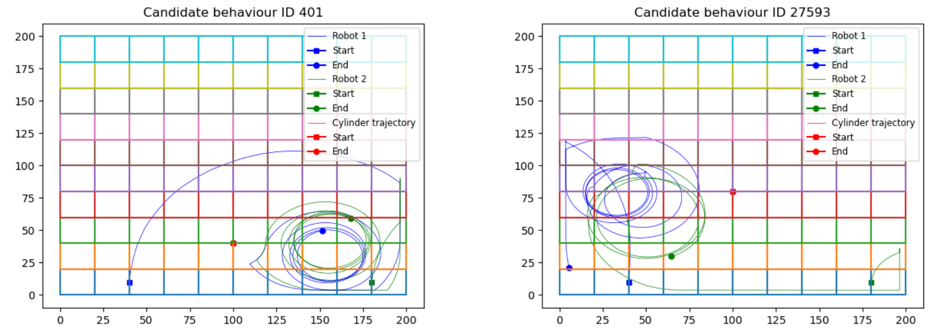
Continuous Time Recurrent Neural Network (CTRNN) is a type of neural network model widely used in the study of dynamical systems due to their simplicity and dynamical universality. A CTRNN is defined by the differential equation:

$$\tau_i \dot{y}_i = -y_i + \sigma\left(\sum_{j=1}^{N} w_{ij} y_j + \theta_i + I_i\right) \tag{1}$$

Where $\tau_i$ represents the time constant, $N$ are the nodes in the network, $y_i$ and $y_j$ refer to the nodes, $w_{ij}$ is the weight from neuron $j$ to neuron $i$, $\sigma$ is the activation logistic sigmoid function, $\theta_i$ the bias, and $I_i$ the internal input [65][66]. Each of these parameters can be adjusted to tune the network's behaviour, making a CTRNN highly flexible.

CTRNNs are particularly suitable for research in swarm robotics and evolutionary robotics because they can generate a wide variety of dynamical behaviours, including complex, periodic, and chaotic patterns. This flexibility is crucial for tasks that involve adaptive and emergent behaviours, which are common in swarm robotics. The universal approximation property of CTRNN ensures that they can approximate any continuous dynamical system to any desired accuracy [65][66]. This capacity allows for the modelling and simulation of the collaborative behaviours observed in robotic swarms, making CTRNNs an ideal choice for implementing and testing algorithms like novelty search in deceptive environments [65][66]. The ability of CTRNNs to evolve complex behaviours through evolutionary algorithms further enhances their applicability in robotics research [65].

In this experimental implementation, CTRNN controller did not prove to be an effective option. The observed behaviours indicated that, despite the robots aggregating, they were not adequately responsive to inputs from the cylinder and the camera. Multiple configurations were tested in an effort to optimize the CTRNN's performance. However, these attempts were unsuccessful. The likely cause of the CTRNN's ineffectiveness is the drastic variance in camera inputs between the cylinder, the background, and the walls. This suggests that the CTRNN requires an image preprocessing step to stabilize and clarify the visual inputs, making them more suitable for processing by the neural network. Therefore, a FNN was chosen as the controller for the experimentation.



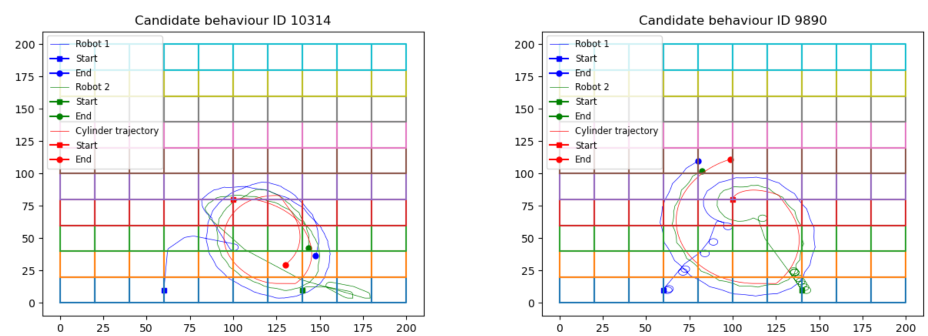

**(a)** Candidate 401 in final novelty archive. **(b)** Candidate 27593 in final novelty archive.

**Figure 1:** Aggregation with CTRNN.

# .2 Appendix B

### .2.0.1 Low difficulty map - Trajectory set as behaviour description

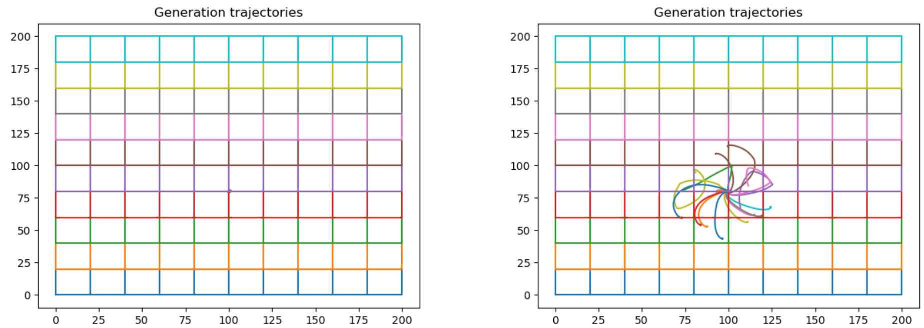
The results of this initial experiment concluded as anticipated. Furthermore, the algorithm demonstrated robustness by overcoming behaviours that did not lead to the desired final position, as illustrated in Figure 2.



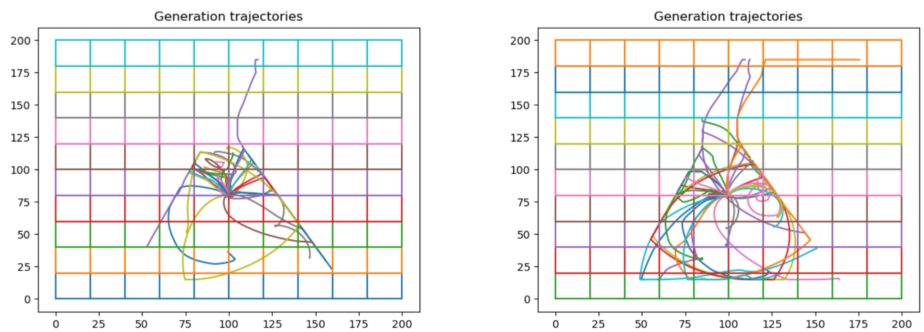

**(a)** Candidate 10314 in final novelty archive. **(b)** Candidate 9890 in final novelty archive.

**Figure 2:** Common behaviours in low difficulty map novelty archive - Trajectory set behaviour description.

Another interesting result using novelty search is illustrated in Figure 3. Which shows the population's evolution throughout the optimization process. The figure plots the trajectory of the cylinder for each candidate, demonstrating that the novelty search algorithm is effectively driving the population towards diverse and successful solutions, as evidenced by the promising results achieved.



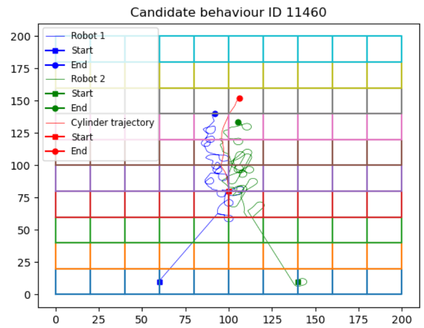

**(a)** Low difficulty map generation 0 example. **(b)** Low difficulty map generation 25 example.



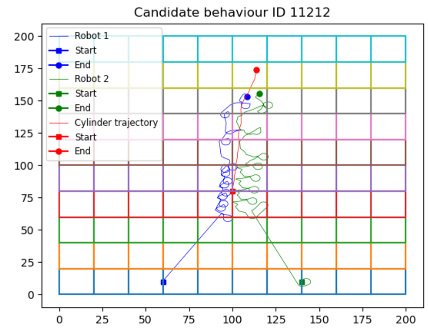

**(c)** Low difficulty map generation 59 example. **(d)** Low difficulty map generation 87 example.

**Figure 3:** Low difficulty map population evolution example - Trajectory set behaviour description.

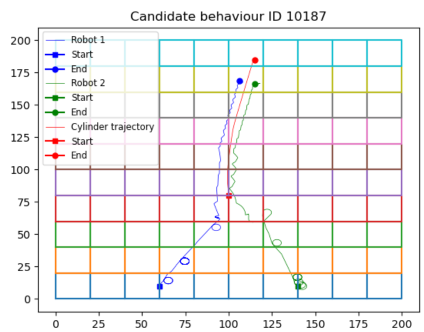
Given the low complexity of the environment and the strategic placement of funnels to guide the robots and cylinder toward the final position, the algorithm resolved the problem in fewer than 90 generations. Additionally, several candidates exhibited behaviours indicative of a successful solution, as shown in Figure 4



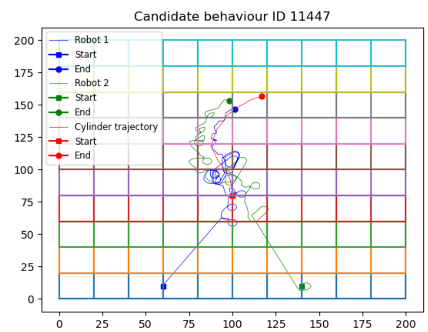

**(a)** Candidate 11460 in final novelty archive.




**(b)** Candidate 11212 in final novelty archive.




**(c)** Candidate 10187 in final novelty archive.




**(d)** Candidate 11447 in final novelty archive.

**Figure 4:** Candidates that led to the solution in low difficulty map - Trajectory set behaviour description.

Multiple solutions emerged during the search. However, the solution with the highest fitness is depicted in Figure 5.

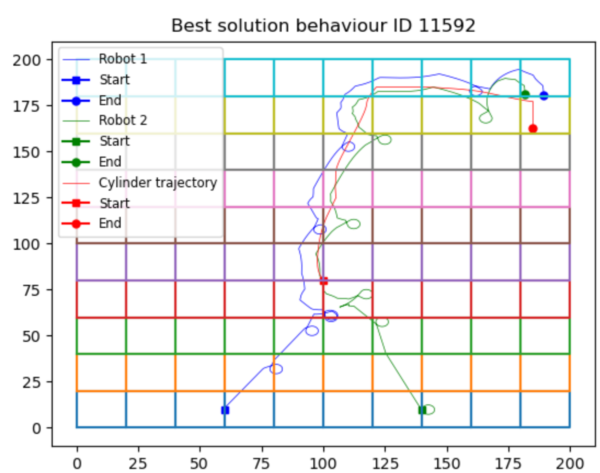
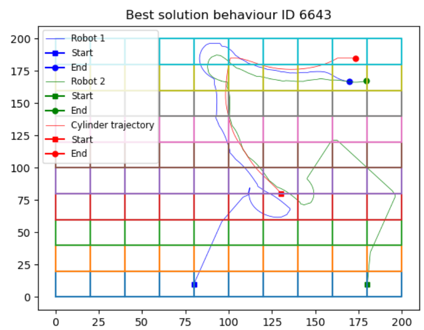


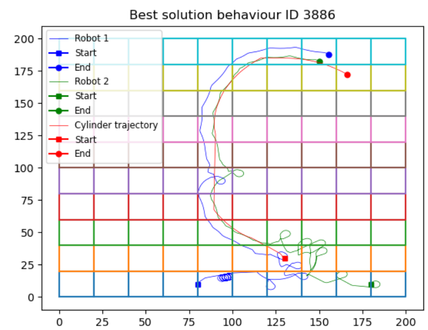

**Figure 5:** Best solution found with trajectory behaviour description.

### .2.0.2 Medium difficulty map - Trajectory set as behaviour description

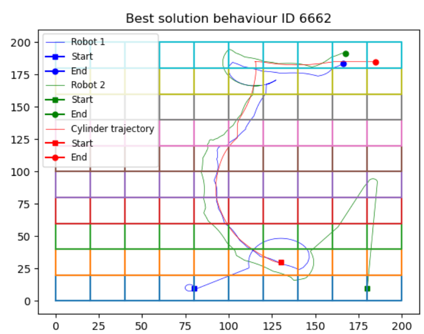
The results from the trajectory set characterization using the medium difficulty map are presented below. Despite the increased environmental complexity, the algorithm successfully identified multiple solutions. Refer to Figure 6 for detailed visualizations of these outcomes.



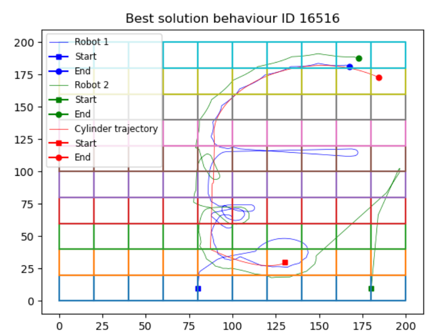

**(a)** Candidate 6643 in final novelty archive.




**(b)** Candidate 3886 in final novelty archive.



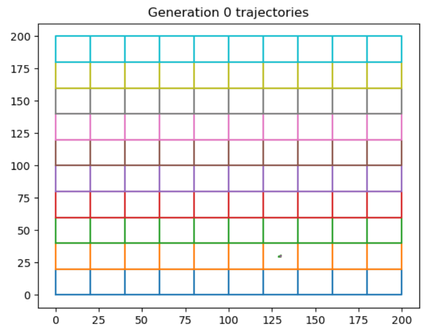

**(c)** Candidate 6662 in final novelty archive.



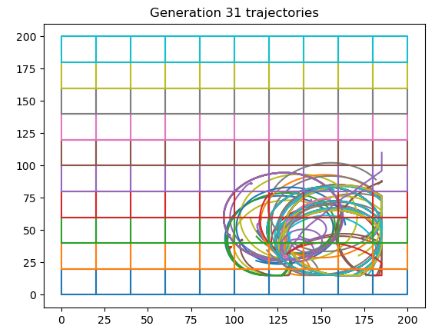

**(d)** Candidate 16516 in final novelty archive.

**Figure 6:** Medium difficulty results - Trajectory set behaviour description.

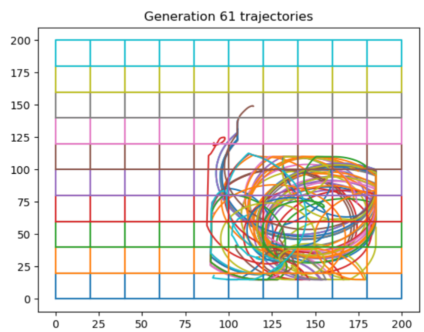
Figure 7 illustrates the evolution of the population for the "medium" difficulty map. Similar to the results with the "easy" difficulty map, the novelty search algorithm effectively drives the evolutionary process, ensuring continuous progress toward improved solutions.



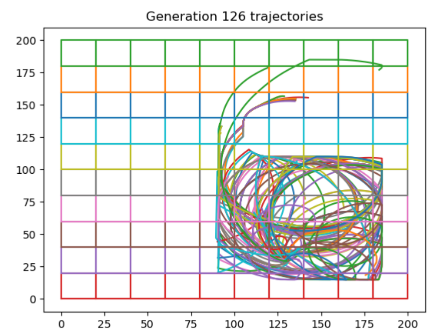

**(a)** Medium difficulty map generation 0 example.




**(b)** Medium difficulty map generation 31 example.




**(c)** Medium difficulty map generation 61 example.

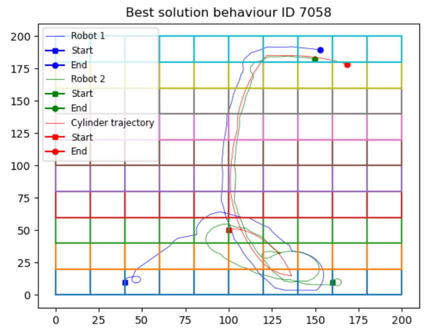



**(d)** Medium difficulty map generation 126 example.

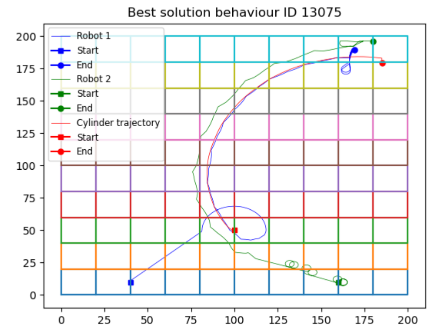
**Figure 7:** Medium difficulty map population evolution example - Trajectory set behaviour description.

### .2.0.3 Medium-High difficulty map - Trajectory set as behaviour description

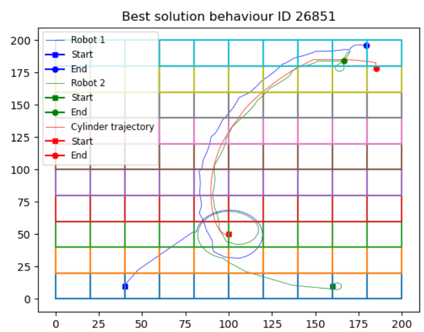
This behaviour description also yielded excellent results, demonstrating that multiple behaviours can lead to a solution, indicating the presence of multiple optima rather than a single global optimum. Figure 8 showcases some of the effective behaviours identified.



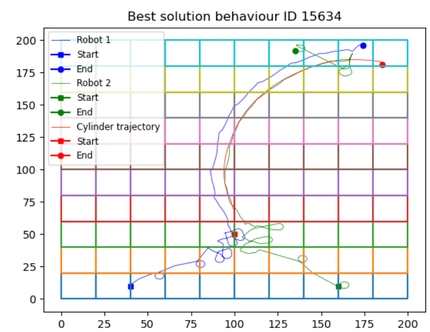

(a) Candidate 7058 in final novelty archive.




(b) Candidate 13075 in final novelty archive.



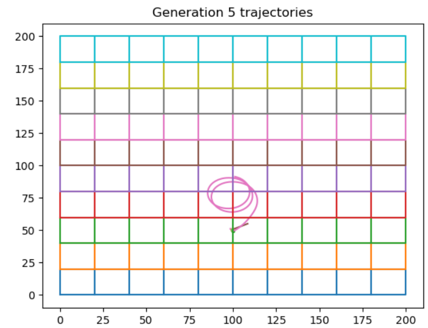

(c) Candidate 26851 in final novelty archive.



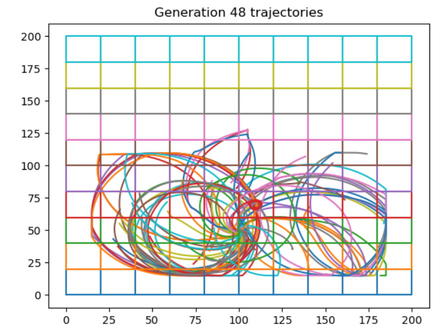

(d) Candidate 15634 in final novelty archive.

**Figure 8:** Candidates that led to the solution in medium difficulty map - Trajectory set behaviour description.

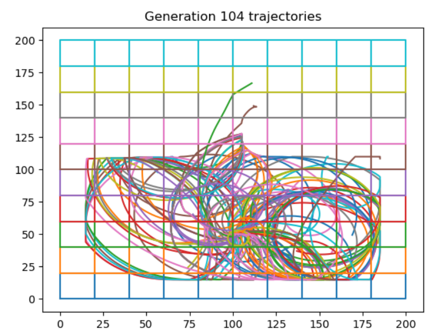
The population evolution throughout generations is shown in Figure 9. Figure 9d show one trajectory as the solution was found within the first candidates in that specific generation.



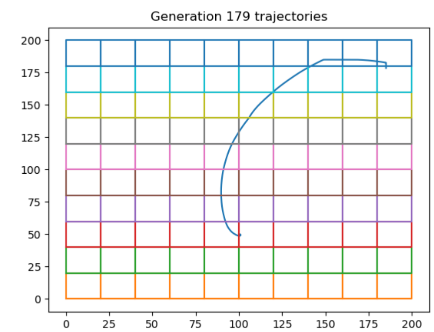

**(a)** Medium-High difficulty map generation 5 example.




**(b)** Medium-High difficulty map generation 48 example.




**(c)** Medium-High difficulty map generation 104 example.

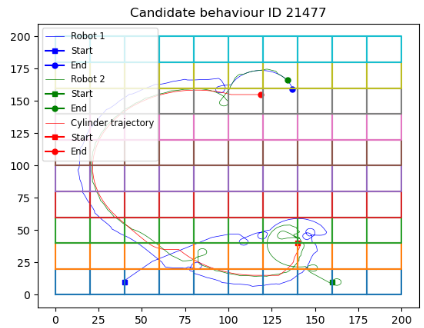



**(d)** Medium-High difficulty map generation 179 example.

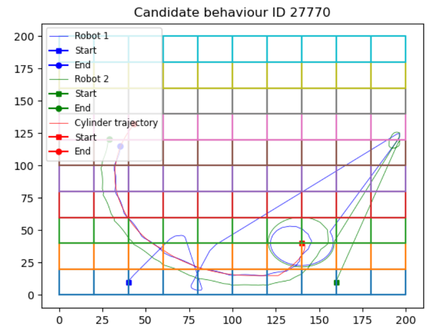
**Figure 9:** Medium-hard difficulty map population evolution example - Trajectory set behaviour description.

### .2.0.4 High difficulty map - Trajectory set as behaviour description

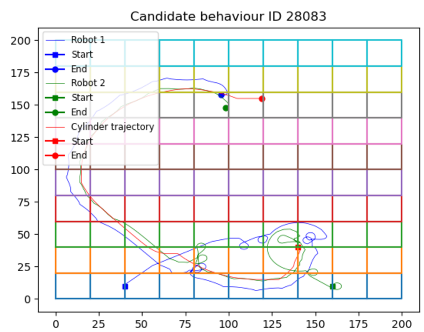
The methods performed exceptionally well on the most challenging map, successfully solving the toughest problem proposed. Although the behaviours did not achieve the final desired position — likely due to insufficient robot speed to push the cylinder to the goal within the time limit — the trajectories of the cylinder and robots indicate they were moving in the correct direction. Figure 10 illustrates these trajectories, showing their alignment toward the goal.



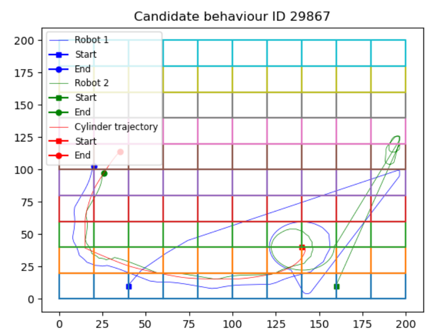

**(a)** Candidate 21477 in final novelty archive.




**(b)** Candidate 27770 in final novelty archive.



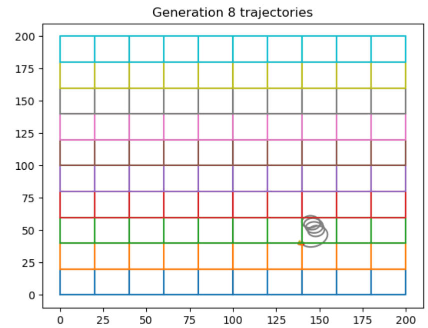

**(c)** Candidate 28083 in final novelty archive.

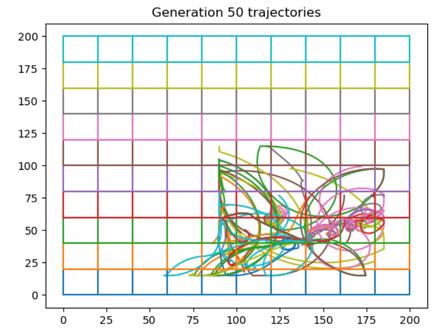



**(d)** Candidate 29867 in final novelty archive.

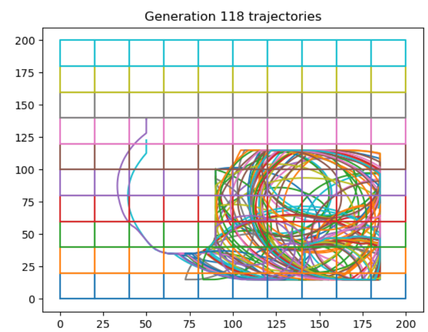
**Figure 10:** Candidates that led to the solution in hard difficulty map - Trajectory set behaviour description.

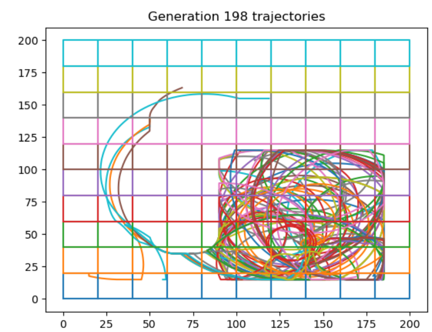
In the "hard" difficulty map. The evolutionary process demonstrated that even though the map is deceptive and the solution is not straightforward (See Figure 11). The novelty search algorithm effectively explores the solution space. It eventually finds the solution, doing so more efficiently than trajectory characterization or fitness-based search methods.




**(a)** Hard difficulty map generation 8 example. **(b)** Hard difficulty map generation 50 example.




**(c)** Hard difficulty map generation 118 example.

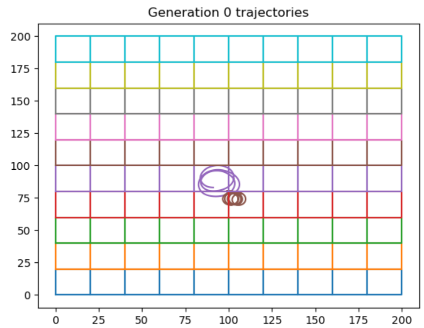
**(d)** Hard difficulty map generation 198 example.

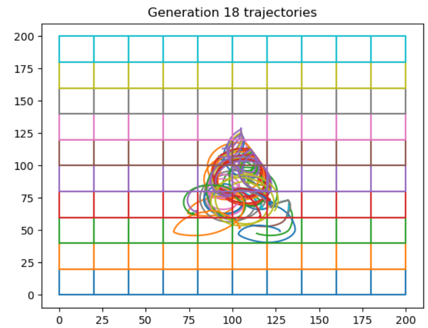
**Figure 11:** Hard difficulty map population evolution example - Trajectory set behaviour description.

# .3 Appendix C

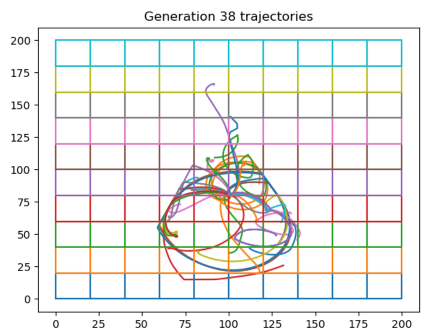
### .3.0.1 Low difficulty map - Final coordinates as behaviour description

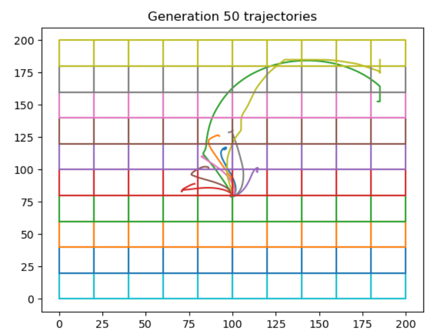
Example of evolution during one code execution.




**(a)** Low difficulty map generation 0 example. **(b)** Low difficulty map generation 18 example.



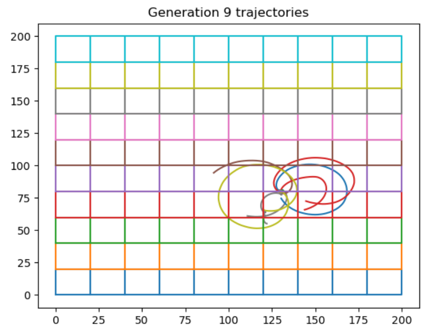




**(c)** Low difficulty map generation 38 example. **(d)** Low difficulty map generation 50 example.

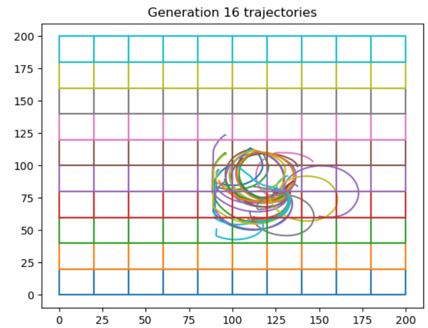
**Figure 12:** Low difficulty map population evolution example - Final coordinates behaviour description.

### .3.0.2 Medium difficulty map - Final coordinates as behaviour description

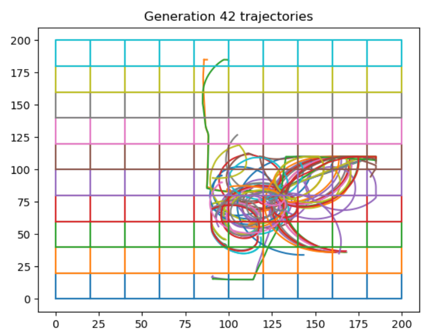
Example of evolution during one code execution for medium difficulty map.



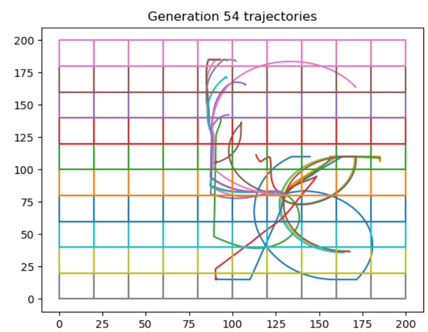

**(a)** Medium difficulty map generation 9 example.




**(b)** Medium difficulty map generation 16 example.




**(c)** Medium difficulty map generation 42 example.

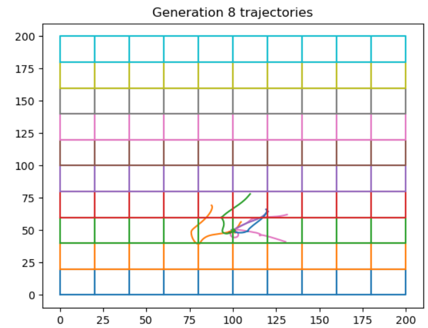



**(d)** Medium difficulty map generation 54 example.

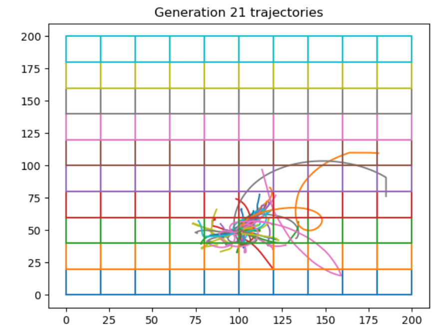
**Figure 13:** Medium difficulty map population evolution example - Final coordinates behaviour description.

#### .3.0.3 Medium-High difficulty map - Final coordinates as behaviour description

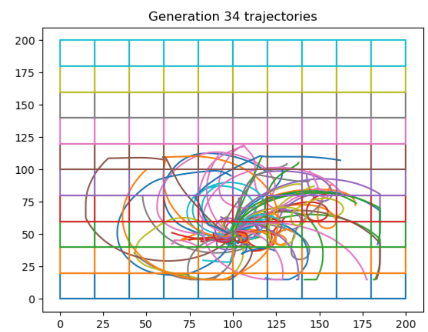
Example of evolution during one code execution for medium/high difficulty map.



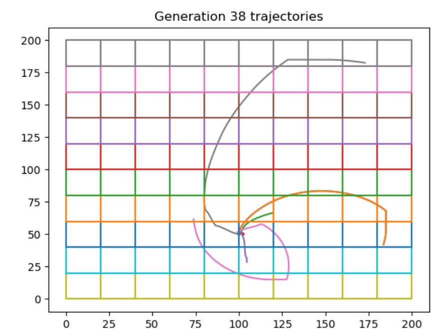

**(a)** Medium-High difficulty map generation 8 example.




**(b)** Medium-High difficulty map generation 21 example.




**(c)** Medium-High difficulty map generation 34 example.

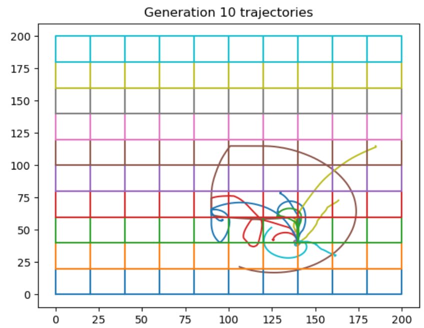


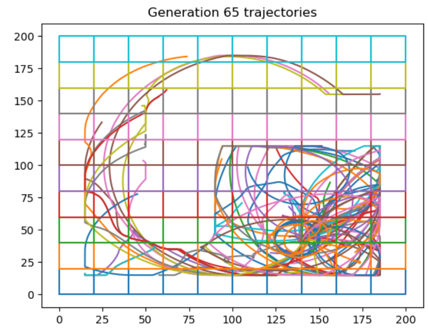

**(d)** Medium-High difficulty map generation 38 example.

**Figure 14:** Medium-High difficulty map population evolution example - Final coordinates behaviour description.

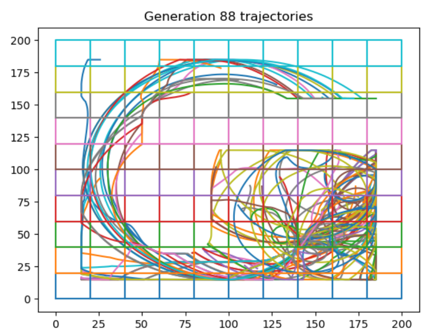
### .3.0.4 High difficulty map - Final coordinates as behaviour description

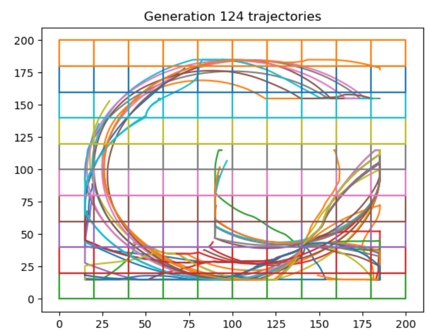
Example of evolution during one code execution for medium/high difficulty map.




**(a)** High difficulty map generation 10 example. **(b)** High difficulty map generation 65 example.




**(c)** High difficulty map generation 88 example. **(d)** High difficulty map generation 124 example.

**Figure 15:** High difficulty map population evolution example - Final coordinates behaviour description.