Utah State University

DigitalCommons@USU

All Graduate Theses and Dissertations, Fall 2023 to Present

Graduate Studies

12-2023

Optimal Stopping of Multi-Robot Exploration for Unknown, Bounded Environments

Trey D. Crowther
Utah State University, a02229257@usu.edu

Follow this and additional works at: https://digitalcommons.usu.edu/etd2023

 Part of the Computer Sciences Commons

Recommended Citation

Crowther, Trey D., "Optimal Stopping of Multi-Robot Exploration for Unknown, Bounded Environments" (2023). *All Graduate Theses and Dissertations, Fall 2023 to Present*. 76.
https://digitalcommons.usu.edu/etd2023/76

This Thesis is brought to you for free and open access by the Graduate Studies at DigitalCommons@USU. It has been accepted for inclusion in All Graduate Theses and Dissertations, Fall 2023 to Present by an authorized administrator of DigitalCommons@USU. For more information, please contact digitalcommons@usu.edu.

OPTIMAL STOPPING OF MULTI-ROBOT EXPLORATION FOR UNKNOWN, BOUNDED ENVIRONMENTS

by

Trey D. Crowther

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Computer Science

Approved:

Mario Harper, Ph.D. Major Professor	Curtis Dyreson, Ph.D. Committee Member
Steve Petruzza, Ph.D. Committee Member	D. Richard Cutler, Ph.D. Vice Provost of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2023

Copyright © Trey D. Crowther 2023

All Rights Reserved

ABSTRACT

Optimal Stopping of Multi-Robot Exploration for Unknown, Bounded Environments

by

Trey D. Crowther, Master of Science

Utah State University, 2023

Major Professor: Mario Harper, Ph.D.
Department: Computer Science

In multi-agent systems, the exploration of unknown environments poses a significant challenge due to inherent uncertainty and limited resources. This research paper investigates the problem of determining the optimal stopping point for multi-agent exploration in such environments. The objective is to devise a strategy that maximizes the discovery of valuable information while considering resource constraints and minimizing exploration time. To evaluate the effectiveness of the approach, extensive simulations are conducted in various scenarios with different environmental characteristics and resource distributions. The findings of this research have significant implications for multi-agent systems deployed in real-world applications such as robotic exploration, search and rescue missions, and autonomous surveillance. The ability to determine the optimal stopping point of multi-agent exploration in unknown environments can lead to more efficient resource utilization, reduced exploration time, and improved decision-making capabilities.

(39 pages)

PUBLIC ABSTRACT

Optimal Stopping of Multi-Robot Exploration for Unknown, Bounded Environments

Trey D. Crowther

Limited resources and uncertainty pose a substantial problem for multi-robot exploration of unknown environments. This research paper looks to determine the optimal time to terminate robot exploration while maximizing information gathered. Whilst making this determination, the system's resources and capabilities must be taken into account. To see if our strategy works, we ran many simulations in varying environments. The results of this research are important for real-world uses like robot exploration, search and rescue missions, and automated surveillance. Determining when to stop exploring can help the system save resources, explore faster, and make better decisions.

To my wife Elaena and son Mason

ACKNOWLEDGMENTS

Thanks to Dr. Mario for his guidance and direction.

Trey D. Crowther

CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

CHAPTER 1

Single Agent Exploration

1.1 Introduction

Exploration and knowledge acquisition play a crucial role in robotics. To start our research, we'll delve into the fundamentals of agent exploration and aim to address these questions: What is required for basic exploration? What limitations do we encounter? What does a Minimum Viable Product entail? We will begin by exploring the fundamental aspects of single-agent search and retrieval. Our discussion starts with the implementation of a single agent assigned the task of identifying and collecting a specific object. This forms a solid foundation for comprehending the capabilities of a fully integrated, multi-agent system.

1.2 Background

The simplest of tasks on the part of an agent may still include many complexities and hurdles to overcome. Our study of the single-agent search and retrieval task will utilize a specific hardware system: the Lynxmotion Hexapod [1]. This implementation will start with bare bone hardware, where the robot has no developed locomotion capabilities and the entire system will have to be designed and implemented from the ground up. The goal of the system is to identify, maneuver towards, and collect a specified object.

1.3 Implementation

1.3.1 Hardware

The Lynxmotion Hexapod is a six-legged, arthropod-inspired robot. It includes a total of 25 servo motors, 18 for leg control and an additional 7 for head and tail movement. Each of the six legs has three degrees of freedom which allow the system to move in any

Fig. 1.1: Lynxmotion Hexapod

direction. The system includes two control boards. The first is a simple actuation controller that communicates via serial UART to each of the servo motors. The second board is similar to an Arduino [2] with its main responsibility being to interpret commands from the user's PS2 controller and pass the corresponding movements to the actuation controller.

As we began our initial inspection of the system we found the first board to be sufficiently capable and that we could proceed with its included abilities. The second board promised to have all locomotion capabilities built in and an easy start up process, but upon further inspection we realized that it was missing several key hardware components and didn't include any documentation relative to its internal API.

This created a significant problem for us to overcome in determining how to proceed. We were presented with two options: we could attempt to understand the internal workings of the second control board and hope to find sufficient information to utilize its pre-built software or we could utilize an entirely new system and implement all of its locomotion from scratch. After attempting the first option for a period without luck, we decided to pursue the second and implement a whole new system.

The primary hurdle in re-engineering all of the robot's motion was to find a system with adequate computational power to simultaneously perform robot locomotion and object detection. We explored several different avenues in the effort to create a cohesive system,

but ultimately decided to utilize a Raspberry Pi 4 as our central control unit. This system promised to provide the necessary brain power for performing both tasks.

With this significantly more capable system backing our hardware we were able to move into the development of the robot's locomotion. As we began this process of implementing its motion, we realized that the included servo motors were significantly under powered. We found that the weight of the robot was often overbearing and led to motor failure in several instances. With this in mind we would need to find a gait design that would provide sufficient stability even with the absence of motor power.

The design that we ultimately utilized was that of the tripod walking gait. This motion is performed by keeping three legs in constant contact with the ground and moving the other three in the desired direction, whether that be forwards, backwards or sideways. This gait provided a stable base and allowed for the under powered stepper motors to adequately control and support the entire system.

1.3.2 Object Detection and Environment Analysis

Another considerable problem to overcome was that of object detection and environment analysis. There were many potential solutions that we could have pursued in the realm of object detection, but in the interest of time we decided to utilize a pre-built Tensorflow based model.

The model that we chose to utilize is a Single Shot Multibox Detector (SSD) model [3] trained on the COCO [4] data set. Common Objects In Context (COCO) is a large collection of images of many every day items such as people, animals and household objects which has been extensively used in the realm of object detection and computer vision. This provided an extremely capable model as our environment analysis tool and allowed for a very robust system even with the semi-limited computational power of the Raspberry Pi. With few exceptions this model was able to quickly identify the desired object in diverse and complex environments.

1.3.3 Results

For the final testing grounds we placed the robot in a complex environment and tasked it with locating and retrieving the desired object, which in this test was a small bottle. We programmed it to rotate in place until the object was located and then move towards the object, keeping it within its vision, until it could be retrieved. The system was sufficiently capable in this desired task and was able to accurately identify the object anywhere within its immediate vicinity. The robot successfully retrieved the object with an approximately 80% success rate.

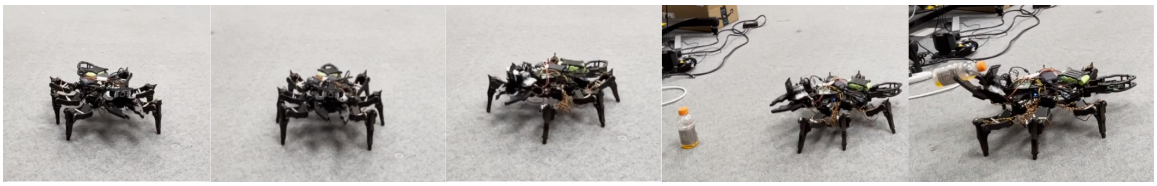

Fig. 1.2: Hexapod completing Search and Retrieval task

1.4 Conclusion

This system proves the viability of creating a simple search and retrieval agent even within the limited capabilities of the Hexapod robot. It demonstrated its capacity to autonomously analyze a room, locate an object, and maneuver itself to retrieve the object. This simple solution allowed us to begin investigating the potential abilities of a more complex multi-agent system. The limitations exposed here have shown the need for an overarching algorithmic structure to coordinate and organize these independent agents. Given the intrinsic limitations, we will begin exploring the potential of optimal stopping and the feasible use cases for similarly limited agents.

CHAPTER 2

Multi-Agent Optimal Stopping

2.1 Introduction

In the field of robotics, multi-robot systems [5] have emerged as a promising avenue for addressing complex exploration tasks in unknown environments. These systems leverage the power of collaboration and coordination among multiple agents to enhance efficiency, coverage, and overall performance. An important challenge in this context is determining when to terminate the exploration process, given limited resources and the desire to maximize the amount of information gathered.

The concept of optimal stopping [6, 7] has significant potential as a means to solve the exploration termination problem. Optimal stopping, a classic problem in decision theory, seeks to find the best moment to halt a sequential decision process in order to maximize an expected reward or minimize a cost. It is particularly relevant in scenarios where the available information is inherently incomplete or uncertain, as is often the case in multi-robot exploration tasks.

2.1.1 Overview

Many multi-agent exploration algorithms rely heavily on a shared knowledge base, where newly acquired information regarding unexplored regions is promptly shared and acted upon. All agents possess awareness of the remaining unexplored area and strategically plan their movements towards these unknown regions. However, this study deliberately eliminates and inhibits the communication capabilities among these agents, aiming to investigate the optimal stopping point for individual agents' exploration.

What is the value in delving into the concept of optimal stopping? We recognize that our systems face tangible limitations. These agents are constrained in their ability to search

and perform tasks within a set time frame. Individually, they cannot cover the entire area efficiently or within their physical constraints. However, by instructing each agent to explore a specific amount ($x\%$) of the total area, we can ensure that collectively, a comprehensive layout is obtained. The goal is to optimize each agent's utility, ensuring that no agent needs to exceed their allotted $x\%$ exploration target.

In numerous scenarios, traditional methods of multi-agent exploration are often impractical due to constraints such as the allotted time frame, the hardware capabilities of the agents, and the accessibility of entry points. One potential approach to address these challenges is area assignment. Nevertheless, these constraints introduce complexity and unreliability when relying solely on individual area assignments.

Furthermore, the inherent complexity of the intended search areas is exacerbated by the possibility of significant layout changes. These changes can be triggered by factors such as fallen debris, renovations, or unforeseen alterations (highly probable occurrences in many of the scenarios we aim to tackle). Consequently, to adapt to these dynamic environments, our study takes an innovative approach: deliberately restricting inter-agent communication, thus shaping it into a distributed algorithm. This strategy compels agents to operate autonomously and gather information as swiftly as possible within the confines of their capabilities and the challenging conditions.

As a result of this experimental approach, substantial overlap among agents occurs, with each individual exploring a significant portion of the total area before the team collectively searches the entire environment. Our experiments include modifications in various variables, allowing us to comprehend their influence on the individual effort required by each agent. These variables encompass the number of agents, the size of the unknown exploration area, the agent starting position, and the relative complexity of each area.

Ultimately, the central question addressed is: How much exploration is required by each individual to achieve collective, complete area coverage?

2.2 Background

In generic multi-robot exploration scenarios, the individual robots are tasked with autonomously navigating through the environment, simultaneously avoiding obstacles and coordinating their actions to efficiently explore the unknown region. However, it is important to acknowledge the inherent communication limitations in such multi-agent environments. Communication overhead becomes a significant factor, as coordinating and sharing information among the robots often incurs additional computational and communication costs. This limitation affects the scalability and efficiency of the exploration process, potentially restricting the types of environments that can be effectively explored. In many real-world scenarios, such as disaster response missions or planetary exploration, the robots may operate in environments with limited or unreliable communication links. Consequently, it becomes crucial to devise strategies that strike a balance between communication requirements and exploration performance.

The main contributions of our work are as follows:

Decentralized Exploration Strategy

Instead of relying on centralized control or extensive communication among agents, our approach allows individual robots to make decisions autonomously based on local information. By employing a decentralized [8] strategy, we reduce the communication overhead while enabling effective exploration.

Scalability and Flexibility

Our proposed approach enhances the scalability of multi-robot exploration by reducing communication bottlenecks. Moreover, the flexibility of the strategy allows it to adapt to various environments, making it applicable in a wide range of real-world scenarios.

The ultimate purpose of any exploration task is to gain information that we can act upon. We want to know when we know "enough". What amount of information about the explored space is sufficient to make an informed decision? We want to know as much about the map as possible, but don't want redundant exploration.

The optimal stopping strategy implemented in these experiments, is well known as the 'Secretary Problem'. [9] This is a reference to the problem many employers face in hiring the best candidate. Their goal is to make the best decision by gaining as much information as possible before deciding, but are often limited by either time or monetary resources. The ultimate question it attempts to answer is "When do we have enough information to act?"

The communication constraint is a critical aspect that sets the stage for the research problem and allows us to explore our optimal stopping question. Experiments have been performed in this realm of limited communication [10] and there are even some that compare and contrast decision algorithms [11]. We, on the other hand, will ignore these decision making algorithms and focus on the vanilla version of this problem: to find the best stopping point for each agent to terminate its exploration.

In conclusion, our research aims to understand the impact of communication-constraints on multi-robot exploration and efficient mapping of unknown bounded environments. By leveraging optimal stopping theory and a decentralized approach, we provide important insights into the realities of these complex systems. Our findings contribute to advancing the field of robotics and have potential applications in various domains, including disaster response, environmental monitoring, and planetary exploration.

2.3 Methods

2.3.1 Environment

The simulation utilizes the Python libraries Pygame and Matplotlib. We assume the area is bounded and the robots have no prior knowledge about the search area. The simulation is capable of varying several hyperparameters, including the method used for exploration, the starting locations, the map length, the number of agents, the experiment iteration, and the area complexity.

The simulation can be run for multiple iterations and multiple experiments can be run in parallel. Many iterations can be used to evaluate the robustness of the different methods, and the impact of varying hyperparameters. In this simulation, the decision was made to

hold a few of the parameters constant to help focus on the most important variables and their impact on the optimal stopping point. The modified variables are as follows:

2.3.2 Starting Locations

The first variable that we manipulated in our experiments was the starting locations of the agents. The variations that we chose to test were edge, random and top left starting positions. See Figure 2.1.

The edge starting location placed the agents evenly around the border of the map in a divide and conquer type scenario. This simulates a tactical group entering an area at all of the entrances and individually exploring the area closest to them.

The random starting location placed each of the agents randomly within the bounds of the map. This simulates more of a planetary exploration where the agents are initially placed somewhere in the center of the area and must expand outwards.

The top left variation started all agents in the same place at the top left section of the map. This simulates a more constrained situation where there are limited entry points, as is the case in many search and rescue [12] situations.

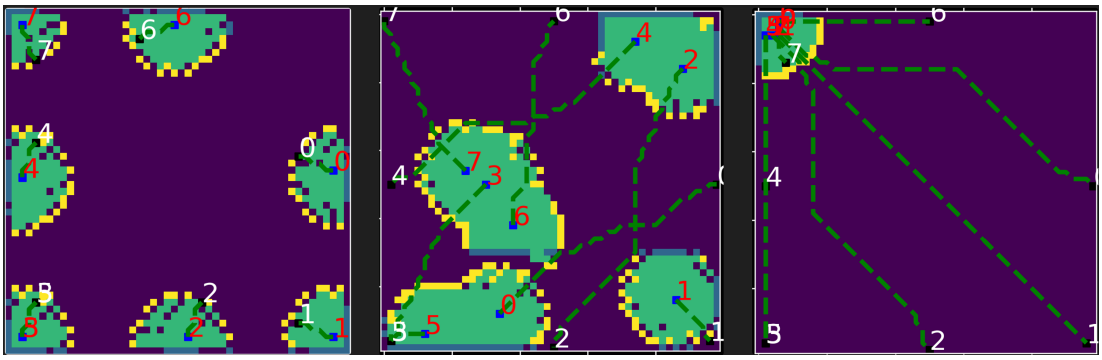

Fig. 2.1: Edge, Random and Top Left Start Locations

For all of the start positions, the agents were given an initial goal location randomly located along the edge of the map. This allowed the agents to start off in non-identical directions and helped jump start initial exploration.

2.3.3 Map Details

The size of the map can also be varied to evaluate the impact, if any, of map size on exploration efficiency. The three map sizes that were used in our experiments were square areas of 25x25, 50x50 and 100x100 units.

The map complexity parameter was modified heavily to see what impact complexity would have on the convergence times of the agents. This value is quantified by the percentage of the unknown area obstructed by untraversable objects such as walls. These values varied from 0% area complexity (or an empty room) to nearly 25% area obstruction. We ran experiments with a total of 7 different complexity levels. See figures 2.2 and 2.3.

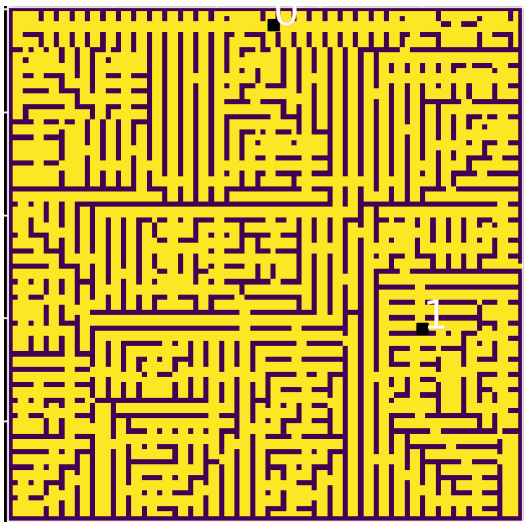

Fig. 2.2: Large Map with high complexity

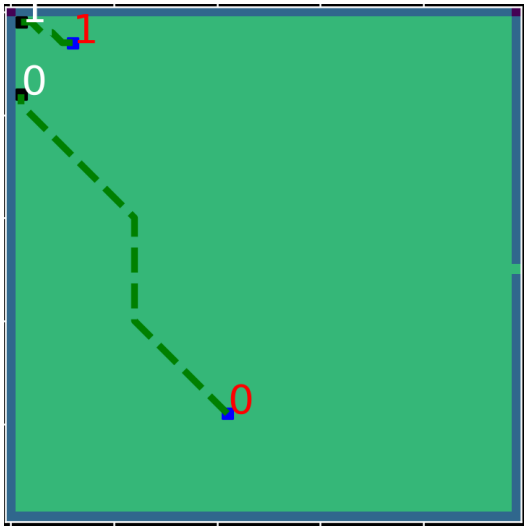

Fig. 2.3: Map with zero complexity

2.3.4 Frontier Horizon

In this research, a careful consideration was given to the selection of an exploration technique that strikes a balance between effectiveness and simplicity. While various exploration methods with their distinct advantages and limitations were considered, the Frontier-Based [13, 14] search method was chosen for this study.

The Frontier-Based search method involves identifying the frontiers, the boundaries between the known and unknown regions, of the environment and prioritizing the explo-

ration of the closest frontier area. This approach was deemed suitable for the experiment due to its demonstrated effectiveness in optimizing the exploration performance of a team, as well as its simplicity of implementation. By utilizing this method, the experiment can avoid the complexities associated with other sophisticated search techniques.

Therefore, the selection of the frontier-based search method serves as a pragmatic approach that balances effectiveness, simplicity, and the experimental objectives. This enables a focused investigation into the optimal exploration stopping point in the context of the proposed research question.

2.4 Results

These results include a compilation of data from more than 30,000 total experiments.

2.4.1 Convergence Time

The first discussion of the final results is related to the convergence time with a varying number of agents in the map. The most interesting find is related to the exploration speed of 2 agents. Even though there are two agents exploring, each must individually explore more than 90% to achieve full map coverage.

With a single agent performing the exploration we expect a linear progression, meaning that as the agent learns more about the area, the more the group knows about the area. From this result we can conclude that adding a single additional agent into the exploration task where there is no communication is almost futile and results in very little benefit. This outcome was invariant among all the results and was independent of all starting locations, area complexities, and map sizes.

At the other end of the spectrum, when there are 16 agents performing the exploration task, they need only individually explore approximately 30% of the total area before reaching complete map coverage. The required exploration ratio with an increased number of agents was expected to decrease, but the significance of the change is only evident as we grow the number of agents in powers of 2.

As we progress through an increasing number of agents we can see an increased benefit. The combined results can be found in Figure 2.4.

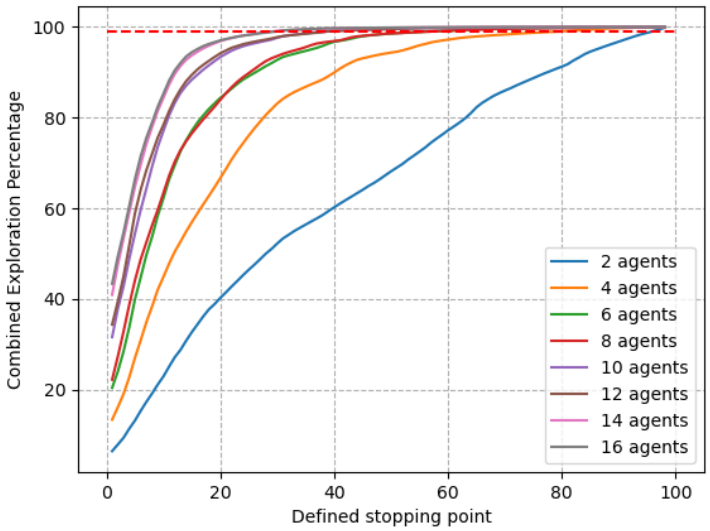

Fig. 2.4: Combined results, from an edge start position

2.4.2 Starting Location

The variation that is seen when utilizing different starting positions introduces an entirely different outlook on the problem. For this discussion we will consider convergence when the group has collectively explored 99% of the map. The random start position produced the most efficient results and the agents converged with lowest required exploration, which got to as low as 27% individual exploration per agent. Edge start and random start were comparable, but random initial positioning had a slight advantage in all agent counts and map complexities.

As mentioned earlier, the average convergence exploration ratio for the edge start was almost 100% with just 2 agents, but with 16 agents the required ratio was reduced to as low as 30%. On the other hand, the top left start position produced the worst results. While 2 agents yielded similar results for all three starting locations, at the maximum agent count

the required individual exploration reached its minima at around 40%.

Table 2.1: Convergence Point in 50x50 Map

Agent Count	2	4	6	8	10	12	14	16
Edge	95	81	59	57	41	39	31	30
Random	95	80	57	53	35	31	28	27
Top Left	94	81	60	58	45	43	41	41

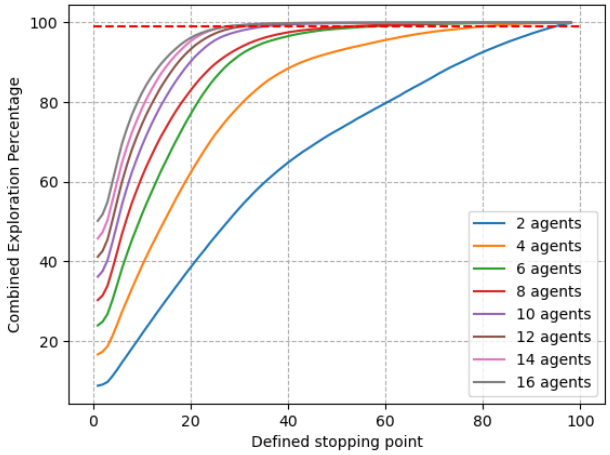

Fig. 2.5: Combined results, from a random start position

At the beginning of the plots for the random start there is a small plateau. This is likely a result of the agents each exploring areas that were already seen by the other agents. The agents expand outwards and are learning more individually, but collectively don't gain any new information.

The random start also saw steady advances in convergence point as more agents were added. With random start positions the required exploration ratio consistently dropped by 4 or 5 percent with each addition of 2 agents. The other starting positions saw less significant jumps as agents were added.

The random start variation received a head start from each of the agents being placed

somewhere in the center of the map. With 16 agents and without moving, the agents already explored nearly 50% of the total area. This result compared to the top left start position where there was no, or very little, information gained at the start of the experiment. This can be seen in Figure 2.6 where, with all agent counts, the agents saw very little of the map at first and needed to progressively expand together.

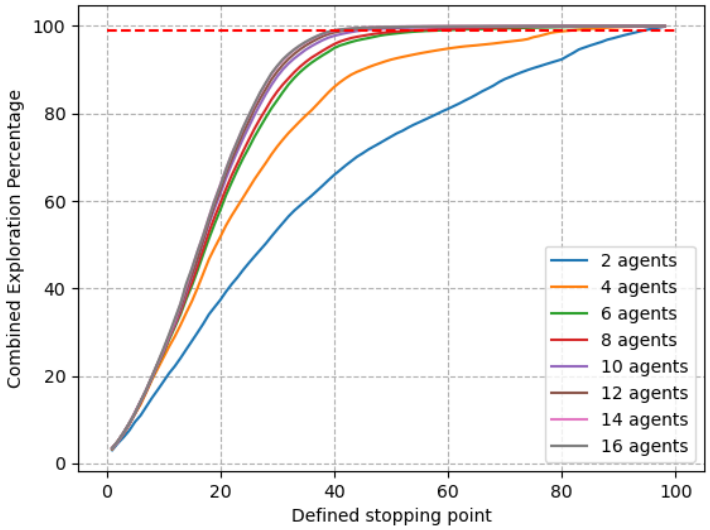

Fig. 2.6: Combined results, from a top left start position

Some of the most interesting results come from the direct comparison of the starting locations in one graph.

Figures 2.7 and 2.8 outline these results

In the experiments with 2 agents, the starting position had very little bearing on the required exploration ratio. All three of the start locations yielded almost identical results and even had the top left start position finish first. As more agents are included, the difference between the starting positions becomes more drastic.

One thing of note in all of the higher agent counts was that the edge start was on pace to converge the fastest but ultimately random start positioning completed area coverage first. As can be seen in the 16 agent graph, the edge start reaches 95% convergence faster

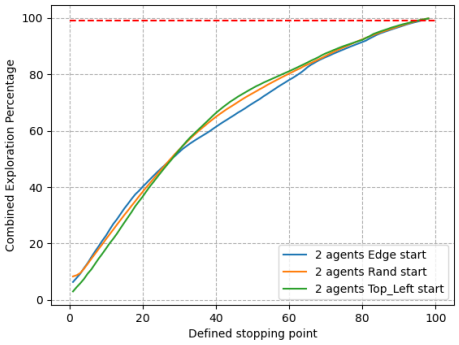

Fig. 2.7: Visual comparison of experiment progression with 2 agents

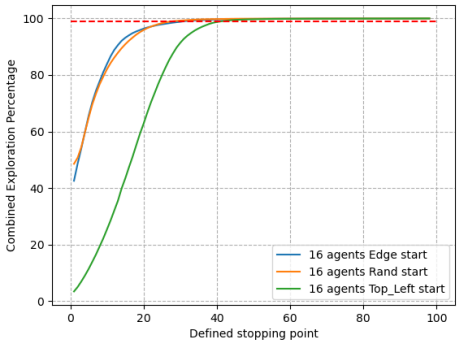

Fig. 2.8: Visual comparison of experiment progression with 16 agents

but was unable to complete the last portion as quickly.

2.4.3 Map Sizes

Here we want to discuss the impact of the different map sizes on the convergence point. As we gathered the data for the convergence point we expected to see similar results among all map sizes, but the convergence data for the 25 by 25 map size yielded slightly different results than the others. They are outlined in table 2.2.

As was mentioned before, we deemed convergence to be 99% combined area coverage. For map sizes of 100x100 and 50x50 this results in a convergence point where the agents can miss up to 100 and 25 area squares respectively. When there are a few missing squares along the edges that has very little impact, but when the margin of error for a 25x25 map is only 6 squares, it is much more difficult to achieve that convergence level. The group can quickly explore a vast majority of the map, but the last few areas require significantly more exploration.

To get a more realistic result we will use a slightly lower convergence level and then compare the map sizes. The results for when we only expect 95% area coverage for convergence are outlined in table 2.3.

Table 2.3 displays the expected results where the map size has little bearing on the required exploration ratio of individual agents. Almost all values of the 25x25 map fall

Table 2.2: 99% Coverage Convergence

Agent Count	2	4	6	8	10	12	14	16
25x25 Edge	96	92	85	73	66	65	54	53
50x50 Edge	95	81	59	57	41	39	31	30
100x100 Edge	96	82	62	59	41	35	32	30
25x25 Random	96	92	83	71	62	60	37	37
50x50 Random	95	80	57	53	35	31	28	27
100x100 Random	95	82	65	56	39	35	30	29
25x25 Top Left	96	93	87	76	66	67	57	56
50x50 Top Left	94	81	60	58	45	43	41	41
100x100 Top Left	94	82	66	64	49	46	43	42

Table 2.3: 95% Coverage Convergence

Agent Count	2	4	6	8	10	12	14	16
25x25 Edge	86	79	64	51	44	37	26	23
50x50 Edge	85	53	36	33	23	22	17	17
100x100 Edge	86	58	36	35	24	20	18	17
25x25 Random	85	70	46	36	25	22	18	18
50x50 Random	85	58	35	32	24	22	19	18
100x100 Random	84	60	38	32	25	23	20	19
25x25 Top Left	87	73	59	38	35	34	31	30
50x50 Top Left	84	60	40	39	35	34	33	33
100x100 Top Left	83	60	43	42	35	34	33	33

within close proximity to those of other map sizes.

2.4.4 Complexity

Thus far, the results have been combined for all complexity levels to get an overall picture of how the starting locations and map sizes impacted the required exploration ratios. At this point we will separate the data into the individual map complexities to understand how complexity impacted the results.

It seems intuitive that the more complex maps will require more time to converge, due to the lower direct visibility of individual agents, but these experiments will help us to

understand whether that higher complexity will also require more individual exploration to achieve that convergence. We will primarily look at the results from a large map due to the larger impact that complexity can have with more area to explore.

The first results were those of varying map complexities with a random start position and large map of 100x100. These results yielded very little variance among the diverse complexity levels. All of the levels converged at a very similar rate, especially at higher agent counts. Here are the results from 16 agents:

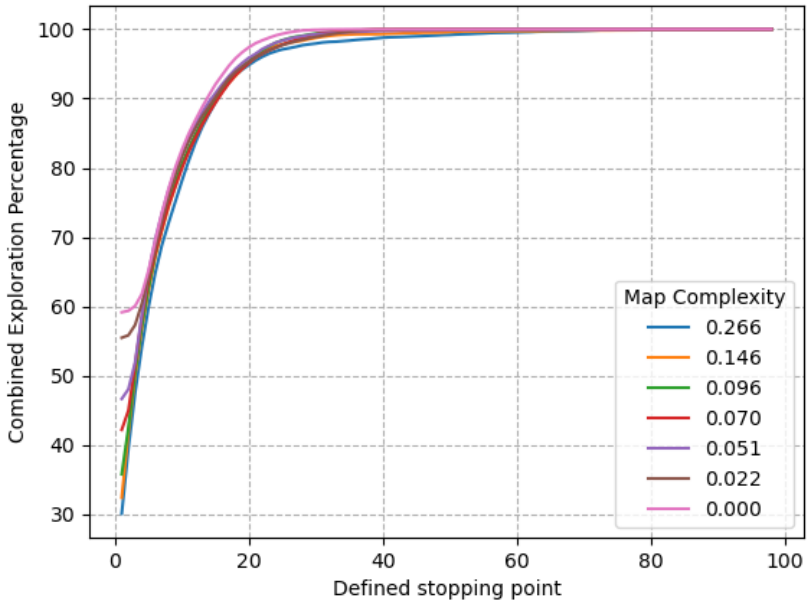

Fig. 2.9: Complexity comparison with 16 agents, random start position and 100x100 map size

One significant note from this graph is that we see greater initial exploration values from lower complexity levels. We also see this result as we increase the number of agents. When we do both simultaneously we see a combined initial exploration percentage that nears the 60% range.

Looking at a different variation of parameters we see higher variance among the complexity levels. The variation with the greatest difference among complexities was the top

left start position. It yielded the following results with 2 and 16 agents in Figures 2.10 and 2.11.

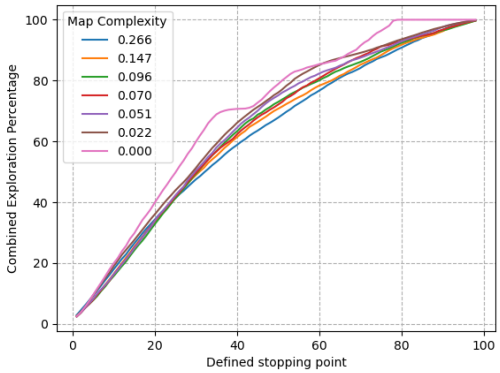

Fig. 2.10: Complexity comparison with 2 agents, top left start position and 100x100 map size

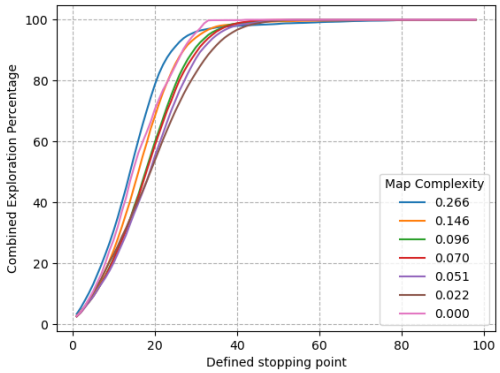

Fig. 2.11: Complexity comparison with 16 agents, top left start position and 100x100 map size

The plateaus in the first graph show that the 2 agents spend a significant amount of time learning more for themselves, but gaining no additional knowledge for the group, as was seen before. These plateaus are also evident in the higher agent counts but are not visible due to the averaging that occurs among the many agents.

A differing result that can be seen in the second graph is that the most complex map converged at a higher rate than any of the other map complexities, which is the opposite result from that found in the random start position.

A final note about these results is that they were averaged across 50 different experiments with maps created from all different random seeds. Thus, these results were derived from a large amount of randomly created samples.

2.4.5 Time Constrained Exploration

One of the many reasons for this research is to understand how quickly a group of agents can learn about a specific environment, but what about situations where time is the real constraint? Thus far we have only discussed the amount of exploration in terms of

ratios and percentages, now we will address that exploration in terms of exploration time. The question is this: how much can we expect to learn assuming we have a specific time constraint? For these results we will utilize the best performing start location and assume a brisk agent velocity of 1 meter/second. These are the results:

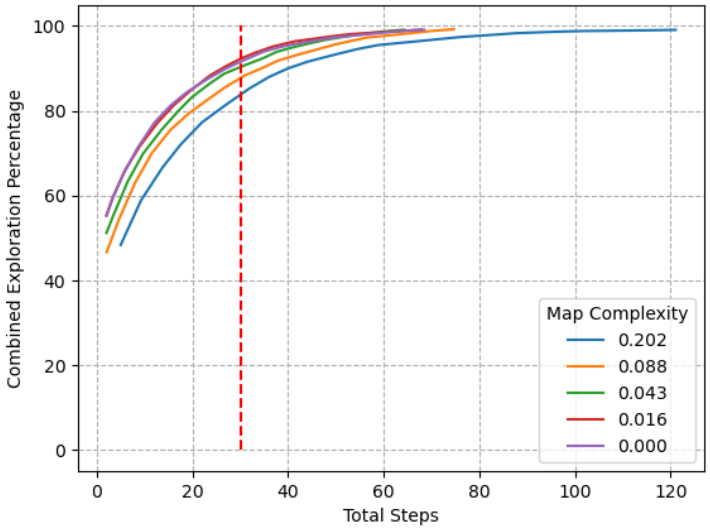

Fig. 2.12: Combined exploration with 30 second time constraint, 16 agents, and random start position

With a time conversion of one time step per second, in the most ideal of circumstances (low area complexity, high agent counts and random starting position) the agents can explore around 90% of the map in a 30 second period. In less ideal scenarios they can explore around 85% of an unknown area.

To achieve the convergence that we have discussed up to this point, that of 99%, it would take over 60 time steps or around 1 minute for most map complexities. For highly complex maps we can expect a convergence time of nearly double that of 2 minutes.

Now let's compare that to a situation where there is perfect communication among all agents in Figure 2.13.

With full communication the agents can explore around 95% of the map in that same

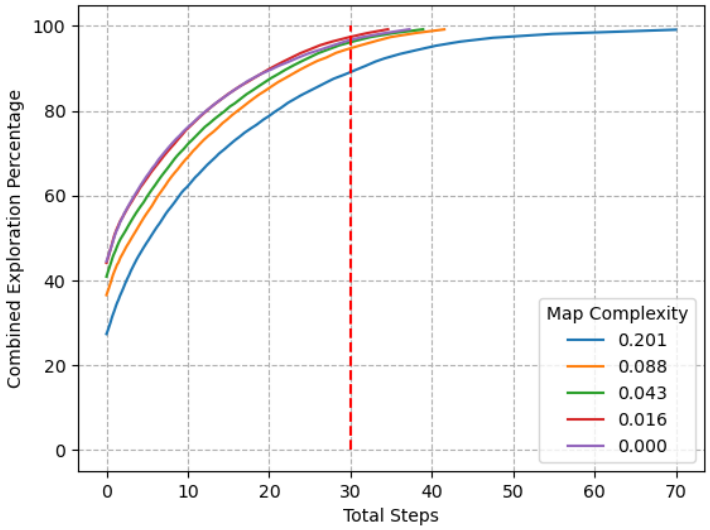

Fig. 2.13: Combined exploration with 30 second time constraint, 16 agents, random start position, and full communication

30 seconds. That is only a 5% increase of area coverage when we allow communication compared to the simulations with no communication. This bodes well for our optimal stopping implementations and shows that their convergence times are very comparable to a fully communicating system.

2.4.6 Cost of Exploration

Now that we have a grasp on the theoretical possibilities of these systems, let's discuss the real cost of implementation and knowledge gathering. Our knowledge gathering is highly dependent on the capabilities of the sensor package available with these robots. Smaller agents are slower, less capable and have lower sensor range, but come with a smaller price tag and therefore greater expendability. For the types of scenarios that we have discussed, it would require agents that are capable of exploring at a high rate of speed and traversing difficult terrain. For example, a robust system filled with several of Boston Dynamic's extremely capable and robust *Spot* [15] robots would cost nearly $75,000 dollars a piece. These systems are very expensive, but are equally capable and may be necessary in disaster areas or areas with primarily uneven terrain.

On the other hand, much of the same capability and ruggedness could be accomplished by utilizing an option much like the *Diablo Direct Drive* [16] which provides similar levels of versatility with a much smaller price tag of $3,500. This type of wheeled system may have difficulty in instances where *Spot* may be comfortable, but overall it provides a similarly capable option. Ultimately, the agent of choice is highly dependent on the types of scenarios where it would be utilized, the desired capability of the user, and the available budget.

2.5 Conclusion and Future Work

This study has explored the impact of different parameters on the multi-robot exploration task in terms of optimal stopping. We have evaluated the performance of few to many agents, small and large maps, and distinctive starting position strategies. Ultimately our primary research question was: How much exploration is required by each individual to achieve complete area coverage?

In the most ideal of circumstances, with many agents, good starting position and a simple map, the experiments yielded a ratio of approximately .30.

Other research can be performed relative to these topics to more fully understand their place among these findings. One such example would be to perform these experiments with live hardware instead of performing only simulation. This could give us further insights into the true optimal stopping aspects of this problem and the need to maximize benefit whilst minimizing cost.

The findings can inform the design of more effective and efficient exploration methods for multi-robot systems, with potential applications in fields such as search and rescue, surveillance, and environmental monitoring. This study contributes to the development of intelligent and autonomous multi-robot systems that can navigate and explore unknown environments.

CHAPTER 3

Multi-Agent Reinforcement Learning

3.1 Introduction

We have explored the results of a deterministic simulation that included explicit rules for the agents' exploration. In this section we will explore a reinforcement learning based solution to this multi-agent exploration problem. We are looking to understand what types of learned actions the agents will adopt and how they will attempt to explore the unknown area. Will the agents adopt new and original strategies? How will they coordinate their actions to most effectively explore?

3.2 Environment

In our implementation we will be utilizing the OpenAI Gymnasium [17] toolkit. The Gymnasium ecosystem is a tool utilized to train and compare reinforcement learning models in simple to complex environments. The custom environment that we have created is a grid-defined, bounded world similar to that of the previous simulation. This environment will help us better understand the capabilities of reinforcement learning without making an extremely complex real world model. The algorithm that we will utilize to train the model is the Proximal Policy Optimization (PPO) [18], which has been shown to be very capable in complex, high-dimensional state and action spaces.

The agents will have a similar field of view compared to that of our previous simulation and will be able to see 4 grid spaces away from their current position in all directions. This will give them an adequate range and is representative of the capabilities of many real world systems. The agents are also capable of moving one space in any of the cardinal directions every time step.

3.3 8 Agent Model

For this first implementation there will be eight agents and a square map of 50x50 units. The observation space of the model will be four values for each agent. The model will receive the x, y coordinate of the agent and the x, y coordinate of the closest unexplored square. This will result in a total observation space and learning model input of 32 values.

In an ideal scenario we would be able to output the entire world space and inform the model of all relevant information. This could potentially include the agents' positions, wall locations and all unexplored areas. If we were to pass all of this information into the model, this would increase the size of the input to around 2500 values, one for each x, y coordinate.

In an effort to limit the complexity of this model and minimize training time we decided to restrict the number of output values of the environment and only provide the absolutely necessary information to the model. Although this can potentially speed up training and produce a better model, it also inhibits our ability to incorporate diverse features into the environment, such as obstacles. Without the additional information provided by a complete world observation space, the model would be unaware of the full set of map features and would be unable to make informed decisions.

The reward structure for the first model will be as follows:

Rewards

- +1 for each agent that moves towards the closest unexplored region
- +100 for achieving complete area exploration

Penalties

- -1 for each agent that attempts to move into a wall or the edge of the bounded area
- -1 for each agent that moves back and forth between two adjacent spaces

This reward structure should incentivize the agents to explore quickly and work together.

3.4 8 Agent Results

Here are the combined results from 50 simulation runs of the trained exploration model after training for 25,000,000 time steps:

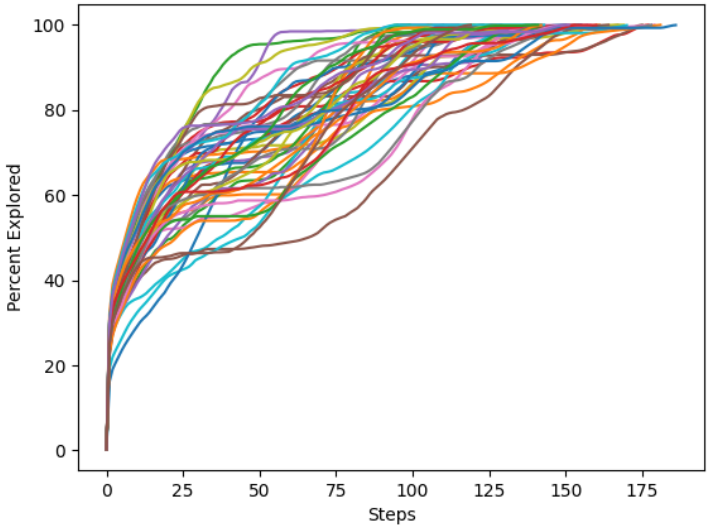

Fig. 3.1: Reinforcement Learning Results

This model produced less performant results compared to that of the deterministic simulation. It took an average of approximately 130 time steps for the agents to complete the combined exploration of the bounded area. This is approximately 4 times longer than that of the previous simulation. The agents did not learn to strategically or collaboratively explore the map, thus resulting in very slow exploration times. Even the most efficient exploration done by the model took 96 steps to complete.

Despite the simplifications made and significant training time, the model was unable to converge to a more efficient or unique stratagem. We believe that given more time and training, the model could produce better results than that of the deterministic solution.

3.5 2 Agent Model

Following the unsatisfactory results obtained from training the initial model, we opted

to develop a more straightforward alternative. For this one we will create a smaller grid of 15x15 and only include two agents. Due to the smaller grid we will be able to incorporate the entire map into the observation without including an unreasonable number of inputs. We hope that this will provide a more optimized result where the model can be more aware of the entire state of the grid.

The reward structure for the second model will also be simpler to determine if that will yield better results. The structure will be as follows:

Rewards

- +1 for each new square that the agents explore
- +100 for achieving complete area exploration

3.6 2 Agent Results

Here are the combined results from 50 simulation runs of the trained exploration model after training for 10,000,000 time steps. We have also included the results from 50 runs of the deterministic simulation with a 15x15 grid and 2 agents which achieved an average of 72 time steps.

The second model yielded significantly improved results, achieving an average of 73 time steps for completion during exploration, consistently delivering reliable outcomes. The biggest improvement that we saw from the second model was the decreased variation in convergence time. Almost all of the runs were able to complete in a small range of time steps where this was not the first case with the first model.

The second model appeared to devise a more effective strategy for exploring the area. Unlike the 8-agent model, individual agents autonomously explored nearby areas. In instances where unexplored regions were situated in opposing corners of the grid, the agents strategically divided and explored independently, which aligned with the anticipated strategy.

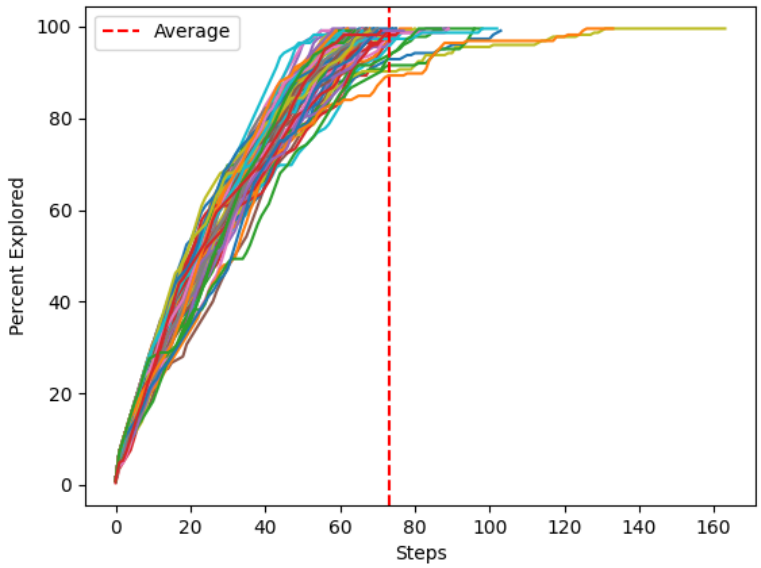

Fig. 3.2: Reinforcement Learning Results with 2 Agents and Full Observation

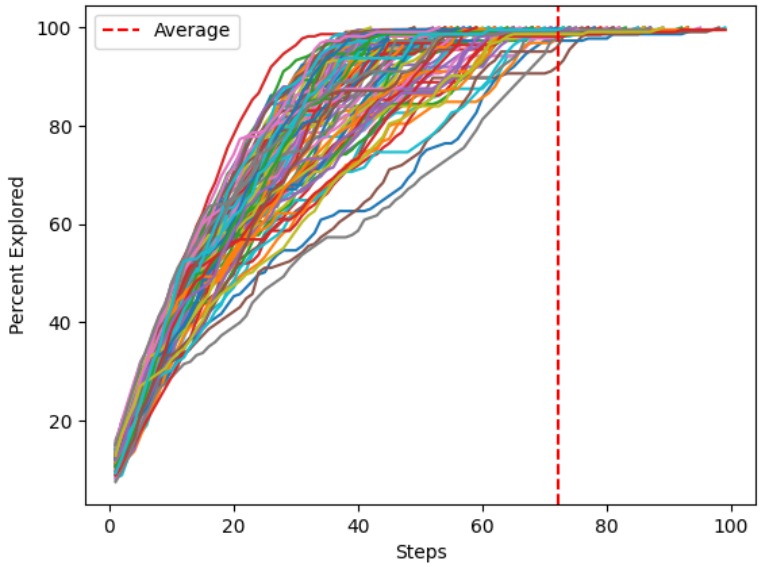

Fig. 3.3: Deterministic Simulation with 2 Agents

CHAPTER 4

Conclusion

The initial research performed was a hardware implementation of a simple exploration task. Due to the limitations of single agent exploration we now more fully understand the need for a comprehensive algorithmic system to coordinate and organize their exploration. This exercise led us to ponder about the implications of utilizing hardware systems with limited physical capabilities and what remedies could be proposed to overcome these limitations.

With these obstructions in mind we decided to investigate concepts within the realm of optimal stopping to see what advancements could be made to help more efficiently utilize resources. This allowed us to see the impact it would have in terms of the amount of required exploration for individual agents. We determined that we can achieve comparable results to systems that utilize a fully integrated communication model.

The integration of a trained reinforcement learning model revealed the intricacies inherent in the proposed environments, underscoring the inherent challenges of training a high-performance model. Despite these challenges, the model demonstrated results closely comparable to those achieved by other implementations.

The three topics discussed have allowed us to explore several facets of multi-agent exploration. The primary contribution of this paper is an analysis of optimal stopping ideology in the domain of multi-agent exploration. Overall, this study contributes to the development of intelligent and autonomous multi-robot systems that can navigate and explore unknown environments.

REFERENCES

[1] "A-pod 3dof hexapod," https://wiki.lynxmotion.com/info/wiki/lynxmotion/view/servo-erector-set-robots-kits/ses-v1-robots/ses-v1-3-4-dof-hexapods/a-pod/, accessed: 2023-10-19.

[2] "Arduino," https://www.arduino.cc/, accessed: 2023-10-.

[3] "Tensorflow object detection model," https://github.com/tensorflow/models/tree/master/research/object_detection, accessed: 2023-10-19.

[4] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.

[5] Z. Yan, N. Jouandeau, and A. A. Cherif, "A survey and analysis of multi-robot coordination," *International Journal of Advanced Robotic Systems*, vol. 10, no. 12, p. 399, 2013.

[6] I. SANGUANMOO, "Optimal stopping time and its applications to economic models," 2020.

[7] S. Becker, P. Cheridito, and A. Jentzen, "Deep optimal stopping," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 2712–2736, 2019.

[8] I. F. P. B. Aaron Hao Tan, Student Member and I. Goldie Nejat, Member, "Deep reinforcement learning for decentralized multi-robot exploration with macro actions." IEEE.

[9] B. Christian and T. Griffiths, *Algorithms to Live By: The Computer Science of Human Decisions*. Holt, Henry & Company, Inc., 2017.

[10] R. W. Ninad Jadhav, Meghna Behari and S. Gil, "Multi-robot exploration without explicit information exchange."

[11] M. Otte, M. J. Kuhlman, and D. Sofge, "Auctions for multi-robot task allocation in communication limited environments," *Autonomous Robots*, vol. 44, no. 3, pp. 547–584, 2020.

[12] F. Niroui, K. Zhang, Z. Kashino, and G. Nejat, "Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 610–617, 2019.

[13] B. Yamauchi, "Frontier-based exploration using multiple robots," in *Proceedings of the second international conference on Autonomous agents*, 1998, pp. 47–53.

[14] Y. Wang, A. Liang, and H. Guan, "Frontier-based multi-robot map exploration using particle swarm optimization," in *2011 IEEE symposium on Swarm intelligence*. IEEE, 2011, pp. 1–6.

[15] "Spot, boston dynamics," https://bostondynamics.com/products/spot/, accessed: 2023-10-19.

[16] "Diablo, direct drive," https://shop.directdrive.com/products/diablo-world-s-first-direct-drive-self-balancing-wheeled-leg-robot?variant=41103436513331, accessed: 2023-10-19.

[17] "Gymnasium environments," https://gymnasium.farama.org/, accessed: 2023-10-19.

[18] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *ArXiv*, vol. abs/1707.06347, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:28695052