B. TECH. PROJECT REPORT

On

Heterogeneous Multi-Agent Search using Reinforcement Learning

by

Nachiket Mokashi

DISCIPLINE OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

May 2022

Heterogeneous Multi-Agent Search using Reinforcement Learning

PROJECT REPORT

Submitted in partial fulfillment of the requirements for the award of the degrees

of

BACHELOR OF TECHNOLOGY

in

MECHANICAL ENGINEERING

Submitted by:

Nachiket Mokashi

Discipline of Mechanical Engineering

Guided by:

Dr. Chandresh Kumar Maurya (IIT Indore)

Dr. Guillaume Adrien Sartoretti (National University of Singapore)

INDIAN INSTITUTE OF TECHNOLOGY INDORE

May 2022

CANDIDATE'S DECLARATION

I hereby declare that the project entitled **"Heterogeneous Multi-Agent Search using Reinforcement Learning"** submitted in partial fulfillment for the award of the degree of Bachelor of Technology in 'Mechanical Engineering' completed under the supervision of **Dr. Chandresh Kumar Maurya,** Assistant Professor, Discipline of Computer Science & Engineering, IIT Indore and **Dr. Guillaume Adrien Sartoretti**, Assistant Professor, Discipline of Mechanical Engineering, National University of Singapore is an authentic work.

Further, I declare that I have not submitted this work for the award of any other degree elsewhere.

Nachiket Mokashi

180003034

CERTIFICATE by BTP Guide(s)

It is certified that the above statement made by the student is correct to the best of our knowledge.

Dr. Chandresh Kumar Maurya
Assistant Professor
Department of Computer Science & Engineering
IIT Indore

Dr. Guillaume Adrien Sartoretti
Assistant Professor
Department of Mechanical Engineering
NUS

PREFACE

This report on "**Heterogeneous Multi-Agent Search using Reinforcement Learning"** is prepared under the guidance of Dr. Chandresh Kumar Maurya and Dr. Guillaume Adrien Sartoretti.

Through this report, I have attempted to explain the process and design of an algorithm for searching for static targets in an unknown environment, using a team of heterogeneous robots having different motion and sensing capabilities. This report also describes the benefits of employing reinforcement learning to distribute the agents efficiently and minimize searching time, compared to conventional approaches toward multi-agent searching.

Through this thesis, efforts have been made to present the methodology, results and conclusions of the study in a lucid and comprehensible manner. Figures, graphs and tables have been included to make the content more illustrative.

Nachiket Mokashi
B.Tech. IV Year
Discipline of Mechanical Engineering
IIT Indore

Acknowledgements

I wish to thank Dr. Chandresh Kumar Maurya and Dr. Guillaume Sartoretti for their kind support and valuable guidance.

I would also like to thank Mr. Weiheng Dai for his contribution to the project and Mr. Yizhuo Wang for his timely help. They enabled all the ideas to take shape into this project and supported me in the completion of the objectives.

I would also like to thank the Discipline of Mechanical Engineering, IIT Indore, for their kind co-operation at each step.

Without their support this report would not have been possible.

Nachiket Mokashi
B.Tech. IV Year
Discipline of Mechanical Engineering
IIT Indore

Abstract

The dynamic and unpredictable nature of our world makes it difficult to design one autonomous robot that can efficiently adapt to all circumstances. Therefore, it makes sense to implement heterogeneous multi-robot systems to be able to solve complex tasks. The aim of this project is to search for targets in an unknown environment, using a team of heterogeneous agents/robots having different motion and sensing capabilities, employing reinforcement learning to distribute the agents efficiently and minimize searching time. The intuition behind heterogeneous search is considering different sensor capabilities, we want to find an online area decomposition to guide agents to search efficiently, finding how and where to go without many optimizations. A literature review of relevant work reveals that a majority of the current methods for multi-agent searching are either about homogeneous agents or using the same policy, under the same action space. There are very few papers describing heterogeneous multi-agent searching, and even those that do focus more on improving communication or other aspects. However, it is clear that heterogeneous multi-agent searching is an important emerging field and with the help of reinforcement learning, has the potential to lead to state-of-the-art performance on complicated tasks. For this project, we start with a model of ergodic search using homogeneous agents, then try to represent the ground truth, assuming perfect sensors and perfect data fusion, by applying concepts similar to CNNs. We then add heterogeneous sensors and decompose the map optimally (i.e., find which areas are best searched by which agent), and then gradually add uncertainty and reward-based trajectory optimization (i.e. reinforcement learning) while balancing exploration and exploitation. The applications of heterogeneous multi-agent searching range from agriculture to search & rescue. Future work in the field includes trying to use distributions other than the Gaussian distribution to represent more complicated sensors, and optimizing paths with fewer iterations.

Keywords: Multi-robot searching, Heterogeneous, Map decomposition, Reinforcement learning, Exploration & exploitation

Nomenclature & Abbreviations

Nomenclature/ Abbreviation	Description
ML	Machine Learning
RL	Reinforcement Learning
MAS / MRS	Multi-agent Search / Multi-robot Search
MADRL	Multi-agent deep reinforcement learning
MDP	Markov decision process
GP	Gaussian process
FFT	Fast Fourier Transform
TD	Temporal difference (learning)
KL divergence	Kullback-Liebler divergence
DP	Dynamic programming
CNN	Convolutional neural network
UAV / UGV	Unmanned Aerial Vehicle or Unmanned Ground Vehicle

Table of Contents

Candidate's Declaration v
Supervisor's Certificate v
Preface vii
Acknowledgement ix
Abstract xi
Nomenclature xiii
Table of Contents xiv
List of Figures xvi

Introduction 1
1.1 Background 1
1.2 Problem definition 3
1.3 Related work 4
1.4 Structure of the document 5

Literature Review 8
2.1 Multi-agent searching (MAS) – Conventional approaches 8
2.2 Heterogeneous MAS 10
2.3 Reinforcement learning (RL) based Heterogeneous MAS 11

Heterogeneous sensor capabilities 16
3.1 Sensor characteristic definition 17
3.2 Sensor Data fusion using Gaussian processes 18
3.3 Probability and Uncertainty maps 21

Heterogeneous Multi-Agent search using CNN approach 24
4.1 Fourier transform 24
4.2 Voronoi partitioning 24
4.3 Spatial Map decomposition (based on Gaussian processes) 25
4.4 Technical details of the model 28
4.5 Scalability testing and scope for improvement 29

Heterogeneous Multi-agent search with reinforcement learning.....31
5.1 Implementing reward structure 32
5.2 Temporal Difference Learning.....32
5.3 Interactive Tool to see Map Resolution in Real Time 33
Results & Discussion.....35
6.1. Map Decomposition.....35
6.2. Heterogeneous RL-based search & decomposition results.....36
6.3. Heterogeneous agents task allocation 37
Conclusion & Scope for Future Work 40
References.....42

List of Figures

Figure number	Title	Page number
Fig 1.1	Branches of the multi-agent search research line at MARMot Lab, NUS	5
Fig 2.1	Reinforcement learning paradigm	11
Fig 2.2(a)	System level perspective of multi-robot search & rescue systems	13
Fig 2.2(b)	Algorithmic perspective of multi-robot search & rescue systems	13
Fig 3.1	Sensor footprint	17
Fig 3.2	3 kinds of sensors	18
Fig 3.3(a)	Accurate sensor	19
Fig 3.3(b)	Large area sensor	20
Fig 3.4	World map & agent sensing footprint	20
Fig 3.5	Probability & uncertainty maps	22
Fig 4.1	Voronoi partitioning	25
Fig 4.2	An example of area decomposition	26
Fig 4.3	Results with 2 distinct sensors	26
Fig 4.4(a)	Map decomposition search	27
Fig 4.4(b)	Decomposed map areas	27
Fig 4.5	Map decomposition results	28

Figure number	Title	Page number
Fig 5.1	Visualizing map resolution	33
Fig 6.1	Searching process	35
Fig 6.2	Exploration process	36
Fig 6.3	Heterogeneous Decomposition Results	37
Fig 6.4(a)	Task allocation –trial 1	38
Fig 6.4(b)	Task allocation –trial 2	38

Chapter 1

Introduction

With the rapid development of affordable robots with embedded sensing and computation capabilities, we are quickly approaching a point at which real-life applications will involve the deployment of hundreds, if not thousands, of robots. Among these applications, significant research effort has been devoted to multi-agent search, where deploying numerous agents can greatly improve the time-efficiency and robustness of search. Motivated by such problems, this project considers the deployment of heterogeneous robots in time-critical scenarios, where search can be improved by combining the different motion and sensing capabilities of the agents.

1.1 Background

The widespread use of intelligent agents such as robots, unmanned ground vehicles (UGVs) and unmanned aerial vehicles (UAVs) - owing to advancements in their capabilities as well as new control, perception, and estimation algorithms - in a variety of applications ranging from rescue to security to transportation to medicine has necessitated automated information processing and exploitation. An important class of problems in these intelligent systems is the detection and locating of objects of interest using intelligent search agents, known as **search problems** [1]. There are different classes of search problems. In terms of the targets being searched, there may be a single or multiple targets. The targets may be stationary or moving. In terms of number of search agents, there are single-agent or multi-agent search problems. The agents would also have a variety of constraints:

- A search agent usually has a limited amount of resources or search effort that it can spend, so there is a *budget constraint*.
- It may have limited area of influence or sensing range, so *visibility constraints* which only allow the agent to search a subset of the search space can be considered.
- There may also be *motion constraints* which restrict the agent to searching only some locations right after searching one location. In addition, when the agent moves from one location to another, a *switching cost* that is not negligible comparing to the search cost may be incurred.

A multi-agent system describes multiple distributed entities which take decisions autonomously and interact within a shared environment. Each agent seeks to accomplish an assigned goal for which a broad set of skills might be required to build intelligent behavior. Depending on the task, an intricate interplay between agents can arise such that agents start to collaborate or act competitively to excel opponents or achieve targets faster. Specifying intelligent behavior - a-priori through programming - is a tough, if not impossible, task for complex systems. Therefore, agents require the ability to adapt and learn over time by themselves. The most common framework to address learning in an interactive environment is *reinforcement learning* (RL), which describes the change of behavior through a trial-and-error approach. However, some conventional methods which do not rely on any form of learning do exist, and shall be detailed in the following sections.

Using multiple mobile robots in search tasks offers a lot of benefits over single-agent searching, but one needs a suitable and competent motion control algorithm which is able to consider sensors characteristics, the uncertainty of target detection and complexity of needed maneuvers in order to make a multi-agent search autonomous.

Many approaches have been proposed thus far to search for unknown targets using a team of agents. The most straightforward way to approach this problem is through *geometric searching*, which is very useful when lacking any a-priori information about the likely positions of targets [2], [3]. On the other hand, when such a-priori information is available, a decentralized or gradient-based method can be applied by exploiting the potential field that describes the likely target positions [4]–[7]. However, gradient-based methods like information surfing are sensitive to noise and can drive agents to local maxima instead of global maximum, which also reduces efficiency. When we want to make full use of the information and find a more optimal solution in a long-term way, optimization-based methods can be used [8]. Through maximizing the gathered information, agents can work in a more efficient way but since these methods are usually centralized, they lose scalability to larger teams.

In this project, we seek a decentralized solution to multi-agent collaborative search where agents

have individual beliefs about the world and make individual decisions. However, they would still build a common ‘map’ of the area, enriching it with information while coordinating with each other, divide areas of the map to be searched by each agent, allocate tasks accordingly and balance exploration of new areas with exploitation of all the information about the known areas. Moreover, we wish to utilize heterogeneous agents, with different sensing and motion capabilities to search the area faster and more effectively, while also improving on the efficiency of the whole process. This leads us to the definition of the problem that this report is based on.

1.2 Problem definition

This project is based on exploiting the heterogeneity of the agents – the fact that they have different kinds of sensors and different motion capabilities – to search faster and more efficiently for targets in an area. The intuition behind heterogeneous search is considering different sensor capabilities, we want to find and demarcate areas that are better suited to particular agents, then ensure that agents spend more time in areas they’re best suited for searching. The project describes a real-world searching task, with a search team having heterogeneous composition and robustness to environmental effects (e.g., occlusions like trees for drones, noise elements like wind, etc.). The search operation terminates when all targets are found (or marked on the map, which is generated by the agents). We start with no prior information about the environment and conduct an initial scan by quickly scanning a large area at a time, but not in-depth. As we start getting demarcated areas where the probability of finding the target(s) is higher, agents with accurate-but-slow sensors move to these areas and resolve the areas further, while the broad-but-fast agents continue exploration. The agents form groups to search particular areas and enrich information on a common map.

Aim: To search for static targets in an unknown environment, using a team of heterogeneous robots having different motion and sensing capabilities, employing reinforcement learning to distribute the agents efficiently and minimize searching time.

Objectives of the project:

- Establish basic framework for heterogeneous searching
- Represent the map with minimal error from ground truth

- Decompose the map optimally – optimal task allocation
- Implement RL-based trajectory optimization
- Show that the solution scales to large number of agents and find (approximate) complexity or relation of searching time to number of agents

We first formulate the problem into a mathematical discretized model that is suitable to cast as a reinforcement learning (RL) problem. We divide the world into a two-dimensional discrete map making up of a number of unit cells, and each cell contains some information comprised of targets' prior probability and uncertainty, target status, and agent position. The probability implies to the likelihood of target existence. The uncertainty represents how confident we are about the prior probability levels in a given area. As for the agents, we give each a local belief of the information and the ability to do several specific actions to move to an adjacent cell. Agents also have visual and sensing characteristics, and they change the environment and update their local beliefs accordingly.

1.3 Related work

The intersection of multi-agent systems and reinforcement learning holds a long record of active research. Survey papers on the topic [9], [10] describe in detail the various frontiers of research within the field and the various approaches present to tackle searching problems. Prior work done in this field which is relevant to the present task is detailed in the second ('Literature Review') chapter. This section is dedicated to description of the prior work done at MARMot Lab, NUS which is relevant to this project, and helped the project take shape. Amongst the several teams working at the lab, this project comes under the 'Search' team. Research at the lab on multi-agent searching is further divided into 3 lines of work as detailed in the figure on the next page.

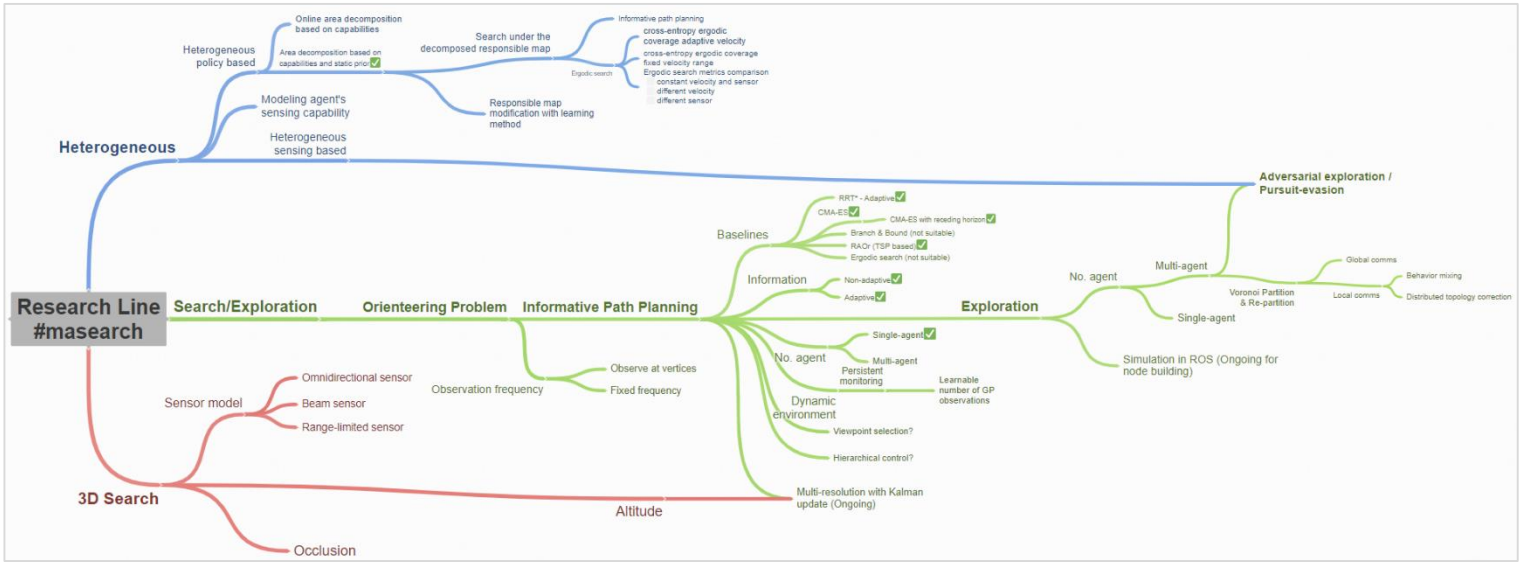


Fig. 1.1: *Branches of the multi-agent search research line at MARMot Lab, NUS*

Research topics under the Multi-Agent Search project at MARMot Lab NUS:

- Heterogeneous Search *(which this project is a part of)*
- Search/ Multi-agent exploration using informative path-planning
- 3D Search

Towards multi-agent search (MAS) problem, the lab had previously investigated the scenario where plenty of stationary targets are searched by a group of agents with limited sensing and communication capabilities in a two-dimensional grid world. The MAS problem was developed and extended to more aspects and organized into three primary domains:

- Heterogeneous search. A variety of team members with different capabilities conduct a search task, optimizing the overall search efficacy.
- 3D search. More realistic models regarding sensors, line-of-sight, angle of view, etc. will be considered to prepared for real-world applications.
- Pursuit-evasion. Targets (evaders) with moving ability will try to escape from the capture of the pursuers, where a cooperative strategy should be devised by both teams.

1.4 Structure of the document

This report document is organized in five main parts, besides this introduction chapter.

Chapter 2 consists of the Literature Review done to establish the problem statement and develop the understanding of concepts pertinent to the study.

Chapter 3 explains the starting point of the project, ergodic search, and describes the process of searching and formally defines sensors, uncertainty, probability and other concepts. Homogeneous multi-agent searching is also described here.

Chapter 4 of this thesis introduces heterogeneity and elaborates upon the various advantages and complications that brings to the searching task. The convolution approach adopted initially to solve the searching task is detailed herein, along with important steps during the project like map decomposition.

Chapter 5 of the thesis deals with heterogeneous multi-agent searching with reward-based trajectory optimization, i.e., the addition of reinforcement learning to the search process.

Chapter 6 is a repository of all the results and tires to succinctly describe the results of each approach adopted over the course of this project through graphs.

Chapter 7, the final part of this thesis, draws insight from the conclusion and outlines the future work – already planned future work at MARMot lab, and further possible future work in various areas of the project – and describes the impact of the work on the task of multi-agent searching, all from an undergraduate student's perspective.

Chapter 2

Literature Review

This section focuses on previous works on multi-robot search methods, as well as multi-agent path finding (MAPF) based on deep reinforcement learning (DRL). In this work, we are inspired by existing search methods and present a new method that relies on DRL to improve search efficiency and scalability.

An extensive literature review was conducted, covering papers that surveyed multi-agent systems, and research work that described conventional or mathematical approaches to multi-robot searching based on graphs, Voronoi partitioning and information-surfing, or special cases of pre-specified distribution of agents. The literature review also included papers on reinforcement learning based heterogeneous as well as homogeneous MA-search methods.

Research work which give a background on MA systems, searching, motion control, reinforcement learning, communication, etc. was also studied and has been noted in the literature review document for this project.

2.1 Multi-agent searching (MAS) – Conventional approaches

Multi-agent search is a central robotics problem, which considers search for different kinds of targets and under different conditions.

Multi-agent search is a central robotics problem that considers search for different kinds of targets and under different conditions. Many search strategies have been summarized in [11], which illustrate the underlying theory in single-agent searching for single or multiple, static or moving targets. Furthermore, collaborative search has been drawing researchers' interests [12]–[16], whose methods have been relying on a variety of tools such as team-optimal, dynamic programming and distributed control.

In general, there are two broad classes of search methods, depending on the availability of a priori information about the likely location of targets. Methods like geometric coverage are applicable in some situations where no information can be acquired, and agents move to cover all the areas of this region.

However, more advanced collaborative search methods can exploit prior information if it is

available. First, the gradient-based method has been proposed in [2], [17], [18]. From the likelihood of targets, agents search greedily by driving agents to local information maxima. Masoud et al. proposed a PSO-based multi-robot cooperation method to search targets where they assume the target emits a signal like heat for the robot to sense it and determine a locally favorable direction. PSO does not use the gradient of the problem being optimized, and it does not guarantee an optimal solution. There are also many other decentralized search strategies. Chung et al. formulated a multi-agent decision-theoretic of probabilistic search problem [19] . Furukawa et al. [20] presented a control technique which uses recursive Bayesian filtering to autonomously search and track multiple targets with distributions and probabilistic motion models. These methods cannot be applied to real-life scenarios easily with large agent teams, and our work looks for scalable, decentralized multi-agent search methods to control the agents searching more intelligently.

A Graph Theoretic-Based Approach for Deploying Heterogeneous Multi-agent Systems is detailed in [21]. This approach divides the environment and creates a graph using Voronoi partitioning with weights based on capabilities of every heterogeneous agent *(more capable robot gets the more important area)*. Areas are marked on a common map, and the proces starts with no a-priori information. Hence, this work is very relevant to the present problem. However, because the algorithm is graph-theory based, it has its own limitations and the paper describes regularly spaced areas only, with strictly short-range communication.

Another recent work [22] describes a distributed terrain coverage algorithm that employs Voronoi partitions to divide the area of interest among the robots and then uses a single-robot coverage algorithm to explore each partition for potential targets. Then, it describes multi-robot task allocation algorithms that use the location information of discovered potential targets and employs either a greedy distance based strategy, or an opportunistic strategy (stochastic queueing based model) to allocate tasks among the robots while attempting to minimize the time (energy) expended by the robots to perform the tasks.

[23] presents a decentralized ergodic control policy for time-varying area coverage problems for multiple agents with nonlinear dynamics. Ergodic control allows the specification of distributions as objectives for area coverage problems for nonlinear robotic systems as a closed-form controller. The paper derives a variation to the ergodic control policy that can be used with consensus to enable a fully decentralized multi-agent control policy.

[24] addresses the problem of coordinating and controlling multiple robots for the detection of multiple dynamic anomalies in the environment. They propose a combined approach for effective exploration under uncertainty, anomaly tracking, and autonomous on-line allocation of agents. Robots explore the work area maintaining the history of the sensed areas to reduce redundancy and to allow for full-map coverage. When an anomaly is detected, a robot autonomously determines how to either track the anomaly or to continue the exploration of the environment, depending on the size of the anomaly, which is estimated by the length of the perimeter of the enclosing polygon.

Multi-Agent Path Finding (MAPF) is an NP-hard *(non-deterministic polynomial)* problem even when approximating optimal solutions. Here we want to emphasize the decentralized learning MAPF planner where agents learn their own policy and can easily implement multi-agent systems. Some DRL-based MAPF decentralized planners show great potential in solving the MAPF problem. For example, Sartoretti et al. [25] proposed pathfinding via reinforcement and imitation multi-agent learning (PRIMAL), a new framework for decentralized, reactive MAPF. They showed that PRIMAL worked well in low obstacle densities situations, which combined the advantages of distributed reinforcement learning and imitation learning. As we want to solve the multi-agent search problem in a decentralized way, such distributed learning-based approaches seem like a good starting point for us to base this work upon.

2.2 Heterogeneous MAS

This section describes the unique perspectives, advantages and challenges that heterogeneous agents bring to the MAS problem. [26] provides a methodology for an autonomous two-dimensional search using multiple unmanned search agents. The proposed methodology relies on an accurate calculation of target occurrence probability distribution based on the initial estimated target distribution and continuous action of spatial variant search agent sensors. The core of the autonomous search process is a high-level motion control for multiple search agents which utilizes the probabilistic model of target occurrence via Heat Equation Driven Area Coverage (HEDAC) method. This centralized motion control algorithm is tailored for handling a group of search agents which are heterogeneous in both motion and sensing characteristics. The motion of agents is directed by the gradient of the potential field which provides near- ergodic exploration of the search space.

This paper by [27] develops a multi-agent heterogeneous search approach that leverages the sensing and motion capabilities of different agents to improve search performance (i.e., decrease search time and increase coverage efficiency). It draws on recent work on ergodic coverage approaches for homogeneous teams, in which the agents' search pathways are tuned so that they spend time in regions proportional to the predicted likelihood of finding targets while still covering the entire domain, balancing exploration with exploitation. This work presents a new strategy for extending ergodic coverage to groups of heterogeneous agents with different sensing and mobility capabilities. Methods for efficiently assigning available agents to distinct regions of the domain and optimally matching the agents' capabilities to the scale at which information needs to be looked for in these regions are examined.

2.3 Reinforcement learning (RL) based Heterogeneous MAS

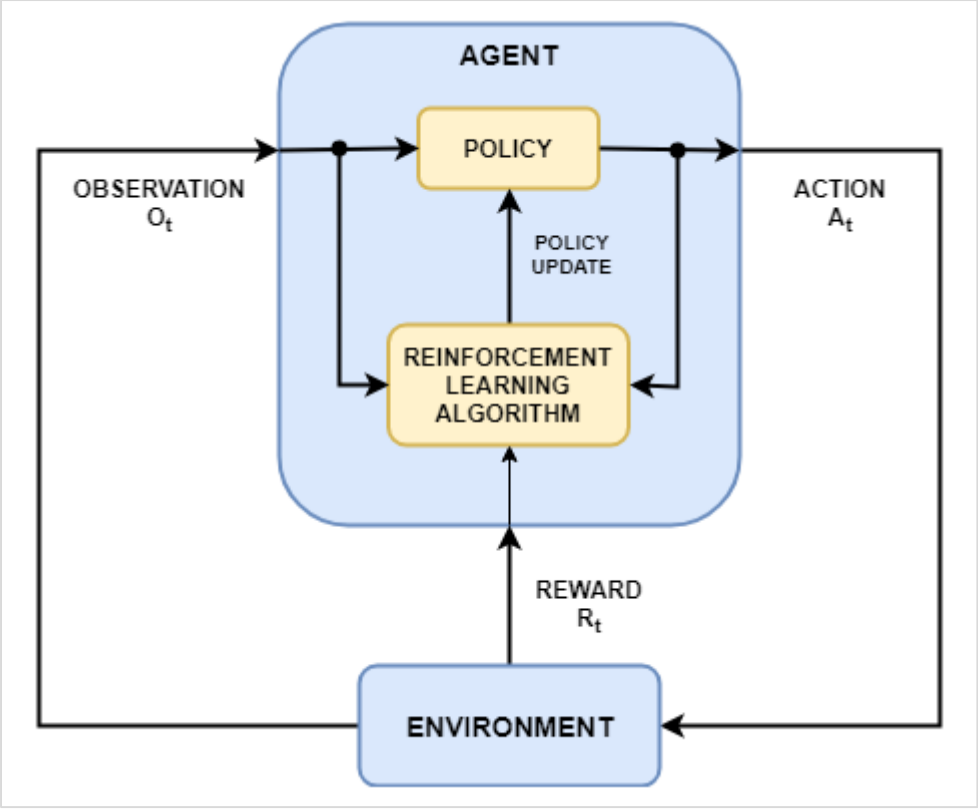


Fig 2.1: *Reinforcement learning paradigm*

Image Source: *"What is Reinforcement Learning?" - mathworks.com*

Link: *https://in.mathworks.com/help/reinforcement-learning/ug/what-is-reinforcement-learning.html*

Reinforcement Learning tackles the subject of how an autonomous agent may learn to pick and

conduct optimal actions to attain its goals based on its present environment. It accomplishes this by modelling the learning task as a Markov Decision Process (MDP). Every action taken by the agent causes a change in the environment, which might be either good or unwanted, as reflected by the reward the agent gains at every step. The agent gradually learns the behavior of the environment through repeated interactions, and then adapts and optimizes its own actions to maximize desirability and achieve the goal, i.e., earn maximum possible reward. The advances in reinforcement learning have recorded sublime success in various domains. Although the multi-agent domain has been overshadowed by its single-agent counterpart during this progress, multi-agent reinforcement learning gains rapid traction, and the latest accomplishments address problems with real-world complexity. We start with a preliminary survey of heterogeneous multi-agent systems and consult three survey papers.

[9] surveys recent contributions, highlights current state-of-the-art methods on existing multi-agent robotic searching systems and identifies their limitations, remaining challenges, and possible future directions. The paper gives special emphasis on challenges of MAS sub-fields like task decomposition, coalition formation, task allocation, perception, and multi-agent planning and control. However, the survey does not cover credit assignment / reward distribution among agents, consensus or agent agreement under different circumstances, containment control (which is a type of consensus in leader-follower models), communication protocols and efficient information exchange strategies and hardware design.

[10] provides an overview of the current developments in the field of multi-agent deep RL. The paper analyzes the structure of training schemes that are applied to train multiple agents. The authors have considered the emergent patterns of agent behavior in cooperative, competitive and mixed scenarios. The paper systematically enumerates challenges that exclusively arise in the multi-agent domain and reviews methods that are leveraged to cope with these challenges.

[28] provides a review of multi-robot systems supporting Search & Rescue (SAR) operations, with system-level considerations and focusing on the algorithmic perspectives for multi-robot coordination and perception. This survey paper covers heterogeneous SAR robots in different environments, active perception in multi-robot systems, while giving two complementary points of view from the multi-agent perception and control perspectives. It presents a literature review of multi-robot systems (MRS) for SAR operations with a focus on coordination and perception algorithms and, specifically, how these two perspectives can be bridged through different active

perception approaches.

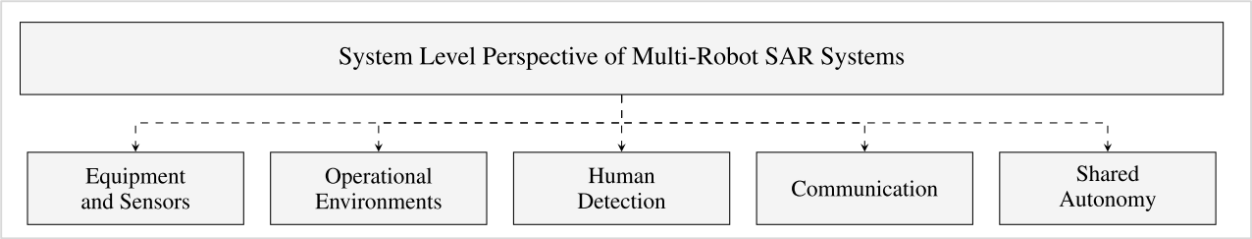


Fig 2.2(a): *System level perspective of multi-robot search & rescue systems*

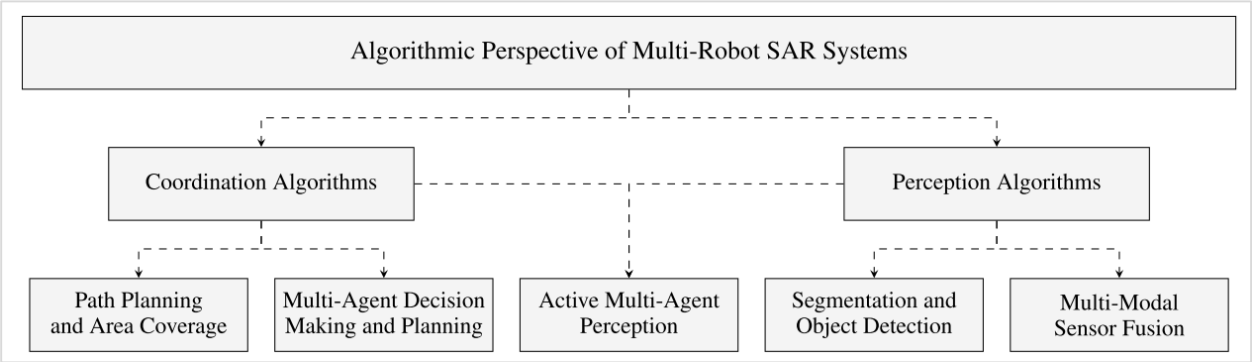


Fig 2.2(b): *Algorithmic perspective of multi-robot search & rescue systems*

Images source: *Queralta, J. P., Taipalmaa, J., Pullinen, B. C., Sarker, V. K., Gia, T. N., Tenhunen, H., Gabbouj, M., Raitoharju, J., & Westerlund, T. (2020). Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision. IEEE Access, 8, 191617–191643. https://doi.org/10.1109/ACCESS.2020.3030190*

Apart from these survey papers, several other papers that describe RL based heterogeneous multi-agent search methods were studied.

[29] presents an actor-critic algorithm that allows a team of heterogeneous agents to learn decentralized control policies for covering an unknown environment. They augment a multi-agent actor-critic architecture with a new state encoding structure and triplet learning loss to support heterogeneous agent learning. This paper is relevant because the topic is very close to that of the present work, and the simulation environment used in this paper includes real-world environmental factors such as turbulence, delayed communication, and agent loss, to train teams of agents and probe their robustness and flexibility to such disturbances.

[30] deals with cooperative multi-agent exploration. Exploration is critical for good results in deep reinforcement learning. However, existing multi-agent deep reinforcement learning

algorithms still use mostly noise-based techniques. Very recently, exploration methods that consider cooperation among multiple agents have been developed. However, existing methods suffer from a common challenge: agents struggle to identify states that are worth exploring, and hardly coordinate exploration efforts toward those states. To address this shortcoming, in this paper, the authors propose cooperative multi-agent exploration (CMAE): agents share a common goal while exploring. The goal is selected from multiple projected state spaces via a normalized entropy-based technique. Then, agents are trained to reach this goal in a coordinated manner.
[31]–[33] also provide important insights into the field of multi-agent reinforcement learning.

Chapter 3

Heterogeneous sensor capabilities

"*A variety of team members with different capabilities conduct a search task, optimizing the overall search efficacy by synergistically leveraging/combining their individual strengths*."

A majority of the current methods investigated in the literature review are either about homogeneous agents or using the same policy, under the same action space. There are very few papers describing heterogeneous multi-agent searching, and many of those focus more on improving communication, etc. However, it is clear from the literature that heterogeneous multi-agent searching is an important emerging field and with the help of reinforcement learning, has the potential to lead to state-of-the-art performance on complicated tasks. In some scenarios, heterogeneous agents may help accelerate the search process by a huge margin. For example, ground vehicles may move slower but can be equipped with much more accurate sensors whereas UAVs are faster but vague in image resolution. Together, they can decompose a map (i.e., find which regions of the map are suited for searching by which agents) much faster, and thereby arrive closer to the ground truth faster. The intuition behind heterogeneous search is considering different sensor capabilities, and we want to find an online or predefined area decomposition to guide agents to search efficiently on how and where to put their sensor on without many optimizations.

Some work studied during the literature review proposed heterogeneous search methods according to frequency like [27], they decomposed a given domain based on high or low frequency and assign these areas to different types of agents. But in this approach, the information may get distorted because of the FFT and its inverse process, sensor capability may also be hard to tune to find the optimal solution. Therefore, we want to find a method that could apply the same decomposition idea to the spatial domain and remain the same accuracy of the maps, which also easy to tune the parameters because the sensor property can be easier to define in the spatial domain.

"Ergodic Search" project, which had been done previously at CMU's Biorobotics lab in collaboration with MARMot Lab, NUS, was chosen as the starting point for the current project.

This approach relied on an a-priori information distribution, representing the likelihood of finding a target at any point over the search domain, to guide the search. By optimizing over parameters that describe the search paths, the ergodic metric drove agents to spend time in areas of the domain in proportion to the a-priori likelihood of finding targets in these areas, while still covering the whole domain, thus balancing exploration and exploitation. The code for this project was available, but written entirely using MATLAB. So, the preliminary step for the current project was to convert this codebase to python, which was the preferred language for future work due to its rich library of modules and general ease of use. Once that was completed, the next move was to make the agents start exploration from an empty prior.

The simplest case of "heterogeneous" agents, i.e., 2 distinct agents searching an area, is the task for the upcoming section. We have defined 2 sensors and describe the fusion of their data to get closer to ground truth.

3.1 Sensor characteristic definition

In this section, we define some simple sensors, where a Gaussian distribution can suitably represent the sensor capability. The sensor footprint is represented as a Gaussian distribution because a camera's center captured images containing more details; inversely, more distortions in the edge of images.

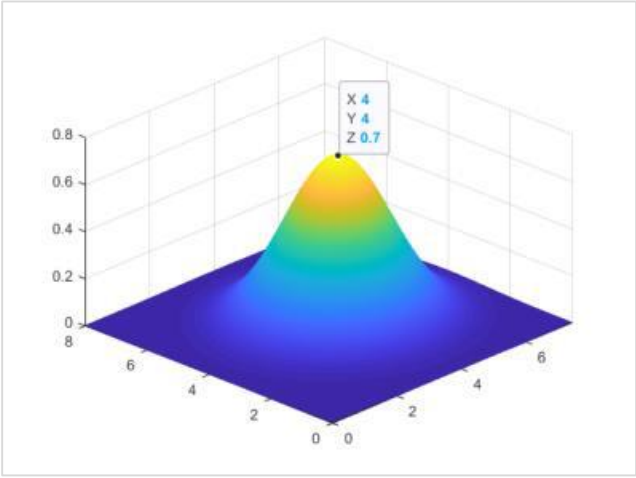


Fig. 3.1: *Figure denoting a sensor with an 8x8 unit cell footprint, with maximum detail at the center (4,4) and uncertainty increasing radially outward from this point.*

The size and variance of the distribution could be tuned to represent the accuracy and area property of the sensor. If the sensor is spike-like, that means it might be a beam sensor and has

great accuracy in the center. Similarly, a larger and smoother curve denotes higher variance and lower accuracy, and could be a camera-based sensor for example. Each sensor is denoted by one Gaussian Process (GP) and has its own kernel, and we want to optimize the trajectories of the agents to reduce searching time. Kernel here refers to the same thing as a kernel in convolution operation, i.e., a matrix of values covering a small portion of the actual data matrix. Each sensor is basically giving a Gaussian distribution over its sensing area, which is being used as the convolution matrix for the CNN.

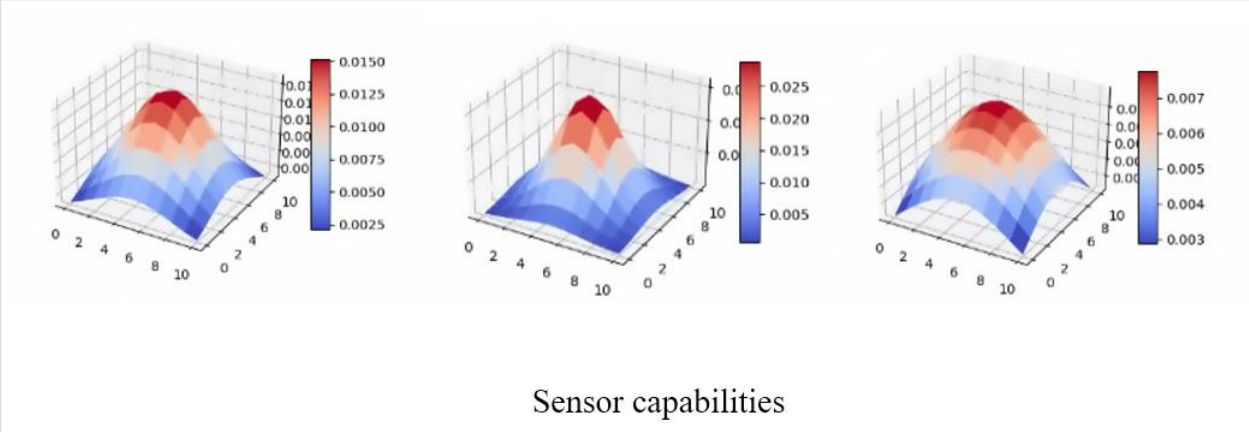


Fig. 3.2: *3 kinds of sensors described using graphs*

These three graphs above each represent a sensor. The middle one has a sharp peak which denotes that it has high accuracy at the center but a lower area of coverage (e.g. a beam sensor). The one on the right is a gradual curve and denotes that it has a high coverage but lower accuracy (e.g. camera).

3.2 Sensor Data fusion using Gaussian processes

Through decomposing the map into different areas based on heterogeneity, we want to guide agents to search more efficiently. Accurate sensor better suits for exploitation whereas larger sensor is good at coverage and exploration. By decomposition, we hope to combine the advantages for different sensor and speed up the convergence of agents' belief. For this, it is necessary to be able to fuse the data from two sensors in a sensible and coherent manner, so that it can give better information about the area. Merging GPs by taking mean, weighted average, incorporating a smoothing factor and Kalman Filtering were tested, to improve the data fusion and get closer to ground truth.

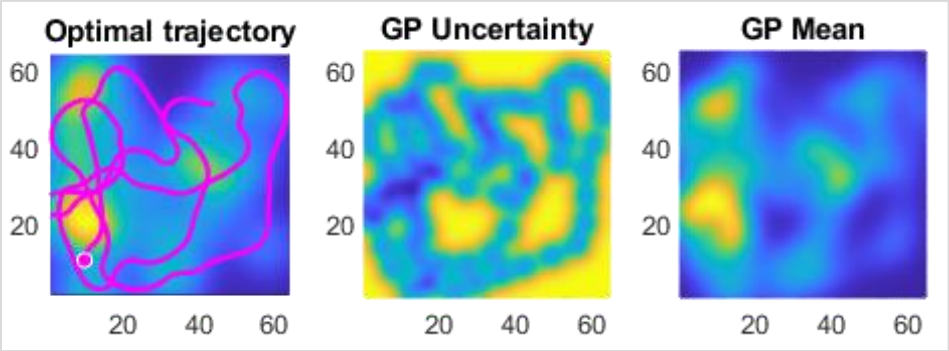
A simple mean implies taking the mean of all the measurements taken at the same spatial position by each agent.

A weighted average was taken on the following basis:

If *info_pred[i]* denotes the measurement of the *i'th* agent at some fixed point in the spatial domain, say (x_0, y_0), then

$$weight[i] = \frac{info\_pred[i]}{sum(info\_pred[i])}$$

Below are images of two kinds of sensors. One is an accurate sensor, that can only scan a small area at a time. It reduces the uncertainty in that area by a large amount, but since its sensing range is small, so is the footprint it leaves and therefore, the uncertainty in the parts it does not visit remains high. Taking a mean at every spatial point for all the measurements the agent has done, we can create our 'map'. This is denoted below by GP mean. The closer this map is to the groud truth, the better our agent is doing. The ground truth is given in the first image as the background, on which the trajectory of the agent is superimposed. The yellow areas in the ground truth map are areas where the probability of finding a target is high, while the blue areas are low-probability regions. Thus, the groud truth is basically a probabilty map which we want to recreate.



Fir. 3.3(a): *Accurate sensor: 3.3.1 denotes the trajectory taken by the agent, superimposed on the ground truth (actual situation); 3.3.2 shows the uncertainty map when the agent is at the given position after taking the shown trajectory; 3.3.3 shows the output map by the agent.*

Since this agent is accurate, it needs a longer time and more number of measurements (larger number of sampling points) to build an image of the map. But, the image it finally builds is close to the ground truth.

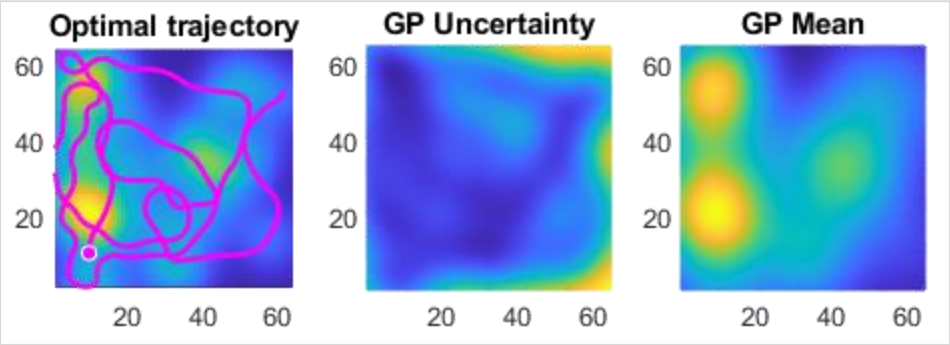


Fig. 3.3(b): *Large-area sensor: 3.3.1 denotes the trajectory taken by the agent, superimposed on the ground truth (actual situation); 3.3.2 shows the uncertainty map when the agent is at the given position after taking the shown trajectory; 3.3.3 shows the output map by the agent.*

Since this agent is sensing a large area at a time but is not as accurate as the one above, it needs a much shorter time and fewer sampling points to build an image of the map. But, the image it builds is not as close to the ground truth as the accurate sensor. Do note, however, that the map is generated much faster here, and uncertainty is lowered in a large area of the map very quickly.

The world map and agent sensing footprint can be discerned visually from the following figure:

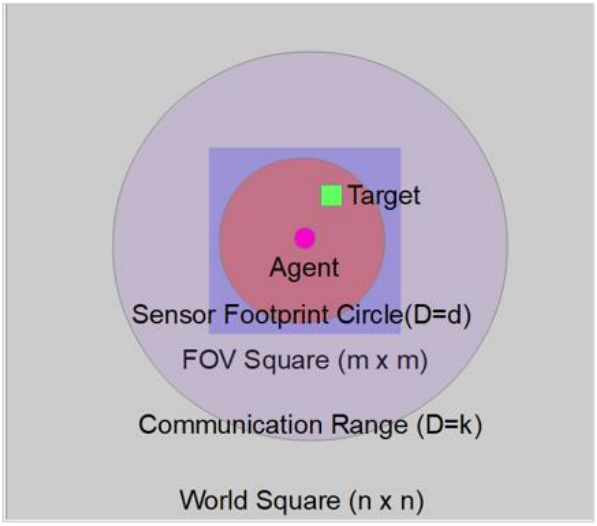


Fig. 3.4: *World map & agent sensing footprint*

For a larger world, agents should cover more areas to find the targets. Agents need to learn how to work collaboratively in different world sizes.

3.3 Probability and Uncertainty maps

These two maps are prior information maps we defined to help agents locate the possible positions of targets. We randomize these maps at the beginning of each episode.

We want to use a multimodal map with several local maxima to represent the information, so we create a map containing several Gaussian distributions. Considering the world size n in the previous section, we add m *($m \in [16,32]$)* Gaussian distributions into the map. For each distribution, the value in each cell of the discrete map can be calculated as equation below:

$$f_m(x,y) = \frac{e^{-\frac{1}{2(1-\rho_m^2)}\left[\frac{(x-\mu_{Xm})^2}{\sigma_{Xm}^2}+\frac{(y-\mu_{Ym})^2}{\sigma_{Ym}^2}-\frac{2\rho(x-\mu_{Xm})(y-\mu_{Ym})}{\sigma_{Xm}\sigma_{Ym}}\right]}}{2\pi\sigma_{Xm}\sigma_{Ym}\sqrt{1-\rho_m^2}}$$

where μXm and μYm are the mean of the distribution and reflect the position of the maximum in this distribution. ρm is a correlation between X and Y; $\sigma Xm2$ and $\sigma Ym2$ express the area of the distribution. Then we sum all the distributions and maps the probability in each cell to the range [0,1] as

$$\mathrm{p}(x,y) = \frac{\sum_{i=1}^{m} f_i(x,y)}{Max(\sum_{i=1}^{m} f_i(x,y))}$$

And the uncertainty map is created in the same way as the probability map:

$$\mathrm{u}(x,y) = \frac{\sum_{j=1}^{m} f_j(x,y)}{Max\left(\sum_{j=1}^{m} f_j(x,y)\right)}$$

The resulting probability map might still contain areas with near-zero levels, which we would like to avoid to encourage exploring the whole domain. Thus we also add an option to set a minimum value of probability with $\eta \in [0,1]$ to every cell. This is what we term a "baseline" of probability. For example, if the baseline value η=0.1, then each cell will have at least a 10% probability to contain a target. An example image of probability and uncertainty maps can be found on the next page.

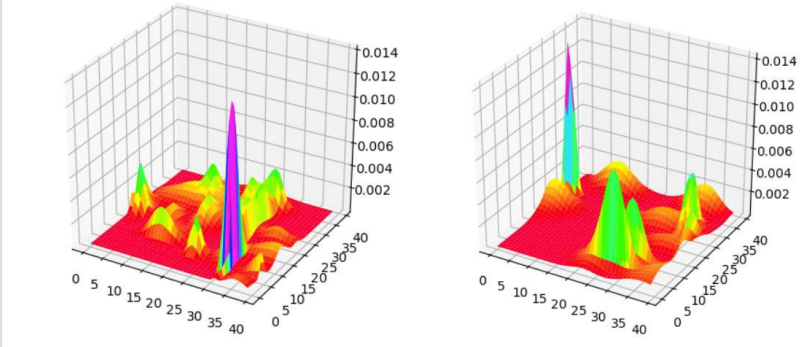


Fig. 3.5: *Sample probability and uncertainty maps*

Chapter 4

Heterogeneous Multi-Agent search using CNN approach

When a group of heterogeneous agents vary in sensor capabilities, detection accuracy, moving speed, we want to take this heterogeneity into account when we distribute/allocate the tasks.
There are several methods of distributing tasks and dividing the area of the map to be searched by each agent:

- Fourier transform: only in frequency domain *(very brief overview given below, beyond the scope of this project)*
- Voronoi diagram: only in spatial domain *(brief overview given below, part of previous work at MARMot Lab, Nus but not a part of this project)*
- Considering methods in both spatial domain and frequency domain:
    - Convolutional kernel *(which will be elaborated in this section)*
    - Wavelet transform

4.1 Fourier transform

Some new detail elements are introduced to the map as a result of the Fourier transform. The spatial information in the two maps is not included. The spatial information is basically being converted to a different domain and then interpreted. Sensor specifications are not taken into account.

4.2 Voronoi partitioning

Voronoi partitioning refers to the partitioning of a plane with n points into convex polygons such that each polygon contains exactly one generating point and every point in a given polygon is closer to its generating point than to any other. Here, our generating point will be the starting position of an agent.

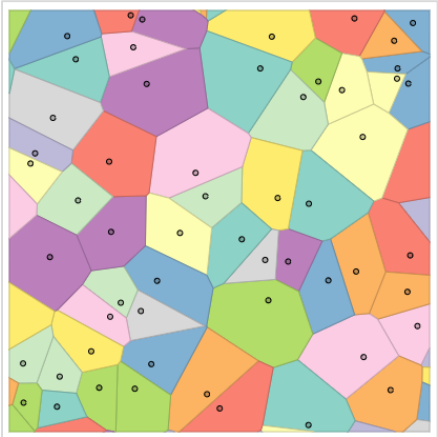


Fig. 4.1: *Voronoi partitioning*

A decentralized multi-agent search system always suffers from the unbalanced workload problem, which means the total workload hasn't been divided equally. For one multi-agent system, the perfect balanced workload leads to the highest working efficiency. However, balancing the workload for the multi-agent searching system is not easy. It can be influenced by multiple factors, like the target region and the initial locations of the drones. Hence, the area is decomposed into several sub-regions by Voronoi Diagram and forces the agents to work in their own sub-regions. This works well for homogeneous agents with perfect sensors, but falls apart when heterogeneity and uncertainty are introduced.

4.3 Spatial Map decomposition (based on Gaussian processes)

Gaussian processes are a powerful tool in the machine learning toolbox. They allow us to make predictions about our data by incorporating prior knowledge. Their most obvious area of application is fitting a function to the data. This is called regression and is used in various fields like robotics or time series forecasting. [34]

We adapted the idea of wavelet transform and convolutional neural networks. Through the predefined sensor capability, we can define it as a kernel which no longer needs to be trained to extract information from the static prior of the domain and get a responsible map of the different types of sensor. That is, for a highly accurate sensor, it will extract more information from the high probability area, which means it better suits for detailed exploitation, in contrast, the larger smoother kernel will extract area that is better for coverage and exploration. Thus, the model will decompose the map into 2 separate areas, each suited to be searched by a particular agent. An

example of the responsible map can be found in the figure below. The middle image represents the sensor type and the last image shows the velocity that best suits that area.

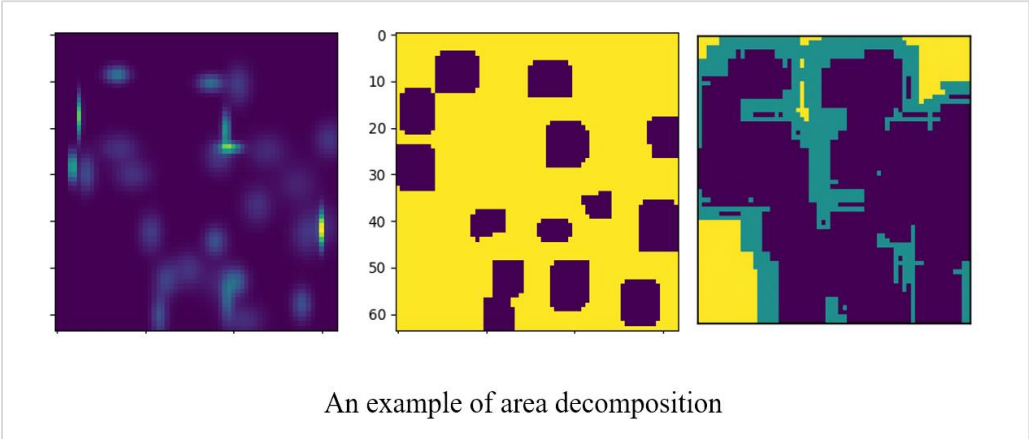


Fig 4.2: ***4.2.1*** *shows ground truth;* ***4.2.2*** *shows decomposed map with marked regions (responsible map);* ***4.2.3*** *shows velocities in each region, i.e., which regions should be covered quickly and which ones require detailed search.*

The figures below denote some results with two distinct sensors. Our goal is to get closest to the ground truth map within a time budget by leveraging the heterogeneity of the agents' capabilities.

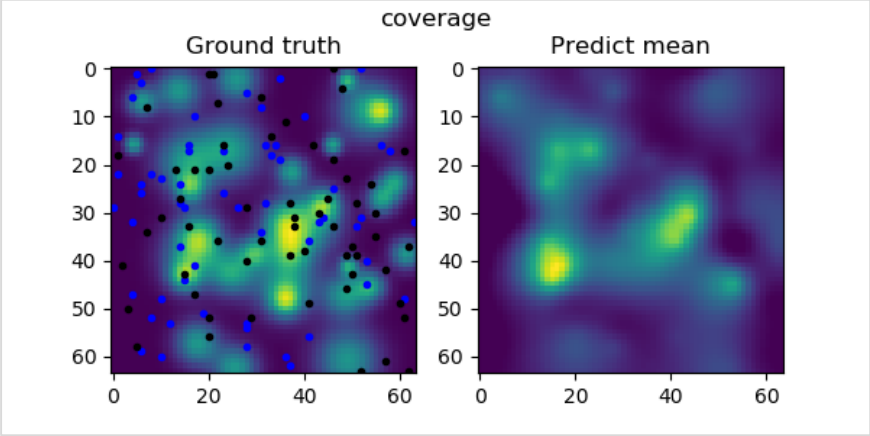


Fig. 4.3: *Results with 2 distinct sensors*

The map labeled 'ground truth' shows the actual probabilities of finding targets in the region and the dots on those maps are sampling points with the color denoting which sensor sampled which point. The map on the right titled 'predict mean' shows our prediction after the agents have sampled all these points. The goal here was just to cover the area as fast as possible and predict the result. We can see a visible difference between the prediction and the ground truth and

sampling points are seen to be random.

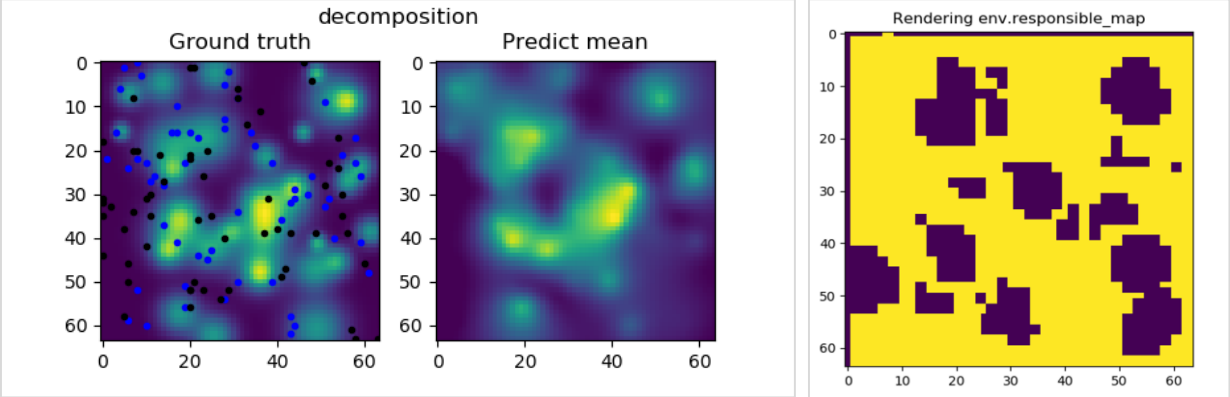


Fig. 4.4(a): *Map decomposition search;* ***Fig. 4.4(b):*** *Decomposed map areas*

Here, the goal was to decompose the map & then search, and the prediction can be seen to be markedly closer to the ground truth. The decomposed map is shown alongside, with yellow being the region to be searched by one sensor and purple being the region to be searched by another. The yellow region has low probability of finding target(s), so it must be searched by the sensor with wide coverage area, fast scanning speed but lower accuracy. The purple region on the other hand has high probability of finding targets, so must be searched by the slow moving but accurate sensor. Sampling points are in specific regions for each sensor – it can be seen that each agent is sampling within the areas marked in the decomposition map alongside.

In summary, through a convolution operation, we extract the weight for different sensors and create a responsible map to represent priority of these sensor in different area.

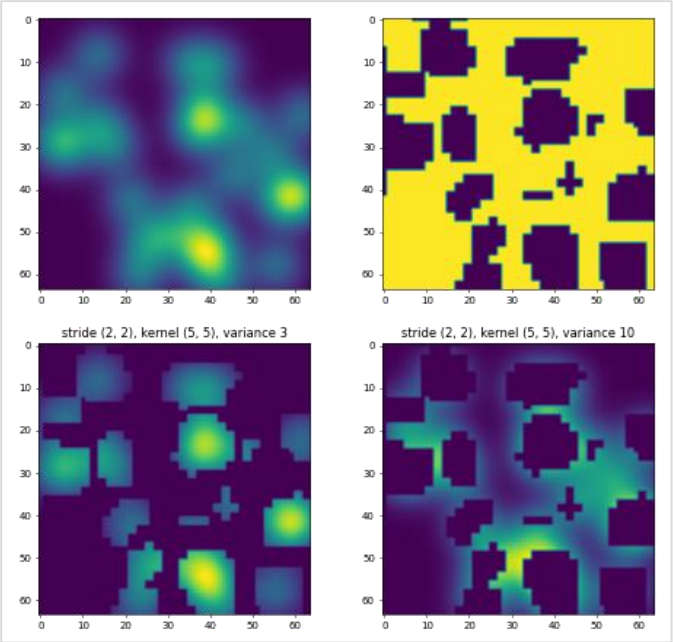


Fig. 4.5: Map decomposition results

In the figure alongside, we can see 4 maps.

Top left: information map, predicted by combining the maps of both the sensors

Top right: responsible map / decomposition map for sensor types

Bottom left: predictions for the area searched by the accurate sensor

Bottom right: predictions for the area searched by the broad sensor

4.4 Technical details of the model

- The model uses Kullback-Liebler divergence for map-decomposition testing, and cross-entropy as loss function.
- We also make use of Gaussian process regression (GPR) from sklearn.gaussian_process, which is based on an algorithm by Rasmussen and Williams. In addition to standard scikit-learn estimator API, GPR allows prediction without prior fitting & enables easier selection of hyperparameters.
- The 'kernel' of each sensor is the same as a convolutional neural network (CNN) kernel, and we are presently using a customized CNN for map decomposition. The matrix of observations of each sensor is convolved with a constantly-updating map, with appropriate weights for the process.
- Each sensor is basically giving a Gaussian distribution over its sensing area, which is being used as the convolution matrix for the CNN.
- If a sensor takes multiple measurements for a particular point with varied uncertainty, we apply a weighted average as described in the previous section.

4.5 Scalability testing and scope for improvement

So far, we have only been dealing with 2 distinct sensors. However, the same model can be scaled to several sensors simply by changing one input parameter – the number of agents of each type. Tests have been done for 2 fast and 2 slow agents, 4 fast and 8 slow agents and 16 fast and 16 slow agents. Provided that the area to be searched is large enough, it is observed that the efficiency increases and time to decompose the map and search each area decreases as the number of agents increases.

However, too many agents, especially of the fast but low-accuracy type, actually lead to a lower efficiency of the search process. In our tested cases, for a 64x64 or 128x128 map, 4 fast and 8 slow but accurate agents led to the best performance. Even though finding the optimal combination of agents is not the goal of this project, it is certainly an area for future work.

To prove the idea of the efficiency of the method, we have applied an optimization-based method, ergodic search. By cutting the prior into pieces and assigning them to different sensors, with less optimization iteration, we might be able to achieve the same or better performance as ergodic search. Secondly, this method needs to be generalized so that we can adapt the sensor capability and changing probabilistic information map and do online guidance of heterogeneous agents. Thirdly, implementing learning methods into our problem to replace the parts like how to generate the responsible map from the features we get will lead to better overall performance and a more ‘self-learning’ model.

Chapter 5

Heterogeneous Multi-agent search with reinforcement learning

The situation here is described below. These are the outcomes of the CNN based approach and now the task is to implement reward-based trajectory optimization for the agents. We now consider a realistic scenario. There are two types of agents - unmanned aerial vehicles (UAVs) and unmanned aerial vehicles (UGVs). UAVs have the advantages of their high maneuverability, which suits the tasks such as domain coverage, area surveillance. However, sensors like cameras, thermal imagers may produce inaccurate data at a high altitude. UGVs, on the other hand, can compensate for the weaknesses of UAVs by measuring from a close distance.

Exploration scenario:

- Agents start from an empty prior.
- The size of high information area varies a lot.
- Heterogeneous agents search this domain, and the objective is to decrease the searching time by coordination among agents.

Goal: Get an accurate belief of the map while using less measurement by pushing specific agents to specific areas.

Searching procedure:

1. Two heterogeneous agent start from random locations.
2. One moves faster and is equipped with a lager fuzzy sensor; another one is slower and equipped with small accurate sensor.
3. During search, agents take measurements and build belief.
4. We fuse the beliefs of these two agents and calculate where a specific agent better suits.
5. Redo step 3-4 until all targets are marked on the map.

5.1 Implementing reward structure

In order to implement reward-based trajectory optimization, i.e., add in reinforcement learning to our searching process, the way the trajectory of an agent is evaluated needs to be changed. A cost is associated with every trajectory step, and consequently every path that an agent takes. This cost is associated with:

- the position of the agent
- the region of the decomposed map that the agent is currently in
- the GP that the sensor has
- the type of agent
- the measurement at that point
- whether the agent is within the boundaries of the world

We implement a **region reward** for the agent first. The agent gets a positive reward if it is in the area of the map it is best suited to search, and a negative reward if it is in the area another agent is better suited to search. This reward also depends on the region boundaries.

A **measurement or sample reward** is added based on which point is sampled. If the measurement at that point leads to a higher probability, or lowers the uncertainty significantly in the map, than it gets a positive reward. Otherwise, it will get a negative reward.

We implement a map-boundary penalty. If any agent goes out of bounds of the world, then it will be penalized.

This reward is obtained at each step. We store this reward and find the sum of the rewards obtained throughout a particular trajectory to get the value function for that trajectory. The goal is to obtain maximum final reward (i.e. sum of all rewards at all steps), as in every reinforcement learning algorithm. Temporal difference learning is the reinforcement learning algorithm implemented to do this. Below is a short description of TD learning.

5.2 Temporal Difference Learning

TD Learning combines the ideas of Monte Carlo methods with those of dynamic programming (DP). Like Monte Carlo methods, TD methods can learn directly from raw experience without a model of the environment's dynamics. Like DP, TD methods update estimates based in part on

other learned estimates, without waiting for a final outcome. Hence, the next position the agent should go to in its current trajectory can be evaluated based on the immediate outcomes, without waiting for the whole trajectory to finish. This leads to faster updates at every step and implements an online trajectory planning of sorts.

5.3 Interactive Tool to see Map Resolution in Real Time

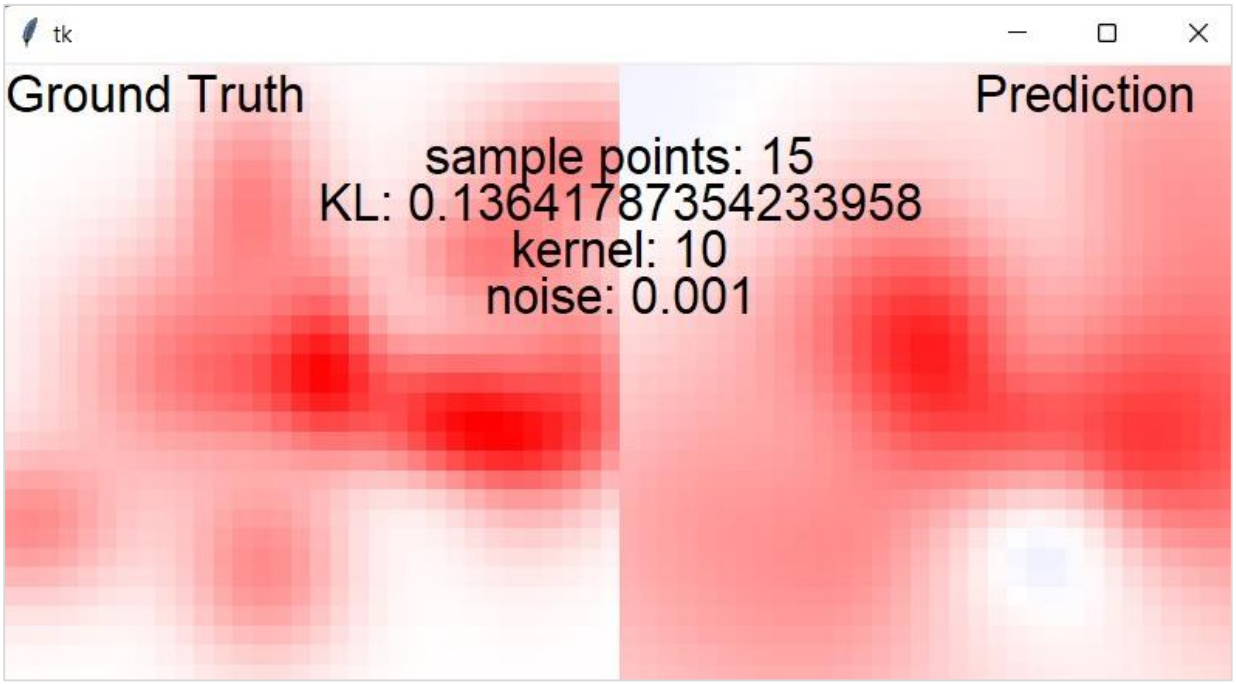


Fig. 5.1: *GUI for visualizing map resolution*

Using TKinter, a visual tool was made where clicking on different places on the map is like sampling of those points by an agent, and results in the prediction being changed after each click. The figure shows the result after 15 clicks on some points chosen by the user (randomly chosen by me in this case). It also shows the KL divergence of the current situation and the kernel size of the sensor. Noise here is the inherent noise in any sensor and can be changed by tweaking the number. This leads to more distorted or clearer maps, based on what the user inputs.

Chapter 6

Results & Discussion

The results obtained from each of the sections have been listed with the conclusions in the following sections.

6.1. Map Decomposition

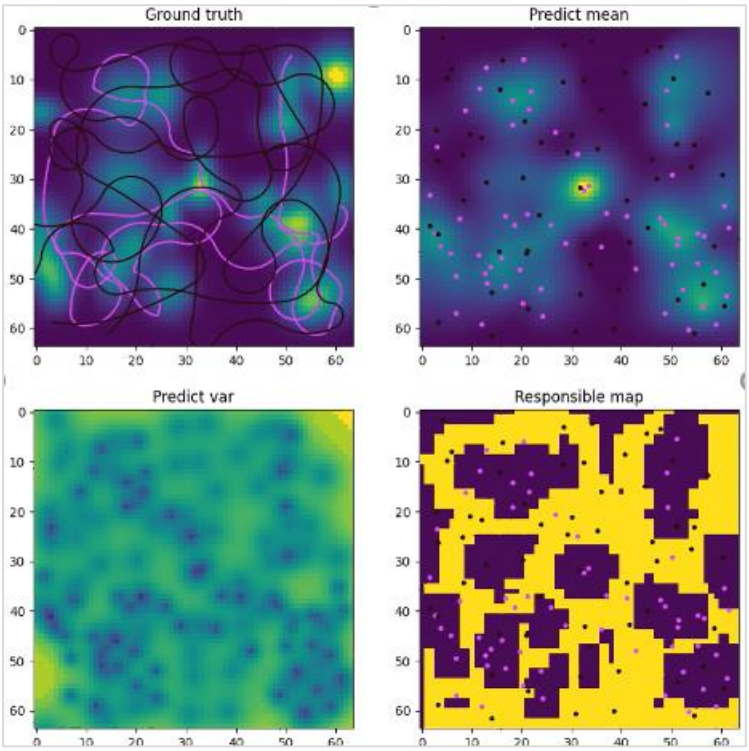


Fig. 6.1: *Searching process*

The trajectories of the agents are visible in the first figure labelled ‘Ground Truth’. The ground truth is given in the first image as the background, on which the trajectory of the agent is superimposed.The second figure shows the sampled points and how our prediction has evolved as more points have been sampled. The map decomposition gets better as the agents move and the final result is visible in the figure titled ‘Responsible map’. The variance of the entire region goes down as more points are sampled, as seen in the figure titled ‘Predict var’.

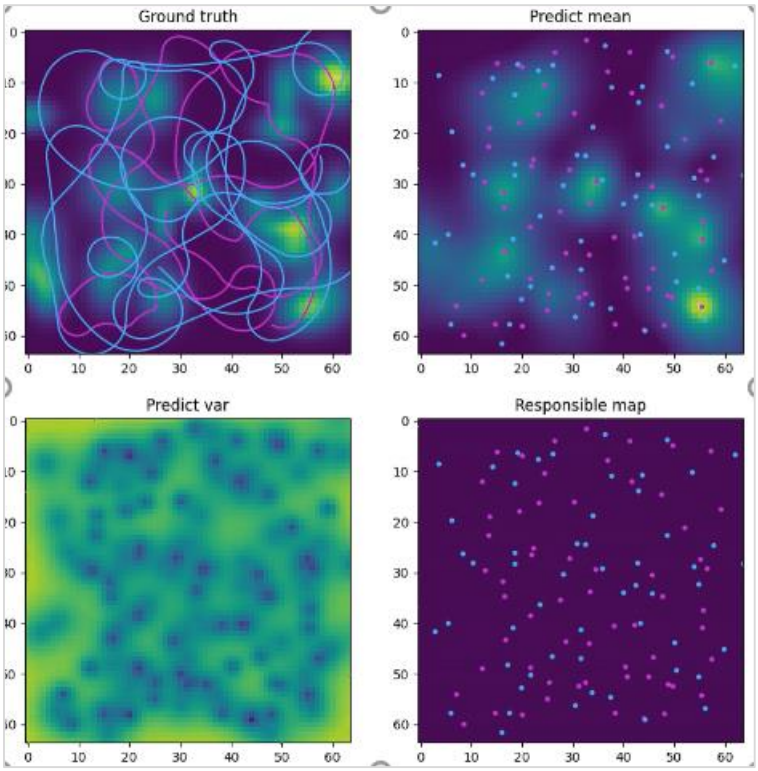


Fig. 6.2: *Exploration process*

A similar analysis is visible for the exploration process, where the map is not decomposed. Comparing the sampling points, we can see that agents are focusing on the areas they are better suited to search when we carry out map decomposition. With sufficient time, both will converge to the ground truth. But, the decomposition search would converge faster than just exploration.

6.2. Heterogeneous RL-based search & decomposition results

The figure on the next page shows the results of heterogeneous agents searching for targets with map decomposition, having 2 fast agents and 2 slow but accurate ones. With an RL-based strategy guiding the agents' trajectories, the map decomposition is sharper and faster. However, note that the central area with a higher probability of finding the targets given more attention by both the types of agents, especially the accurate sensor agents. This is because the reward there is so high compared to surrounding areas that it inhibits exploration of other areas to an extent and instead makes the agents focus on searching that area further. Also, since the accurate agent decreases uncertainty in the areas it samples by a large amount and is constantly focused on the higher probability area, its rewards and the trajectory they lead to seem to be dominating the end result. Still, it is a fact that dispersed faraway points with high probability of finding a target are

tough to be searched by a multi-agent system like this one and the model is doing well on the central part, so it is a positive result.

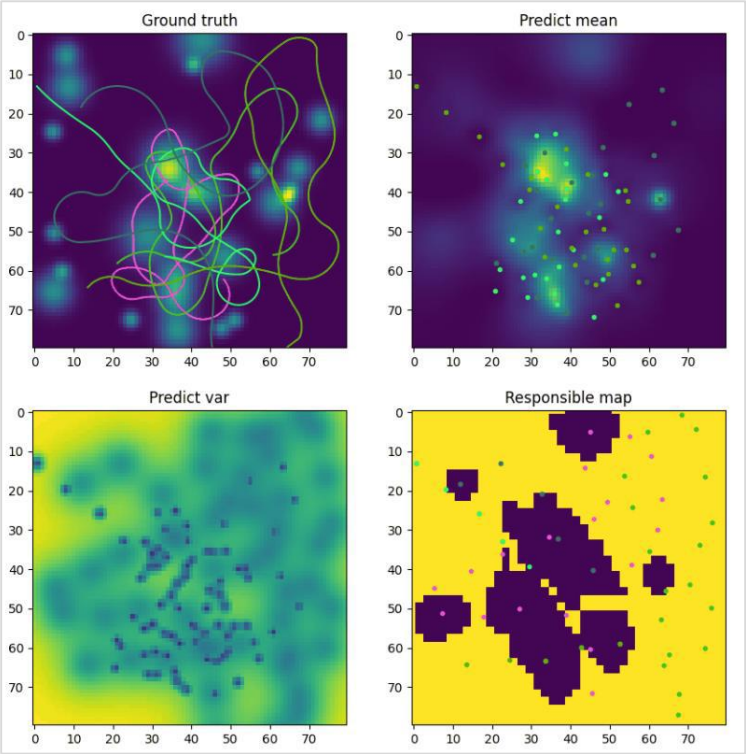


Fig. 6.3: *Heterogeneous Decomposition Results*

6.3. Heterogeneous agents task allocation

This section shows how the agents are allocating tasks on the map and the overall way the map will be searched by combining these tasks.

Different tasks have requirements for a specific agent. The areas covered by an agent while completing a task may overlap with those covered by another agent. Below we can see three agents undertaking three different tasks, and the areas covered by the three of them overlap to cover the whole map. The separate maps for each agent (red, green and blue) also show the trajectories each agent took while searching that area, superimposed on the ground truths for each task. The maps in the lower row are the predictions by each agent. The maps on the right are a combination of the 3 individual maps. Hence, the rightmost one in the top row denotes the ground truth while the one below it denotes the prediction.

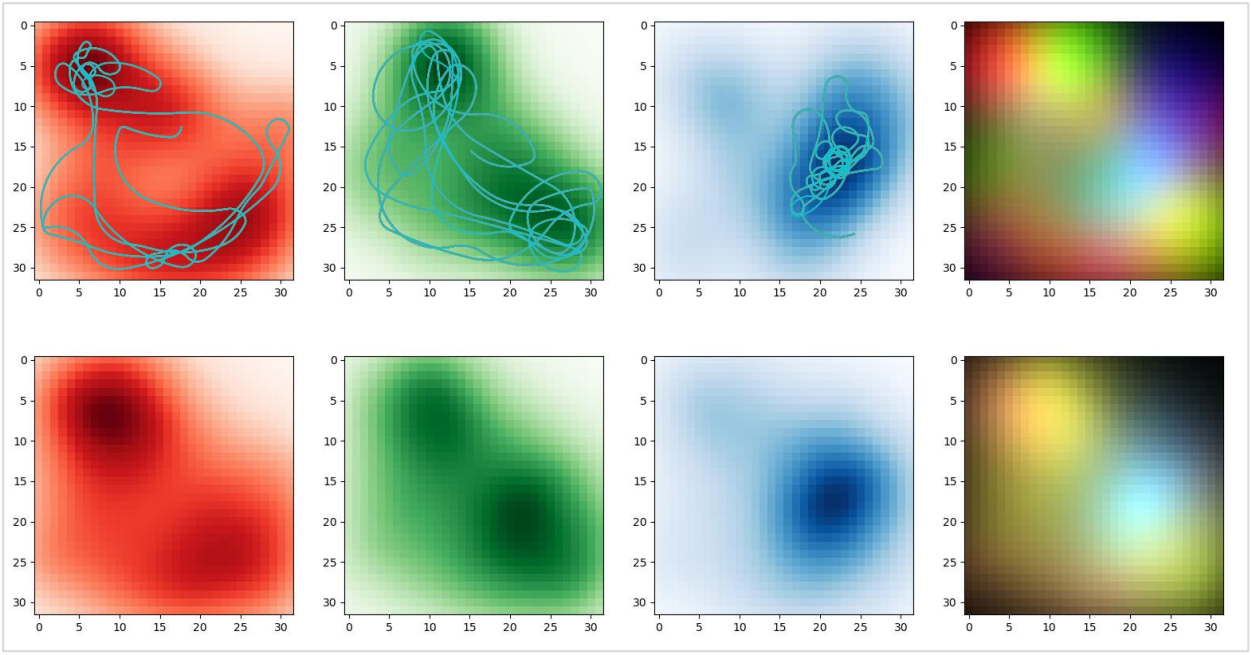


Fig. 6.4(a): *Task allocation - Trial 1*

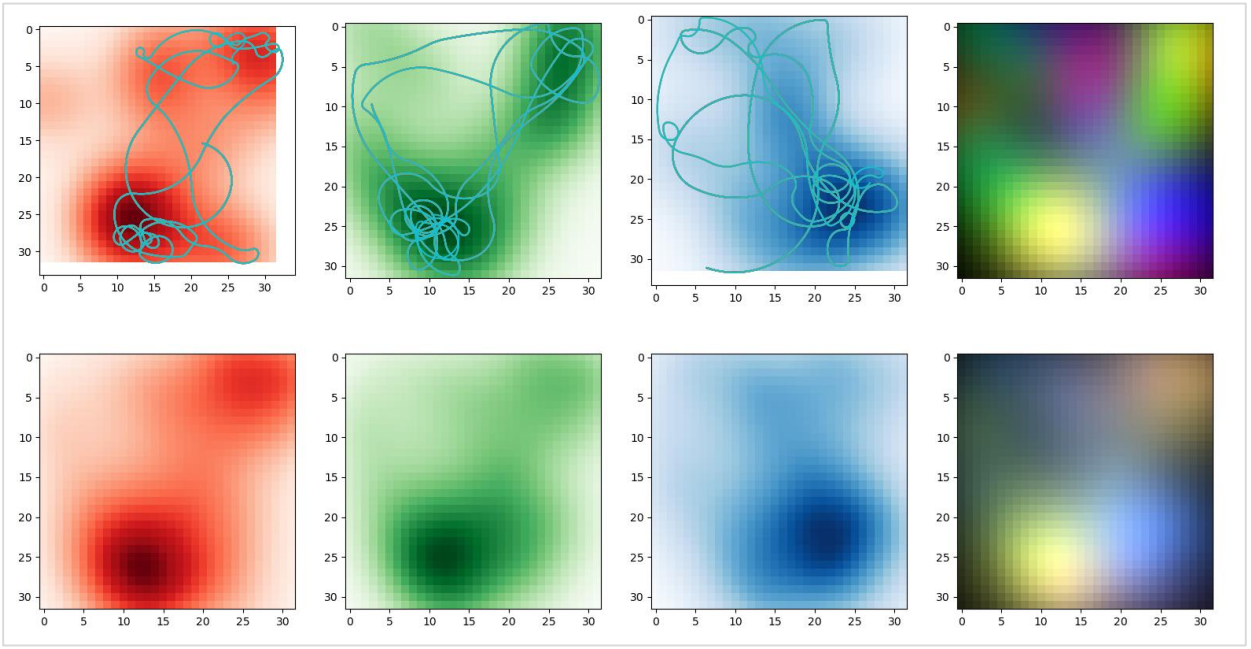


Fig. 6.4(b): *Task allocation - Trial 2*

Chapter 7

Conclusion & Scope for Future Work

This project was attempted with an objective of creating a method for using heterogeneous teams of agents to search an area for targets, and was part of a larger project at MARMot lab, NUS. With the guidance and help from my partner, a Ph. D student at the lab, the project was able to reach completion, and will be further explored and utilized in the larger parent project. A thorough literature review has been the bedrock for this project. Since Heterogeneous searching is still a nascent field compared to its Homogeneous counterpart, this project will benefit future researchers and students.

From an undergraduate student's point of view, this project helped to acquire knowledge regarding robotics, searching procedures and the field of reinforcement learning, at the same time giving a firsthand experience of the steps involved in the research methodology while getting acquainted with the variety of software utilized. Various Python modules, along with MATLAB and ROS to some extent have been used in this project.

Heterogeneous Multi-Agent Search is a very active research field, and is deployable in various scenarios, and we have already documented papers detailing its use in:

- Agriculture
- Search and rescue operations
- Searching in hazardous environments
- Military applications like detection of landmines
- Aerial scanning

Search and rescue (SAR) operations, for example, can take significant advantage from supporting or fully independent autonomous robots and multi-robot systems. These can aid in mapping and situational assessment, monitoring and surveillance, establishing communication networks, or searching for victims.

There are several major areas with a huge potential of future investigation in this project.

Starting with things specific to the project and some technicalities,

- GP may not be the best way to simulate accurate or inaccurate sensor measurements, because more results always improve the GP prediction. But in most scenarios in search and rescue, staying in the right place is more important. A different distribution in place of the Gaussian, or a totally different approach might help to improve this part.
- Agents tend to gather, which means further tuning of the cost function may be required. Data-driven method can also improve the performance when we want to decompose the area for more types of agents, where learning could make a big difference.

In a more general sense,

- The RL method used here is quite basic, so this is an area where significant improvement can be achieved.
- Teams of agents can be used, which opens the door to a host of possibilities based on agent coordination, task allocation, hierarchy of the agents and team structure.
- Coalition formation has not been explored.
- In a real world scenario, hardware considerations and communication comes into play. Here, we have assumed perfect communication among the robots and full availability of all information globally. If constraints in any of these fields are present, then the problem becomes more complicated.

References

[1] H. Ding, “Models & Algorithms for Multi-Agent Search Problems.” 2012. [Online]. Available: https://open.bu.edu/handle/2144/32075

[2] H. Choset, “Coverage for robotics - A survey of recent results,” *Ann. Math. Artif. Intell.*, pp. 1–14, 2001, [Online]. Available: papers3://publication/uuid/4FE047F1-929D-4BBD-8A61-2C6C697E186E

[3] A. Baranzadeh and A. V. Savkin, “A distributed control algorithm for area search by a multi-robot team,” *Robotica*, vol. 35, no. 6, pp. 1452–1472, 2017, doi: 10.1017/S0263574716000229.

[4] F. Bourgault, T. Furukawa, and H. F. Durrant-Whyte, “Optimal search for a lost target in a Bayesian world,” *Springer Tracts Adv. Robot.*, vol. 24, no. October 2003, pp. 209–222, 2006, doi: 10.1007/10991459_21.

[5] P. Lanillos, S. K. Gan, E. Besada-Portas, G. Pajares, and S. Sukkarieh, “Multi-UAV target search using decentralized gradient-based negotiation with expected observation,” *Inf. Sci. (Ny).*, vol. 282, pp. 92–110, 2014, doi: 10.1016/j.ins.2014.05.054.

[6] E. M. Wong, F. Bourgault, and T. Furukawa, “Multi-vehicle Bayesian search for multiple lost targets,” *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2005, no. April 2005, pp. 3169–3174, 2005, doi: 10.1109/ROBOT.2005.1570598.

[7] J. L. Baxter, E. K. Burke, J. M. Garibaldi, and M. Norman, “Multi-robot search and rescue: A potential field based approach,” *Stud. Comput. Intell.*, vol. 76, pp. 9–16, 2007, doi: 10.1007/978-3-540-73424-6_2.

[8] E. Ayvali, H. Salman, and H. Choset, “Ergodic coverage in constrained environments using stochastic trajectory optimization,” *IEEE Int. Conf. Intell. Robot. Syst.*, vol. 2017-Septe, pp. 5204–5210, 2017, doi: 10.1109/IROS.2017.8206410.

[9] Y. Rizk, M. Awad, and E. W. Tunstel, “Cooperative heterogeneous multi-robot systems: A survey,” *ACM Comput. Surv.*, vol. 52, no. 2, 2019, doi: 10.1145/3303848.

[10] S. Gronauer and K. Diepold, *Multi-agent deep reinforcement learning: a survey*, no. 0123456789. Springer Netherlands, 2021. doi: 10.1007/s10462-021-09996-w.

[11] and J. R. W. S. J. Benkoski, M. G. Monticino, “A survey of the search theory literature,” *Nav. Res. Logist.*, vol. 38, pp. 469–494, 1991.

[12] R. W. Beard and T. W. McLain, “Multiple UAV Cooperative Search under Collision Avoidance and Limited Range Communication Constraints,” *Proc. IEEE Conf. Decis. Control*, vol. 1, no. December, pp. 25–30, 2003, doi: 10.1109/cdc.2003.1272530.

[13] M. Flint, E. Fernández-Gaucherand, and M. Polycarpou, “Cooperative Control for UAV’s Searching Risky Environments for Targets,” *Proc. IEEE Conf. Decis. Control*, vol. 4, no. December, pp. 3567–3572, 2003, doi: 10.1109/cdc.2003.1271701.

[14] and K. M. P. M. M. Polycarpou, Y. Yang, “A cooperative search framework for distributed agents,” 2001, [Online]. Available: https://www2.ece.ohio-state.edu/~passino/ISIC01s.pdf

[15] G. Sharon, R. Stern, A. Felner, and N. Sturtevant, “Conflict-based search for optimal

multi-agent path finding," *Proc. 5th Annu. Symp. Comb. Search, SoCS 2012*, pp. 97–104, 2012.

[16] J. Hu, L. Xie, K. Y. Lum, and J. Xu, "Multiagent information fusion and cooperative control in target search," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 4, pp. 1223–1235, 2013, doi: 10.1109/TCST.2012.2198650.

[17] M. S. Vitaly Ablavsky, "Optimal Search for a Moving Target : A Geometric Approach," no. August, 2000.

[18] M. Dadgar, S. Jafari, and A. Hamzeh, "A PSO-based multi-robot cooperation method for target searching in unknown environments," *Neurocomputing*, vol. 177, no. 1, pp. 62–74, 2016, doi: 10.1016/j.neucom.2015.11.007.

[19] T. H. Chung and J. W. Burdick, "Multi-agent probabilistic search in a sequential decision-theoretic framework," *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 146–151, 2008, doi: 10.1109/ROBOT.2008.4543200.

[20] T. Furukawa, F. Bourgault, B. Lavis, and H. F. Durrant-Whyte, "Recursive Bayesian search-and-tracking using coordinated UAVs for lost targets," *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2006, no. May, pp. 2521–2526, 2006, doi: 10.1109/ROBOT.2006.1642081.

[21] M. Davoodi, S. Faryadi, and J. M. Velni, "A Graph Theoretic-Based Approach for Deploying Heterogeneous Multi-agent Systems with Application in Precision Agriculture," *J. Intell. Robot. Syst. Theory Appl.*, vol. 101, no. 1, 2021, doi: 10.1007/s10846-020-01263-4.

[22] P. Dasgupta, A. Muñoz-Meléndez, and K. R. Guruprasad, "Multi-robot terrain coverage and task allocation for autonomous detection of landmines," *Sensors, Command. Control. Commun. Intell. Technol. Homel. Secur. Homel. Def. XI*, vol. 8359, pp. 83590H-83590H–14, 2012, doi: 10.1117/12.919461.

[23] I. Abraham and T. D. Murphey, "Decentralized Ergodic Control: Distribution-Driven Sensing and Exploration for Multiagent Systems," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 2987–2994, 2018, doi: 10.1109/LRA.2018.2849588.

[24] D. Saldana, R. Assuncao, and M. F. M. Campos, "A distributed multi-robot approach for the detection and tracking of multiple dynamic anomalies," *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2015-June, no. June, pp. 1262–1267, 2015, doi: 10.1109/ICRA.2015.7139353.

[25] G. Sartoretti *et al.*, "PRIMAL: Pathfinding via Reinforcement and Imitation Multi-Agent Learning," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2378–2385, 2019, doi: 10.1109/LRA.2019.2903261.

[26] S. Ivic, "Motion Control for Autonomous Heterogeneous Multiagent Area Search in Uncertain Conditions," *IEEE Trans. Cybern.*, pp. 1–13, 2020, doi: 10.1109/tcyb.2020.3022952.

[27] G. Sartoretti, A. Rao, and H. Choset, "Spectral-Based Distributed Ergodic Coverage for Heterogeneous Multi-agent Search," *Springer Proc. Adv. Robot.*, vol. 22 SPAR, pp. 227–241, 2022, doi: 10.1007/978-3-030-92790-5_18.

[28] J. P. Queralta *et al.*, "Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision," *IEEE Access*, vol. 8, pp. 191617–191643, 2020, doi:

10.1109/ACCESS.2020.3030190.

[29] C. Wakilpoor, P. J. Martin, C. Rebhuhn, and A. Vu, “Heterogeneous Multi-Agent Reinforcement Learning for Unknown Environment Mapping,” 2020, [Online]. Available: http://arxiv.org/abs/2010.02663

[30] I.-J. Liu, U. Jain, R. A. Yeh, and A. G. Schwing, “Cooperative Exploration for Multi-Agent Deep Reinforcement Learning,” 2021, [Online]. Available: http://arxiv.org/abs/2107.11444

[31] R. Reijnen, Y. Zhang, W. Nuijten, C. Senaras, and M. Goldak-Altgassen, “Combining Deep Reinforcement Learning with Search Heuristics for Solving Multi-Agent Path Finding in Segment-based Layouts,” *2020 IEEE Symp. Ser. Comput. Intell. SSCI 2020*, pp. 2647–2654, 2020, doi: 10.1109/SSCI47803.2020.9308584.

[32] Maxim Egorov, “Multi-Agent Deep Reinforcement Learning,” 2016, [Online]. Available: http://cs231n.stanford.edu/reports/2016/pdfs/122_Report.pdf

[33] A. Mavrommati, E. Tzorakoleftherakis, I. Abraham, and T. D. Murphey, “Real-Time Area Coverage and Target Localization Using Receding-Horizon Ergodic Exploration,” *IEEE Trans. Robot.*, vol. 34, no. 1, pp. 62–80, 2018, doi: 10.1109/TRO.2017.2766265.

[34] O. D. Jochen Görtler, Rebecca Kehlbeck, “A Visual Exploration of Gaussian Processes,” 2019, [Online]. Available: https://doi.org/10.23915/distill.00017