Multi-agent Multi-objective Ergodic Search

Akshaya Kesarimangalam Srinivasan

CMU-RI-TR-23-55

July 14, 2023

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Dr. Howie Choset, *chair*
Dr. Maxim Likhachev
Dr. Katia Sycara
Dr. Zhongqiang Ren

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Robotics.

To my family.

Abstract

In order to find points of interest in a given domain, many planners use *a priori* information to guide the search to expedite the detection of targets. Information-based search techniques use an *a priori* information map, typically represented as a probability distribution, to guide their search. An information map can describe, for example, the likelihood of finding survivors at a location in search and rescue applications. We present an approach to direct multiple agents (MA) to search a given domain subject to multiple objectives (MO), each characterized by its own information map. Our approach embraces an information-based method, called *ergodic search* (ES). ES utilizes an ergodic metric, defined later in this work, that is minimized when the time spent by an agent in a region is proportional to the amount of information contained in that region. In this thesis, we introduce the Multi-Agent Multi-Objective Ergodic Search (MA-MO-ES) problem and prescribe computationally efficient methods to solve it.

The primary goal of this work is to determine the trajectories for all agents such that the worst-case objective or highest ergodic metric on any information map is minimized (minmax objective). Naively computing a joint trajectory of all the agents by optimizing the ergodic metric on the average of all the information maps results in a high ergodic metric with respect to each map. This is because spending time in the high-information region on one map can correspond to spending time in the low-information regions of another map. If each agent's trajectory is instead computed by considering a subset of maps, our results show that the maximum ergodic metric on the information maps can be considerably reduced. This requires determining which information maps should be considered when optimizing the trajectory of a particular agent. In other words, this work computes the optimal allocation, of information maps to agents, that minimizes the maximum ergodic metric on the given information maps.

The main challenge in determining the optimal allocation is the exponential growth of the number of possible allocations with the number of maps and agents. Further, computing the cost of an allocation is itself an expensive planning problem. This is because evaluating an allocation requires identifying the maximum ergodic metric with respect to the information maps by computing the trajectory for each agent that optimizes the ergodic metric on its assigned maps. The expensive evaluation of one allocation coupled with the exponentially growing number of possible allocations makes solving for an optimal allocation using brute force computationally intractable.

To mitigate the computational challenge of exponential growth, we present a branch and bound-based algorithm with pruning techniques that reduce the number of allocations to be searched to find the optimal allocation. To reduce the branching factor in branch and bound, we propose two approaches for clustering information maps before allocation: k-means and minimum bounding sphere clustering. This sacrifices guaranteed optimality in exchange for improved computational performance. These clustering approaches leverage the similarity between information maps to approximate the cluster of maps that should be assigned to a single agent to achieve minmax ergodic metric on the information maps. Clustering information maps before allocation decreases the number of possible allocations and thus the branching factor in the branch and bound, further reducing the problem's computational complexity. Testing on 70 randomly generated test cases shows an order of magnitude improvement in runtime for our branch and bound approach compared to an exhaustive brute force approach. Using similarity clustering, the runtime further reduces by two orders of magnitude even for tests with ten information maps and four agents while maintaining good quality allocations with an average 20% deviation from the optimal minmax ergodic metric.

Acknowledgments

I would like to thank my advisor Professor Howie Choset for his continued support and guidance throughout this work. I would also like to thank Bhaskar Vundurthy, Geordan Gutow, and Richard Ren for their valuable contributions and feedback at every step of the way. I would also like to thank my committee: Professor Maxim Likhachev and Katia Sycara for their insights and direction on this work.

Endless love to all my friends at RI for making this journey more fun and easier. Finally, I owe many thanks to my friends and family for listening to my endless rants, ranting with me, and providing all the encouragement to help me keep moving forward.

Contents

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

List of Tables

Chapter 1

Introduction

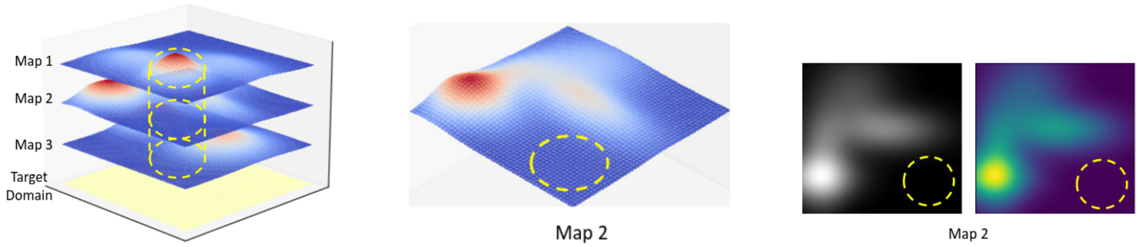
1.1 Motivation

Applications like search and rescue require effective planning for the robot to find survivors or potential hazards over a large area after a huge disaster [6][34][44][45]. With some *a priori* information over the domain, the robot can adapt its path such that more investigative effort is given to the high-information parts of the domain to increase the likelihood of detecting targets. We chose to represent this *a priori* information as an information map. For a domain, $\mathcal{W} = [0, L_1] \times [0, L_2] \times \cdots \times [0, L_\nu]$, an information map is a probability distribution represented as $f : \mathcal{W} \to \mathbb{R}$ such that $\int_{x \in \mathcal{W}} f(x)dx = 1$ and $f(x) > 0$ where, $f(x)$ can represent for instance the probability of finding a survivor, hazardous objects, or important artifacts in different parts of the search domain.

The previous literature has produced different approaches for uniformly covering a domain; some approaches include geometric (lawnmower patterns [4][1]), trajectory optimization [12] and decomposition-based [14] approaches. However, with some *a priori* knowledge of the domain, like the information map, we can leverage more intelligent approaches that improve upon coverage in terms of metrics like information gained. Ergodic search [29] has demonstrated its effectiveness in exploring a domain while striking a balance between exploitation (i.e., myopically searching high-information areas) and exploration (i.e., attempting to visit all possible locations for new information). The ergodic search uses the ergodic metric [29] that is minimized when the time spent observing a region is proportional to the information contained in that region. A lower ergodic metric implies a better ergodic coverage of the domain.

Ergodic search outperforms the non-information theoretic method random walks [33], as well as information-theoretic approaches such as information gradient ascent [33], information maximization [32][43], and greedy expected entropy reduction [33]. Moreover, ergodic search methods have shown robustness to measurement distractors [33], information modeling errors [26], and sensor noise [16]. Therefore, in this work, we utilize ergodic search to determine the trajectories of all the agents in the domain.

In many cases, there may be multiple types of information to collect which can have different distributions across the domain as illustrated in Figure 1.1 where three information maps are shown over the same domain. Then, it is natural to cast the search problem as a multi-objective optimization problem where each objective is characterized by its own information map. In this work, we consider this multi-objective problem. Figure 1.1a further highlights a region of high information on Map 1 that corresponds to low information on Maps 2 and 3. The objectives represented by these information maps are thus conflicting.



(a) (b) (c)

Figure 1.1: **Information maps illustration:** (a) Three information maps, Map 1, 2, and 3, represented as probability distributions over the same domain. The region highlighted by the yellow circle illustrates a region of high information in Map 1 that corresponds to a region of low information in Maps 2 and 3. This shows that the objectives characterized by these information maps are conflicting. (b) 3D visualization of Map 2, and (c) 2D visualization of Map 2 with 'coolwarm' and 'gray' color schemes that are used in the rest of this work.

For a multi-objective problem, no single solution exists that simultaneously optimizes each objective. In that case, the objectives are said to be conflicting as in Figure 1.1a. A solution to a multi-objective problem is called non-dominated or

Pareto-Optimal, usually not unique, if none of the objectives can be improved in value without degrading some of the other objective values. Our prior work [38] found a set of Pareto-Optimal solutions for a single agent tasked to search a domain subject to multiple objectives, each represented by an information map. In this thesis, we are interested in reconciling the multiple information maps by minimizing the maximum of the objectives, i.e., the maximum ergodic metric on the information maps.

When a single agent is tasked to search the domain subject to multiple information maps if we compute the trajectory of the agent by optimizing the ergodic metric on a linear combination of all the information maps, this results in a high ergodic metric with respect to each map. On any particular map, only a small fraction of the agent's time is spent on the map's high-information region and the majority of the time would be spent on the low-information regions because it corresponds to the high-information regions on other maps as highlighted in Figure 1.1. When multiple agents are available and a similar strategy is used for each agent, it will again result in a high ergodic metric on each map as supported by our results in Section 4.2. However, with multiple agents, a better strategy can be used to reduce the ergodic metric on the maps.

Each agent can focus on one map or a subset of maps and only access information from the maps assigned to them. In this case, agents can spend a larger fraction of their time on high-information regions compared to low-information regions resulting in a lower ergodic metric on any map. Thus, in a multi-agent setting, it is beneficial to allocate information maps to agents in order to minimize the maximum ergodic metric on the maps. We thereby implement a task allocation scheme aimed to allocate one or more information maps to each agent such that the maximum ergodic metric on any map is minimized, i.e., minmax optimization. We approach this problem as a Multi-Agent Multi-Objective Ergodic Search (MA-MO-ES) problem as shown in Figure 1.2.

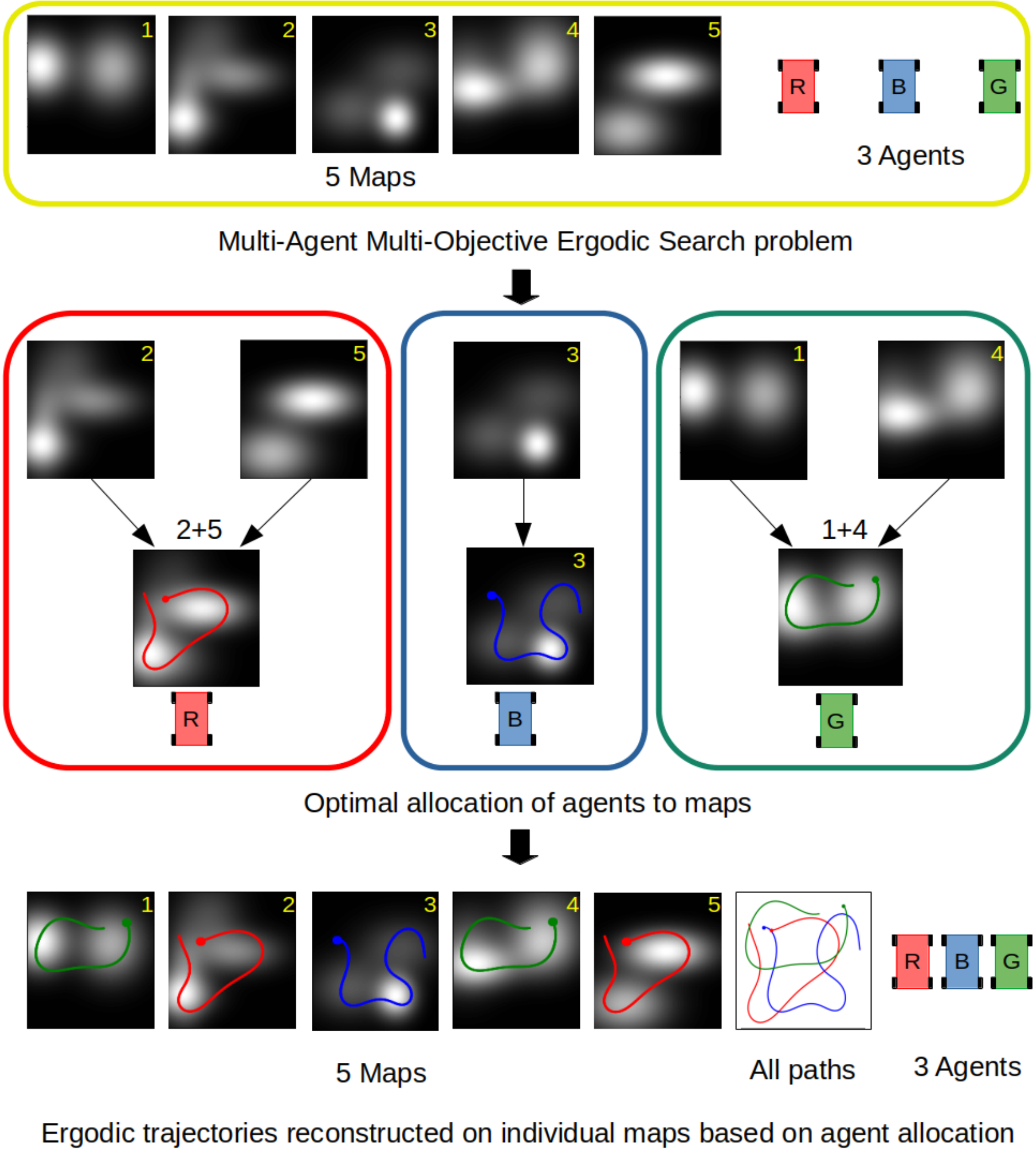

1
2
3
4
5
R
B
G
5 Maps
3 Agents
Multi-Agent Multi-Objective Ergodic Search problem
2
5
2+5
R
3
3
B
1
4
1+4
G
Optimal allocation of agents to maps
1
2
3
4
5
R
B
G
5 Maps
All paths
3 Agents
Ergodic trajectories reconstructed on individual maps based on agent allocation

Figure 1.2: **Example problem and solution overview:** This figure illustrates the Multi-agent Multi-objective Ergodic Search (MA-MO-ES) problem using five information maps that span the same physical region $(1-5)$, and three agents (as for Red, Green, and Blue). The optimal allocation is shown with each agent in a different box. An agent's trajectory is computed by optimizing the ergodic metric on a scalarization of its assigned maps and is evaluated against its assigned maps.

1.2 Challenges

According to the taxonomy defined in [21], our problem is a multi-task single-robot time-extended variation of the Multi-Robot Task Allocation (MRTA) problem that is NP-Hard [25]. Prior works have addressed multi-agent multi-objective task allocation problems using auctioning systems [9], greedy allocation [7] etc. While the approaches in [7],[9] are fast, they do not guarantee finding the optimal allocation scheme.

The brute force approach of exhaustively searching through the allocation space to find the optimal allocation has exponential time complexity. This is because the number of possible allocations of information maps to agents increases exponentially with the number of maps and agents. For M information maps and N agents with $M \geq N$, the number of possible allocations can be computed using the formula $S(M, N) = \sum_{i=0}^{N} \frac{(-1)^{N-i} i^M}{(N-i)!i!}$ where S is the Stirling number of the second kind. Further, to evaluate each allocation, we need to identify the maximum ergodic metric with respect to the information maps. This requires computing the trajectory for each agent that optimizes the ergodic metric on its assigned maps. The expensive evaluation of one allocation coupled with the exponentially growing number of possible allocations makes solving for the optimal allocation using brute force computationally intractable.

1.3 Contributions

In this thesis, we first address computing the information map to optimize the trajectory of an agent that is assigned multiple information maps. For a single information map, we can directly compute the locally optimal ergodic trajectory. However, when a single agent is assigned more than one information map as in Single Agent Multi-Objective Ergodic Search (SA-MO-ES), the optimal trajectory could be defined as the one that minimizes the sum of the ergodic metric (min-sum), the maximum ergodic metric (minmax) or the minimum ergodic metric (min-min) on the maps. In this work, for SA-MO-ES, we define the optimal trajectory as minimizing the maximum ergodicity on individual maps (worst-case objective). We present an approach to compute the information map for optimizing the agent's trajectory to achieve the minmax ergodic metric on individual maps, without computing the entire Pareto-Optimal set of weight vectors as done in [38].

Next, we cater to the multi-agent multi-objective ergodic search (MA-MO-ES) problem, aimed at finding the optimal allocation of information maps to agents to achieve minmax ergodic metric on the information maps. We leverage the minmax formulation to identify specific non-optimal allocations without evaluating their performance on all the maps. Consider that the highest ergodicity on maps assigned to a subset of agents exceeds that of the current best allocation. This implies that the highest ergodicity across all maps would also exceed that of the current best allocation. Using this idea, we present a branch and bound formulation for allocating maps to agents. This approach's runtime and minmax metric were compared against baseline approaches (joint trajectory optimization, exhaustive search, and greedy allocation). The joint trajectory optimization and greedy allocation approach though fast do not guarantee optimal allocation. Compared to exhaustive search, the branch and bound approach reduces runtime by an order of magnitude while guaranteeing minmax optimal allocation.

Further, we also propose two approaches to leverage similarity between information maps to reduce the branching factor in branch and bound. This sacrifices guaranteed optimality in exchange for improved computational performance. In both approaches, the information maps are represented as vectors of weighted Fourier coefficients. The first approach employs k-means clustering, while the second approach uses the idea of minimum bounding spheres to cluster the information maps. Once the clusters are identified, they are assigned to the agents using a similar branch and bound formulation. Since the number of clusters is less than the number of information maps, the number of possible allocations of clusters to agents is less than that of maps to agents. This reduces the branching factor in the branch and bound, further reducing the problem's computational complexity. The performance of these approaches was compared against the branch and bound without clustering and k-means clustering with a distance-based allocation approach on 70 randomly generated test cases. Our experiments demonstrate that combining clustering with branch and bound further reduces the runtime by two orders of magnitude while producing good quality allocations with minmax ergodicity within 20% of the optimal.

Chapter 2

Background and Related Works

2.1 Coverage Path Planning Methods

Coverage Path Planning (CPP) describes the process of generating robot trajectories that pass over all points of an area or volume of interest[18]. CPP algorithms can be classified as heuristic or complete depending on whether they can guarantee the coverage of the entire free space. It can also be classified as online or offline where the latter relies on stationary information and the environment is assumed to be known in advance [13].

Coverage planning can have various requirements such as finding the shortest path, avoiding collisions, trading maximal coverage with time budgets, or just generating attractive and intuitive movement patterns [8]. Geometric approaches like lawnmower patterns are primarily used for uniform coverage of a domain.

When the area of interest contains a non-uniform information distribution, for example, a probability distribution representing the likelihood of finding a target at different parts of the domain, other more intelligent coverage methods can be used. Grid-based approaches like wavefront propagation [49][42], and potential field [22] divide the environment into a grid, and each cell is assigned a coverage value. Optimization-based approaches formulate information-based CPP as an optimization problem and consider objectives such as minimizing path length, maximizing coverage, or balancing energy or time consumption. Ergodic coverage, a type of information-based coverage is explained further in the next section.

We note that since in many realistic scenarios, the information distribution is unknown or uncertain, other intelligent methods like adaptive sampling with an information-theoretic metric can be used to first estimate this unknown distribution [27]. This can also be done in a distributed manner and with heterogeneous agents in the context of multi-robot sensor coverage [28] [41].

Finally, coverage planning with multiple agents has been addressed using centralized and distributed approaches and different ideas like identifying critical points, region-division, and reinforcement learning in [19][35][47].

2.2 Ergodic Coverage

2.2.1 Introduction

From the prior section, ergodic coverage is an offline approach to the information-based coverage problem. Given an information map or information distribution over the search domain, a trajectory is ergodic when it visits every subset of the domain with a probability equal to the measure of that subset[29], i.e., when the time spent in an area is proportional to the information present in that area. Figure 2.1 shows an example of an ergodic and non-ergodic trajectory on an information map.

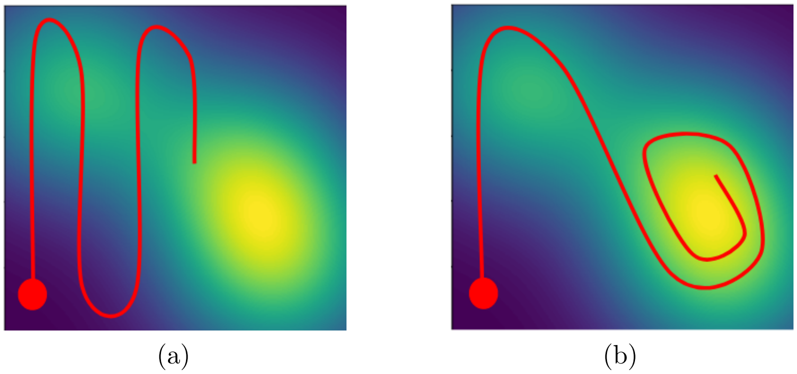


Figure 2.1: **Ergodic Trajectory:** The figure represents a single information map with (a) a non-ergodic trajectory and (b) an ergodic trajectory. Here, the yellow regions represent areas of high information.

Ergodic trajectory planning has been studied in various contexts such as receding horizon control [31], stochastic optimization [5], active learning and search [3][33], decentralized exploration [2], and real-time area coverage and target localization [30]. Here, we will introduce the mathematical preliminaries required to compute the locally optimal ergodic trajectory given an information map and agent dynamics.

2.2.2 Mathematical Preliminaries

Let $\mathcal{W} = [0, L_1] \times [0, L_2] \times \cdots \times [0, L_\nu] \subset \mathbb{R}^\nu, \nu \in \{2, 3\}$ denote a ν-dimensional workspace that is to be explored by the robot. The robot has an n-dimensional state space $S = \mathcal{W} \times V$ $(n \geq \nu)$. V is comprised of the robot state components, such as velocities or orientations, that do not affect what the sensor can see. Let $q : [0, T] \rightarrow S$ denote a trajectory in the state space with $T \in \mathbb{R}^+$ representing the time horizon. The robot has deterministic dynamics given by $\dot{q}(t) = f(q(t), u(t))$, where $u(t)$ is the control input of the robot. Let $P : S \rightarrow \mathcal{W}$ project the state space into the workspace.

Let $c(x, q), x \in \mathcal{W}$ denote the time-averaged statistics of a trajectory q, which is defined as:

$$c(x, q) = \frac{1}{T} \int_0^T \delta(x - P(q(\tau))) d\tau, \tag{2.1}$$

where δ is a Dirac function.

Let $\phi : \mathcal{W} \rightarrow \mathbb{R}$ denote a static information map that describes the amount of information at each location in the workspace. Each information map is a probability distribution with $\int_\mathcal{W} \phi = 1$ and $\phi(x) \geq 0, \forall x \in \mathcal{W}$.

An ergodic metric [29] between $c(x, q)$ and an information map ϕ is defined Equation 2.2:

$$\begin{aligned} \mathcal{E}(\phi, q) &= \sum_{k \in \mathcal{K}} \lambda_k (c_k - \phi_k)^2 \\ &= \sum_{k \in \mathcal{K}} \lambda_k \left(\frac{1}{T} \int_0^T F_k(q(\tau)) d\tau - \phi_k \right)^2 \end{aligned} \tag{2.2}$$

where,

(i) $k \in \mathcal{K}$ is a set of ν coefficient indices $\{k_1, k_2, \cdots, k_\nu\}$ with $k_i \in \mathcal{N}$ so that

$\mathcal{K} = \{k \in \mathcal{N}^{\nu} : 0 \leq k_i \leq K\}$ and K being the number of Fourier bases considered, (ii) $\phi_k = \int_{\mathcal{W}} \phi(x) F_k(x) dx$ represents the Fourier coefficients of the information map, with $F_k(q) = \frac{1}{h_k} \Pi_{j=1}^{\nu} \cos(\frac{k_j \pi q_j}{L_j})$ being the cosine basis function for some index $k \in \mathbb{N}^{\nu}$, (iii) c_k denotes the k^{th} Fourier coefficient of $c(x, q)$,
(iv) h_k denotes the normalization factor and for 2D domain, it is defined as: $h_k = \sqrt{\int_0^{L1} \int_0^{L2} \cos^2(p_1 x_1) \cos^2(p_2 x_2) dx_1 dx_2}$ and $p_1 = \frac{K\pi}{L_1}$, $p_2 = \frac{K\pi}{L_2}$ [29], and
(v) $\lambda_k = (1 + ||k||^2)^{-\frac{\nu+1}{2}}$ denotes the weight for each corresponding Fourier coefficient. This weighs large-scale (or low frequency) modes more than small-scale modes so that as $K \to \infty$, the series converge.

Thus, a locally optimal ergodic trajectory on the information map ϕ is one that minimizes the ergodic metric and can be defined as in Equation 2.3.

$$q^* = \underset{\forall q}{\operatorname{argmin}} \, \mathcal{E}(\phi, q) \tag{2.3}$$

2.3 Multi-agent Multi-objective Task Allocation and Path Planning

Various approaches have been studied for multi-agent task allocation and path planning problems [36][37][39][46]. The work in [6] addresses the multi-target multi-agent discrete search and rescue path planning problem. It proposes a mixed integer linear problem (MILP) to maximize the weighted (based on priority) cumulative probability of detecting heterogeneous targets whose locations are unknown apriori. A prior target location probability density distribution is considered domain knowledge similar to the information maps in our problem. However, this approach does not have an explicit task allocation, and the paths are computed for a single team of ‘n’ agents.

The core contribution of this thesis is the computationally efficient method for computing optimal task allocation for multiple agents with multiple objectives. Multi-agent task allocation is widely studied in literature [20]. In multi-robot exploration that uses a frontier-based approach, robots need to be assigned to the identified frontiers (free spaces on the boundary between explored and unexplored regions of the environment) in the environment. While exploring the environment different objectives can be considered like maximizing information gain and minimizing distance traveled by the robot. The allocation can then be performed using market-based strategies [50], high-level task allocation [11], or simply a greedy allocation [10]. The authors of [10] also used a frontier-based exploration but with the addition of hard and soft constraints like inter-robot distances, heading bias, and logical areas of influence for exploration (like the regions of high information in our problem). However, [10] uses a greedy allocation scheme. Authors of [50] and [11] employ a more informed allocation. However, in all these works, there is only one type of information to be gained, the different objectives are not different types of information across the domain and each task is a frontier rather than a coverage problem.

One of the main challenges in the multi-agent multi-objective problem as stated in the introduction is the high cost of evaluating each possible allocation's performance, which is unknown in advance. Work in [24] addresses target retrieval from a cluttered environment using multiple manipulators. The number of object/clutter re-arrangements needed for target retrieval is unknown beforehand, similar to the unknown allocation cost in our problem. This work estimates the number of object re-arrangements required using a heuristic based on the maximum and minimum grasping angles of all the feasible paths to the target in the environment. It then uses a utility function proportional to the inverse of this value for task allocation. Further, this utility is also used to determine the best task decomposition (as different object re-arrangements can be used to retrieve the same object). However, it only used a single heuristic objective function for task allocation and path planning.

Two prior works related to our approach are [9] and [7]. The authors of [7] propose two frameworks for task allocation, the Compromise View model and the Nearest Neighbour Search model. The former uses the agent's distance from the target as the heuristic for greedy task assignments whereas the latter clusters target locations using k-means clustering for faster but higher path-length solutions. The work in [9] presents a framework for multiple agents to perform exploration, rendezvous, and tasks in a cell-decomposed environment. The cells to be visited are clustered, one for each agent, and assigned via centralized auction based on agents' distance to the cluster centroids and a multi-objective optimization method based on weighted prioritization of exploration and task completion. Both works demonstrate the effectiveness of clustering for task allocation in multi-agent scenarios. Our work differs from [9] and [7] in two ways. First, computing the cost of allocation is itself a trajectory optimization problem and hence expensive to obtain accurately. Second, the tasks are not waypoints that need to be visited but probability distributions that need to be covered. Thus, there aren't explicit targets whose distance from the agents can be measured and used to compute the cost/value of an allocation. In our problem, there is a lack of a good estimate of the cost of allocating a task (coverage task on an information map) to an agent. Our problem can be slightly modified in order to fit this framework and is explained further in Section 5.3.

Chapter 3

Single Agent Multi-Objective Ergodic Search (SA-MO-ES)

3.1 A Pareto-Optimal Local Optimization Framework for Multi-Objective Ergodic Search

[1] When a single agent is tasked to cover a domain subject to one information map, then the locally optimal ergodic trajectory on the map can be directly computed using equation (2.3). However, when an agent is tasked to cover a domain subject to more than one information map (SA-MO-ES), the problem is truly multi-objective in the sense that in general, there is no single trajectory that optimizes the ergodic metrics with regard to all information maps at the same time.

Our prior collaborative work [38] instead aims to find the so-called Pareto-Optimal solutions for this problem, which offers human decision-makers a range of solutions offering a different trade-off in the level of coverage on the individual maps. It generates a Pareto-Optimal front where each element is a weight vector used to generate a linear combination of the information maps (scalarized information map). The trajectory optimized on each of these scalarized information maps is a Pareto-

[1] This work is a prior collaborative work with Dr. Zhongqiang Ren in [38]

Optimal trajectory. A solution is Pareto-Optimal if one can not improve the ergodic metric with respect to one information map without deteriorating the ergodic metric with regard to at least one of the other information maps.

An existing brute-force scalarization method [15], can be applied to solve SA-MO-ES by sampling a set of weight vectors, computing the weighted sum of the information maps for each weight vector, and running an ergodic trajectory optimization on the scalarized information map. This is time-consuming to search through all the weight vectors.

To efficiently solve SA-MO-ES, [38] develops a framework called Sequential Local Ergodic Search (SL-ES). SL-ES combines the recent advances in ergodic search techniques with the idea of local optimization based on the inherent convexity of the ergodic metric in the Fourier coefficient space to efficiently compute the Pareto-Optimal solutions. SL-ES samples weight vectors from the weight space (i.e., the space that contains all possible weight vectors) in a breadth-first manner by (i) episodically sampling new weight vectors in the neighborhood of the current weight vector, and (ii) optimizing the trajectory corresponding to the new weight vector by using the current solution as the initial guess (to warm-start the optimization). This reduces the time taken to compute the Pareto-Optimal solutions.

To expedite the computation, [38] also develops a variant called Adaptive SL-ES (A-SL-ES), which can adjust the density of the sampled weight vectors based on the similarity of the information maps in the Fourier coefficient space. It transforms the weight vectors' space based on the similarity between the maps. If the information maps are similar then close to equal weights for the maps can be used to obtain a Pareto-Optimal trajectory. Experimental results showed that both SL-ES and A-SL-ES required less than half of the run time of a naive scalarization method that does not leverage local optimization.

An example of the generated Pareto-Optimal front for one agent tasked to cover a domain subject to two information maps is shown in 3.1. Each point in this front corresponds to a weight vector used to compute the scalarized map. The choice of a solution from the front is left up to the user depending on their preference/priority among the different objectives. We note that in our problem, we are concerned with the minmax objective. Thus among the weight vectors used to create the Pareto-Optimal front, the one highlighted with a square box would be chosen. However, the

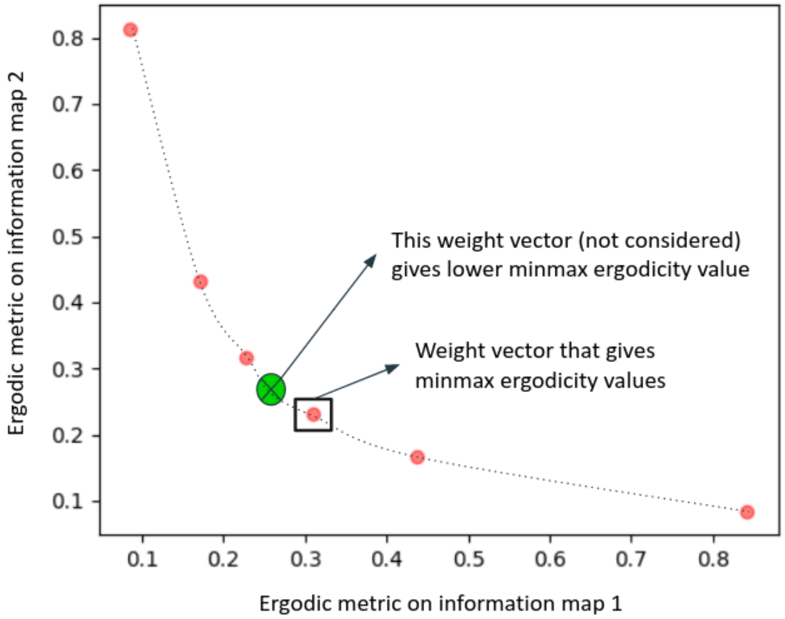


Figure 3.1: **Example Pareto-Optimal front:** Pareto-Optimal front for two information maps. Each point corresponds to a 2D weight vector that is used to compute a scalarized information map. The trajectory optimized on the scalarized information map is used to compute the ergodic metric on information maps 1 and 2. The squared point corresponds to the choice of weight vector as per minmax optimality in the Pareto-Optimal front among the weight vectors sampled. However, the green × represents another weight vector, that was not sampled, with a lower minmax metric on the individual ergodicities.

weight vector highlighted by the green × would provide a better minmax metric on the individual ergodicities.

3.2 Choosing from Pareto-Optimal solutions: TOPSIS

To pick a solution from the generated Pareto-Optimal front, we can pick the weight vector that results in the minimum maximum ergodic metric among the ones tested as shown in Figure 3.1. We can also employ other methods like TOPSIS [23] [48], one of the fundamental methods in multi-attribute decision-making (MADM) problems, to pick a solution from the Pareto-Optimal front. TOPSIS makes a decision by looking at the similarity to an ideal solution. It chooses a solution that is closest to the ideal and farthest away from the worst-case solution. In our problem, the ideal solution is the minimum ergodic metric achieved on each map across all the solutions. The worst-case solution would be the maximum ergodic metric achieved on each map across all the solutions. TOPSIS thus requires the entire Pareto front to be computed for each possible allocation, it only picks a solution from the weight vectors used to compute the Pareto-Optimal front, and it does not explicitly minimize the highest ergodic metric on the information maps.

3.3 Minimum Bounding Sphere (MBS) Scalarization

In this work, we propose an approach for SA-MO-ES that focuses on computing the optimal scalarized information map directly. This is done without the need to calculate the entire Pareto-Optimal front, and the goal is to minimize the worst-case coverage or highest ergodic metric on any information map. The highest ergodic metric achieved on the individual maps depends on the scalarized information map used to optimize the trajectory of the agent. When considering the minmax optimality criterion, the best-scalarized information map is determined by minimizing the maximum distance between the scalarized map and the individual maps. This concept is similar to solving the minimum enclosing circle problem. Thus, we can compute the required scalarized map as the center of the minimum bounding sphere of the information maps in Fourier space.

The minimum bounding sphere for a set of points $\{p_1, p_2, \cdot, p_n\}$ is unique and is the sphere with minimum volume such that all the points lie within or on the surface of the sphere.

In an ideal scenario, a trajectory optimized towards the center of the minimum bounding sphere will precisely converge to that distribution and the ergodic metric in the trajectory optimization converges to zero. By definition of minimum bounding spheres, the maximum distance of the center of the sphere to any of the information maps is equal to the radius of the sphere (derived later in this section). As a result, the maximum ergodicity of this trajectory on any of the individual information maps

Table 3.1: **MBS scalarization variables definition:** Variables and functions used in the proof

Variable	Explanation
n_o	Number of assigned information maps
$\mathcal{M} = \{\phi^{(1)}, \phi^{(2)}, \cdots\}$ with $\phi^{(i)} : \mathcal{W} \to \mathbb{R}$	Set of information maps or information distributions
$K \in \mathbb{R}$	Number of Fourier basis functions used
F: 2D distribution $\to \mathbb{R}^{K^2}$	Fourier transform: Outputs vector of Fourier coefficients
$\lambda \in \mathbb{R}^{K^2}$	Weight for each corresponding Fourier coefficient
feature: 2D distribution $\to \mathbb{R}^{K^2}$	Outputs a vector of weighted Fourier coefficients: feature(A) $= \lambda^{0.5} \odot F(A)$
C $\in \mathbb{R}^{K^2}$	Center of the MBS of features of assigned maps, Can be reconstructed into an information map
r $\in \mathbb{R}$	Radius of the MBS of features of assigned maps
getMap: $R^{K^2} \to \mathcal{M}$	Reconstructs the information map from a feature vector
W $\in \mathbb{R}^{K^2}$	Feature of the average of the assigned information maps, i.e., $W = feature(\frac{1}{n_o} \sum_{i \in [1,n_o]} \phi^{(i)})$
$Q = \{q, q : [0,T] \to \mathcal{W}\}$	Set of feasible trajectories for agent considered ($T \in \mathcal{R}^+$: time horizon)
$\mathcal{E}$: $\mathcal{M} \times Q \to \mathbb{R}$	Computes the ergodic metric of $q \in Q$ on $\phi \in \mathcal{M}$
q^*_ϕ: $[0,T] \to \mathcal{W}$	The locally optimal ergodic trajectory on map $q^*_\phi = \text{argmin}_{\forall q \in Q} \mathcal{E}(\phi, q)$
$E_\phi \in \mathbb{R}$	The minimum ergodic metric achieved by the agent on the map $E_\phi = \min_{\forall q \in Q} \mathcal{E}(\phi, q)$
$\mathcal{E}_w$: $\mathcal{M}^{n_o} \times Q \to \mathbb{R}$	Computes the highest ergodic metric of $q \in Q$ on the maps: $\mathcal{E}_w = \max_{i \in [1,n_o]} \mathcal{E}(\phi^{(i)}, q)$

will be proportional to the square of the radius of the sphere. Importantly, this result holds true regardless of the starting position of the agent being considered and is the minmax ergodic metric that can be achieved when the agent is assigned multiple information maps. Consequently, for SA-MO-ES, we compute the optimal scalarized map by determining the center of the minimum bounding sphere (MBS) of the information maps in Fourier space.

Now, let us suppose the trajectory optimized does not achieve an ergodic metric of zero against the center of the minimum bounding sphere. Even in this scenario, we can bound the maximum ergodic metric of the trajectory on the maps as stated in Theorem 1. Consider the following variables defined in Table 3.1.

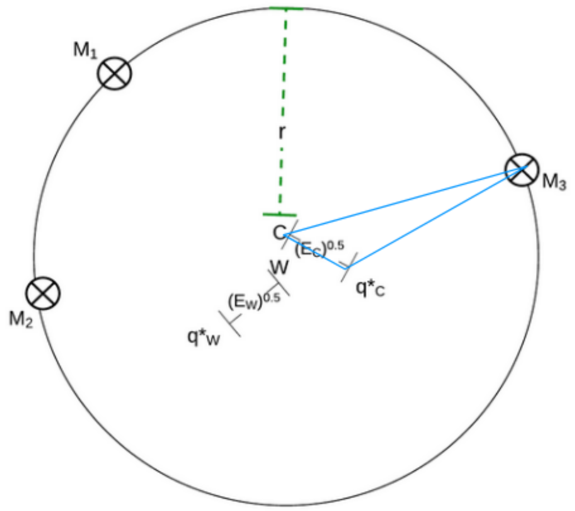
Theorem 1. *Consider an agent tasked to cover a domain subject to information maps $\Phi = \{\phi^{(1)}, \phi^{(2)}, \cdots \phi^{(n_o)}\}$. Let the weighted Fourier coefficients (features) of each element of Φ be represented as $M_s = \{M_1, M_2, \cdots M_{n_o}\}$ where $M_i = feature(\phi^{(i)})$ and hence $M_i \in \mathbb{R}^{K^2}$. Let C and r represent the center and radius of the minimum bounding sphere of M_s. Consider q^*_C to be the trajectory obtained by ergodic trajectory optimization against $getMap(C)$. Let the ergodic metric of q^*_C on $getMap(C)$ be E_C.*

Then the ergodic metric of q_C^* on any $\phi^{(i)} \in maps$ is at most $E_C + r^2 + 2r\sqrt{E_C}$.

Proof of Theorem 1. The minimum bounding sphere of the feature vectors in M_s is represented by the center and radius C and r respectively. The center C is a feature vector in $\mathbb{R}^{K^2}$ and can be reconstructed to an information map, using $getMap$, that can be used to optimize for a trajectory. Let the feature vector of the average of Φ be W. The locally optimal ergodic trajectory of an agent when optimized against $getMap(C)$ and $getMap(W)$ is q_C^* and q_W^* with ergodic metric E_C and E_W respectively:

$$\mathcal{E}(getMap(C), q_C^*) = E_C$$
$$\mathcal{E}(getMap(W), q_W^*) = E_W$$

Next, we can derive the relation between Euclidean distance in the feature space and the ergodic metric. In the feature space, the Euclidean distance between two distributions is computed as:



(a)

Figure 3.2: **Minimum bounding sphere scalarization:** The figure represents three information maps, as feature vectors in Fourier space $\{M_1, M_2, M_3\}$, that are assigned to one agent. The trajectories q_C^* and q_W^* are the locally optimal ergodic trajectory of the agent when optimized against the information maps represented by $getMap(C)$ and $getMap(W)$ respectively.

$$\begin{aligned} dist(A,B) &= \sqrt{\sum (feature(A) - feature(B))^2} && (3.1)\\ &= \sqrt{\sum (\lambda^{0.5} \odot f(A) - \lambda^{0.5} \odot f(B))^2} && (3.2)\\ &= \sqrt{\sum_{k \in K} \lambda_k (f(A)_k - f(B)_k)^2} && (3.3) \end{aligned}$$

If we consider $A \in \mathcal{M}$ and $B \in q$, then the corresponding ergodic metric can be computed as:

$$\mathcal{E}(A,B) = \sum_{k \in K} \lambda_k (f(A)_k - f(B)_k)^2$$

Thus equation (3.3) becomes:

$$\begin{aligned} dist(A,B) &= \sqrt{\mathcal{E}(A,b)} && (3.4)\\ dist^2(A,B) &= \mathcal{E}(A,B) && (3.5) \end{aligned}$$

If $\mathcal{E}(getMap(C), q^*_C) \neq 0$, then the trajectory did not converge to C. In that case, the center of the minimum bounding sphere C, the feature vector of one of the information maps M_i, and the agent's trajectory q^*_C form a triangle as illustrated in Figure 3.2 for the 2D case. Then using triangle inequality we can write:

$$dist(M_i, q^*_C) \leq dist(M_i, C) + dist(C, q^*_C)$$

Since, the distances are positive, by squaring both sides we get:

$$dist^2(M_i, q^*_C) \leq dist^2(M_i, C) + dist^2(C, q^*_C) + 2dist(M_i, C)dist(C, q^*_C) \qquad (3.6)$$

Substituting equation (3.5) in equation (3.6), we get:

$$\mathcal{E}(M_i, q_C^*) \leq dist^2(M_i, C) + E_C + 2 * dist(M_i, C) * \sqrt{E_C} \tag{3.7}$$

By definition of the minimum bounding sphere, the maximum distance of any feature vector from the center of the sphere is equal to the radius of the sphere. Further, since equation (3.7) is true for all $M_i \in M_s$, we can bound the highest ergodic metric achieved on individual maps to be:

$$\mathcal{E}_w(M_s, q_C^*) \leq r^2 + E_C + 2r\sqrt{E_C} \tag{3.8}$$

□

And a consequence of theorem 1 is the statement in the next corollary.

Corollary 1.1. *The highest ergodic metric of q_C^* on any map does not exceed $E_C + 2r\sqrt{E_C}$ more than that achieved by q_W^* on the information maps.*

Since r^2 is the minimum maximum ergodic metric that can be achieved when M_s are assigned to a single agent (by definition of minimum bounding spheres), we also have:

$$\mathcal{E}_w(M_s, q_W^*) \geq r^2 \tag{3.9}$$

Thus, substituting equation (3.9) in equation (3.8), we can get an expression showing that the highest ergodic metric on the maps for the trajectory optimized against the center of the minimum bounding sphere does not exceed $E_C + 2r\sqrt{E_C}$ more than that achieved by the trajectory optimized against the average of the information maps.

$$\mathcal{E}_w(M_s, q_C^*) \leq \mathcal{E}_w(M_s, q_W^*) + E_C + 2r\sqrt{E_C}$$

This shows that even when the time average statistics of the trajectory does not converge exactly to the center of the minimum bounding sphere, the minmax metric achieved will be bounded by $E_C + r^2 + 2r\sqrt{E_C}$, where E_C is the ergodic metric of the trajectory optimized against the center of the minimum bounding sphere, r^2 is the best minmax ergodic metric achievable and r is the radius of the minimum bounding sphere.

Chapter 4

Multi-Agent Multi-Objective Ergodic Search (MA-MO-ES)

Now that we have addressed the single-agent sub-problem, we elaborate on our proposed approach for the main focus of the thesis: multi-agent multi-objective ergodic search (MA-MO-ES). We first introduce the mathematical preliminaries and problem formulation for the multi-agent multi-objective ergodic search.

4.1 Mathematical Preliminaries

Let $\mathcal{W} = [0, L_1] \times [0, L_2] \times \cdots \times [0, L_\nu] \subset \mathbb{R}^\nu$ denote a ν-dimensional workspace that is to be explored by the robots. Each robot has an (identical) n-dimensional state space $S = \mathcal{W} \times V$ $(n \geq \nu)$. V is comprised of the robot state components, such as velocities or orientations, that do not affect what the sensor can see. Let $q_i : [0, T] \to S$ denote a trajectory of the i^{th} robot in its state space with $T \in \mathbb{R}^+$ representing the time horizon. Let the set of all state space trajectories be H. Let $P : S \to \mathcal{W}$ project the state space into the workspace. The robots have deterministic dynamics given by $\dot{q}_i(t) = f(q_i(t), u_i(t))$, where $u_i(t) \in \mathbb{R}^m$ is the control input of the i^{th} robot.

Let $c(x, q_i) : \mathcal{W} \times H \to [0, 1]$ denote the time-averaged statistics of a trajectory q_i, which is defined as:

$$c(x, q_i) = \frac{1}{T}\int_0^T \delta(x - P(q_i(\tau)))d\tau, \tag{4.1}$$

where δ is a Dirac function. Let $\phi : \mathcal{W} \to \mathbb{R}$ denote a static information map that describes the amount of information at each location in the workspace. In this work, each information map is a probability distribution with $\int_{\mathcal{W}} d\phi = 1$ and $\phi(x) \geq 0, \forall x \in \mathcal{W}$. An ergodic metric [29] for a trajectory q_i and an information map ϕ is defined as:

$$\mathcal{E}(\phi, q_i) = \sum_{k=0}^{K} \lambda_k (c_k - \phi_k)^2 = \sum_{k=0}^{K} \lambda_k \left(\frac{1}{T}\int_0^T F_k(q(\tau))d\tau - \phi_k\right)^2 \tag{4.2}$$

where ϕ_k, $F_k(x)$, K, c_k, h_k, and λ_k are as defined in the background section 2.2.

Consider a set Φ of M information maps and a set R of N agents with $M \geq N$. Let $\mathcal{A}$ denote the set of all possible allocations of agents to maps. Each allocation is a *surjective* function $A : \Phi \to R$, i.e., every robot is assigned at least one information map. Let $p(r, A)$ be the set of maps assigned to robot r by allocation $A \in \mathcal{A}$.

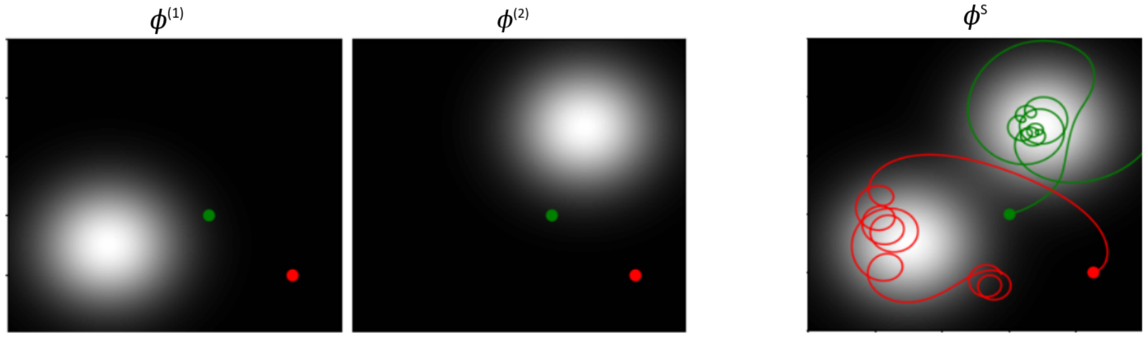
A scalarized information map combines a set of n_o information maps, using weights w_i for each map, into a single map against which the trajectory of an agent can be optimized with respect to the ergodic metric:

$$\phi^S = \frac{\sum_{i=1}^{n_o} w_i \phi^{(i)}}{\sum_{i=1}^{n_o} w_i} \tag{4.3}$$

4.2 Naive Approach: Joint Trajectory Optimization (JTO)

This section discusses a naive approach to the MA-MO-ES problem. We adapt our prior work in single-agent multi-objective ergodic search [38] to multiple agents, in a similar way as in [29], by optimizing a concatenated trajectory of all agents for the ergodic metric on a scalarization of all maps with $w_i = 1$ in equation (4.3).

For each agent, i, consider a trajectory $q_i(t)$ of length T. The concatenated tra-



(a) (b)

Figure 4.1: **Joint Trajectory Optimization (JTO):** (a) Example problem with two maps and two agents R (red) and G (green) with specified start position and zero orientation, (b) Trajectories of agents on the scalarized information map (ϕ^S)

Table 4.1: **Comparison of Joint Trajectory Optimization and Greedy Allocation:** Individual ergodicities on maps in Fig 4.1a using joint trajectory optimization (JTO) and greedy allocation of $\phi^{(1)}$ and $\phi^{(2)}$ assigned to agent green (G) and red (R) respectively

Method	Ergodic metric evaluated on		
	Scalarized Map	Map 1	Map 2
JTO	0.0079101	0.2536889	0.1626450
Greedy Allocation ($G \rightarrow \phi^{(1)}, R \rightarrow \phi^{(2)}$)	-	0.0009996	0.0009993

jectory of length NT can be represented as $q(t) = [q_1, q_2, \cdots, q_N]'$. The concatenated trajectory is then optimized to minimize the ergodic metric ($\mathcal{E}(\phi^S, q)$), calculated using equation (4.2). This strategy is implemented on an example problem with two information maps and two agents with random start positions, as shown in Figure 4.1a. The resulting trajectories of the two agents and the individual ergodicities are shown in Figure 4.1b and Table 4.1, respectively.

It can be seen that since it is a joint trajectory optimization, the agents naturally divide the high information regions among them. This seems favorable, however, since the agents are trying to cover the domain while considering all the M maps simultaneously, the lowest ergodic metric it can achieve on each map without compromising others is higher than what it can achieve even with a simple greedy allocation as shown in Table 4.1. A visual representation of this limitation can be seen in Figure

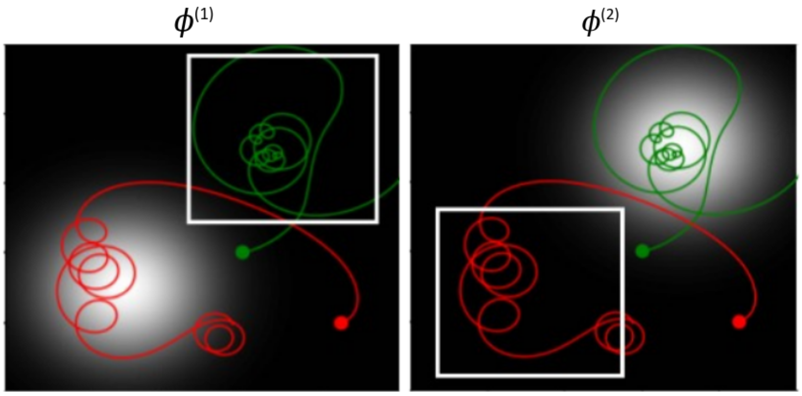


Figure 4.2: **Result of Joint Trajectory Optimization:** Agent trajectories on the individual maps in Figure 4.1a

4.2 where agents spend a lot of time in low-information regions of one map, indicated by the portion of the trajectory in the white box, as that region spatially corresponds to a region with high information on another map. This problem can be overcome by allocating these maps and having the agents access information only from the assigned maps instead of having each agent consider all the maps. The trajectories of the agent with the simple greedy allocation of $\phi^{(1)}$ and $\phi^{(2)}$ assigned to agent green (G) and red (R) respectively is shown in Figure 4.3. **Thus, the rest of the thesis focuses on computing the optimal allocation of information maps to agents for the MA-MO-ES problem.**

4.3 Problem Formulation

Each problem has N agents tasked to cover a domain subject to M information maps that span the same physical region, $M \geq N$. We present results for identical forward-moving-only differential-drive robots, varying only in initial position, with $\mathcal{W} = [0, 100] \times [0, 100]$ and $S = \mathcal{W} \times SO\{2\}$. The dynamics for the i^{th} robot are shown in equation (4.4), in which $\nu = 2, n = 3, m = 2$.

$$\dot{q}_i = f(q_i(t), u_i(t)) = \begin{bmatrix} v_i cos(\theta_i(t)) \\ v_i sin(\theta_i(t)) \\ \alpha * \omega_i \end{bmatrix} \tag{4.4}$$

where, $[v_i,\omega_i]$ is the control input for robot i, α is a constant and $q_i(t) = [x_i(t), y_i(t), \theta_i(t)]$ is the state of robot i at time t.

Let $q_r^*(A)$ be the trajectory (satisfying the dynamics above) that minimizes ergodic metric on the scalarization of the maps assigned to robot r by allocation A:

$$q_r^*(A) = \underset{q \in H}{\operatorname{argmin}}\, \mathcal{E}(\phi_{p(r,A)}, q) s.t. q(0) = q_r(0) \tag{4.5}$$

$q_r^*(A)$ is obtained in practice via trajectory optimization on the scalarized information map obtained using minimum bounding sphere scalarization detailed in Section 3.3. Then the allocation optimization problem can be stated as follows:

$$\underset{A \in \mathcal{A}}{\operatorname{argmin}}\, \max_{\phi \in \Phi} \mathcal{E}(\phi, q^*_{A(\phi)}(A)) \tag{4.6}$$

which finds the allocation for which the map with the largest ergodic metric has the smallest achievable value.

4.4 Branch and bound approach (BB)

We consider four baseline approaches to this problem explained further in Section 4.5. One approach involves optimizing a single joint trajectory for all agents on a scalarized information map obtained using minimum bounding sphere scalarization, which may not be optimal for individual maps. Another approach is a suboptimal greedy allocation method that assigns each agent to a map based on the information around the agent. Yet another approach assigns agents to information maps based on the distance of the agent to the peaks of information on the maps. However, this too is not optimal. The optimal solution can be found by the fourth baseline approach, exhaustive search, but this is not practical for larger numbers of agents and maps. To tackle this we present a branch and bound approach to this problem.

The branch and bound (BB) algorithm is commonly used in literature to speed up the exhaustive search of combinatorial problems such as equation (4.6). It constructs a tree structure and uses bounds to eliminate sub-problems that cannot contain the optimal solution. In a minimization problem, if the cost of a sub-problem is greater than the current best cost (upper bound), this sub-problem can be eliminated. In this approach, we leverage the minmax metric to construct a branch and bound algorithm that reduces the number of possible allocations to be checked to find the optimal allocation. The formulation is described in Algorithm 1.

The algorithm begins by computing an incumbent solution using the greedy allocation described in Section 4.5.1 (Step 1). For this incumbent allocation, the ergodic trajectory for each agent is computed using (4.5), and the resulting individual ergodicities (using **getIndvErg()**) on the information maps are computed using (4.2) (Steps $2, 3$). The maximum of the individual ergodicities is set as the initial upper bound for the branch and bound algorithm (Step 4).

The algorithm works on a tree structure where each node corresponds to an allocation of maps to one agent. It first creates a root node with a null allocation (Step 5). Each subsequent level of the tree contains nodes corresponding to possible allocations of maps to one agent. Hence, the depth of the tree generated is equal to the number of agents in the problem, and a path from the root node to a leaf node corresponds to one complete allocation.

For every node in the tree, the individual ergodicities on the maps are computed using equations (4.5) and (4.2) (Step 12). If the maximum of the individual ergodicities is greater than the current upper bound, any solution containing this partial allocation will have a minmax metric greater than the current upper bound, and hence the node is pruned (Steps $13 - 15$). Whenever a complete solution is obtained, the upper bound is updated to facilitate more node pruning (Steps $17 - 24$).

On the tested problems this branch and bound algorithm achieves an order of magnitude compared to the exhaustive search algorithm while maintaining optimality.

Algorithm 1: Branch and Bound Algorithm

```
Data: Information maps: $\Phi = \{\phi^{(1)}, \cdots, \phi^{M}\}$, Agent start positions:
        $S_0 = \{s_1, \cdots, s_N\}$
Result: Optimal allocation scheme
$incumbent \leftarrow$ GreedyAllocation($\Phi$, $S_0$);
$best\_alloc \leftarrow incumbent$;
$indv\_erg \leftarrow$ getIndvErg($incumbent$,$S_0$);
$UB \leftarrow \max(indv\_erg)$;
$root\_node \leftarrow \emptyset$;
$candidate\_nodes \leftarrow [root]$;
for $i \in [1, N]$ do
    $new\_nodes \leftarrow \emptyset$
    for $n \in candidate\_nodes$ do
        $allocs \leftarrow$ Possible assignments of maps left for agent $i$;
        for $alloc \in allocs$ do
            $indv\_erg \leftarrow$ getIndvErg($alloc$,$s_i$);
            if $max(indv\_erg) > UB$ then
                continue; // Prune
            end
            $new\_nodes.append(alloc)$;
            if $i == N$ then
                $new\_alloc \leftarrow$ path from root to node;
                $new\_UB \leftarrow \max($getIndvErg($new\_alloc$,$S_0$)$)$;
                if $new\_UB < UB$ then
                    $UB \leftarrow new\_UB$;
                    $best\_alloc \leftarrow new\_alloc$;
                end
            end
        end
    end
    $candidate\_nodes \leftarrow new\_nodes$;
end
return best_alloc
```

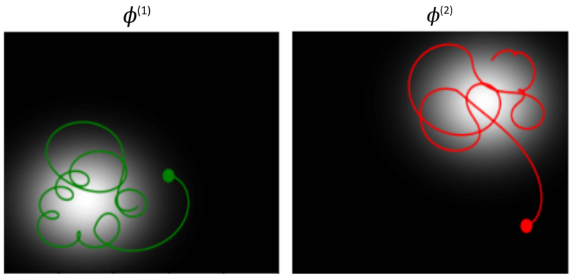


Figure 4.3: **Trajectories obtained using greedy allocation:** Here the agent G (green) and R (red) are assigned to the first and second information maps, respectively

4.5 Comparison against baseline methods and prior work

We discuss the baseline methods used to compare against the branch and bound approach.

4.5.1 Greedy Allocation

The greedy baseline is presented in Algorithm 2. For each agent and information map combination, the amount of information inside a window of size 30X30 centered on the agent in that information map is computed as the score of that agent-map combination (Steps $3-7$). The agent with the maximum score on a map is assigned to that map unless this would leave more unassigned agents than maps, in which case the next highest score is assigned. This ensures all agents are assigned at least one map (implemented as **getBestAgent**(scores) in Step 10). This is tested on the example in Figure 4.1a, and the trajectories obtained from that allocation are plotted in Figure 4.3. The agents now spend less time in areas of low information. It can be seen from Table 4.1 that the individual maps have a lower ergodic metric when the agents are explicitly allocated. Though the greedy allocation performs better than JTO, it does not guarantee an optimal allocation as defined in Section 4.1.

Algorithm 2: Greedy Allocation

```
Data: Information maps: $\Phi = \{\phi^{(1)}, \cdots, \phi^{(M)}\}$, Agent start positions:
      $S_0 = \{s_1, \cdots, s_N\}$
Result: Allocation Scheme
$scores \leftarrow zero\_matrix(M, N)$;
$W \leftarrow$ window centered on each start position;
for $\phi_i \in \Phi$ do
    for $s_j \in S_0$ do
        $scores[i][j] \leftarrow \sum_{x,y \in W_j} \phi^{(i)}(x, y)$;
    end
end
$allocation \leftarrow \{\}$;
for $i \in [1, M]$ do
    $allocation[i] \leftarrow$ getBestAgent($scores[i]$);
end
```

4.5.2 Exhaustive Search

An optimal baseline is an exhaustive search over $\mathcal{A}$. The algorithm iterates through all possible allocations for the given set of agents and information maps. For each allocation, the individual ergodicities on the information maps are computed. It is worth noting that when an agent is assigned more than one information map, we scalarize the maps using the minimum bounding sphere scalarization. Finally, the optimal allocation is chosen as described in equation (4.6). For the example shown in Figure 4.4, the search through possible allocations is shown in Table 4.2.

4.5.3 Numerical Results

The effectiveness of the proposed branch and bound (BB) algorithm was tested on 70 randomly generated test cases. Each test case contained 4 to 10 information maps (objectives) and a random start pose for each of the 4 agents. The branch and bound algorithm was compared against (a) Joint Trajectory Optimization, (b) Exhaustive Search, and (c) Greedy Allocation. The exhaustive search was run with a net time limit of 100 hr. The approaches are evaluated against the branch and bound algorithm in terms of runtime and the difference in the minmax metric. The results

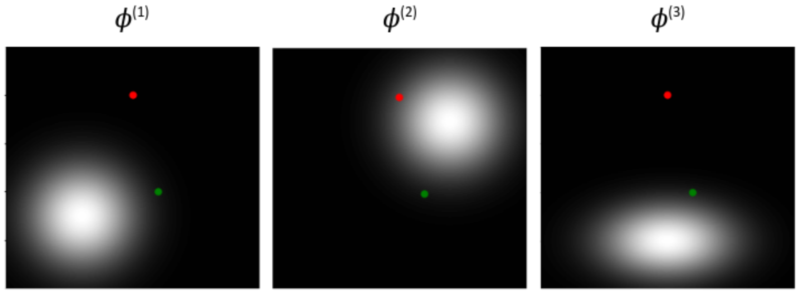


Figure 4.4: **Example test case:** An example test case with three information maps $\phi^{(1)}, \phi^{(2)}, \phi^{(3)}$ and two agents whose start positions are represented by G (green) and R (red) dots respectively

Table 4.2: **Exhaustive search through allocation space:** Search through all possible allocations for the problem shown in Fig 4.4 by the exhaustive search algorithm. Here $[1, 2] \rightarrow R$ refers to agent R being assigned to information maps 1 and 2. The best allocation according to the minmax metric is highlighted.

	Individual Ergodicities			
Allocation	**Map 1**	**Map 2**	**Map 3**	**Max**
$[1] \rightarrow R, [2, 3] \rightarrow G$	0.003175	0.194149	0.191870	0.194149
$[2, 3] \rightarrow R, [1] \rightarrow G$	0.001005	0.190901	0.194245	0.194245
$[1, 2] \rightarrow R, [3] \rightarrow G$	0.201785	0.200855	0.005364	0.201785
$[3] \rightarrow R, [1, 2] \rightarrow G$	0.201844	0.201537	1.076468	1.076468
$[2] \rightarrow R, [1, 3] \rightarrow G$	**0.047297**	**0.000712**	**0.046686**	**0.047297**
$[1, 3] \rightarrow R, [2] \rightarrow G$	0.047829	0.000657	0.048254	0.048254

for test cases with 4 agents are shown in Table 4.3.

Comparison between BB and Joint trajectory optimization The joint trajectory optimization, on average, showed a 98% reduction in runtime compared to the branch and bound algorithm. This is because there is no allocation involved in this approach. However, the minmax metric is much higher compared to the branch and bound approach, as shown in Table 4.3. The comparison of runtime for the joint trajectory optimization and the branch and bound algorithm for test cases with 4 agents is shown in Figure 4.5.

Comparison between BB and Exhaustive Search The branch and bound algorithm finished solving all the test cases within 100 hr while the exhaustive search algorithm solved only 6 test cases within 100 hr. Thus, the branch and bound

Table 4.3: **Comparing branch and bound against baselines:** Comparing runtime and minmax metric of the branch and bound algorithm (optimal) against prior works: joint trajectory optimization and greedy allocation

	Average Percentage		Average		
Method	Improvement in runtime	Increase in minmax metric	Runtime (s)	Increase in minmax metric	Maximum difference in minmax metric
Joint Trajectory Optimization	**98.88%**	**915.57%**	**4.38**	**0.1484**	**0.4613**
Greedy Allocation	**96.02%**	**399.86%**	**15.06**	**0.0935**	**0.4465**

algorithm described in Section 4.4 achieves an order of magnitude improvement in runtime while maintaining optimal allocation.

Comparison between BB and Greedy Allocation The greedy allocation, on average, showed a 96% reduction in runtime compared to the branch and bound algorithm. However, the minmax metric is higher compared to the branch and bound approach, as shown in Table 4.3, as the approach only takes information in the local neighborhood of agents for task assignment. The comparison of runtime for the greedy algorithm and the branch and bound algorithm for test cases with 4 agents is shown in Figure 4.5.

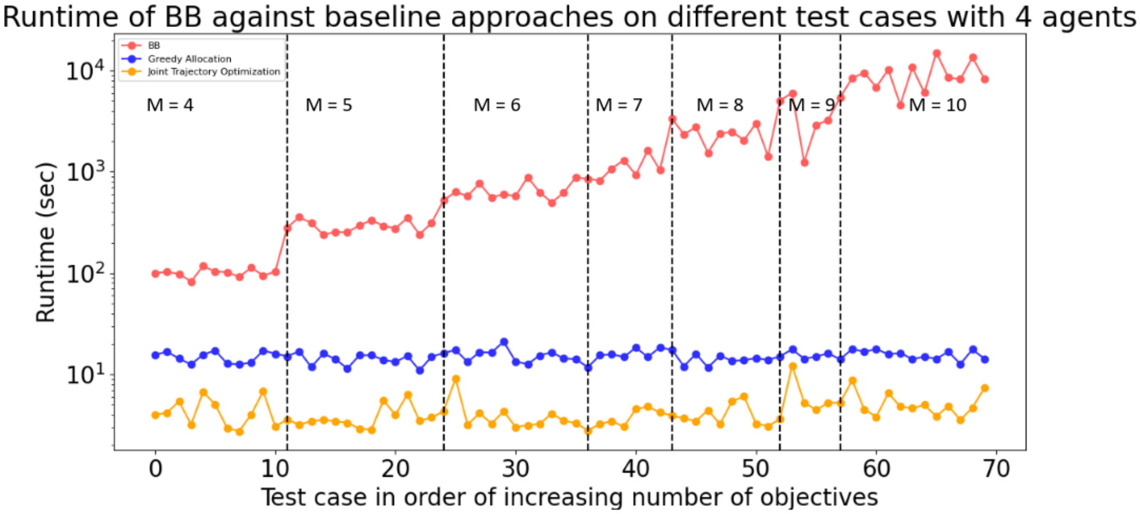


Figure 4.5: **Runtime comparison:** Runtime comparison of BB with baseline approaches on test cases with 4 agents and $M \in [4, \cdots, 10]$ information maps

Chapter 5

Clustering

On the tested problems the proposed branch and bound algorithm achieves an order of magnitude improvement in runtime compared to the exhaustive search algorithm while maintaining optimality. However, the branching factor is equal to the size of the power set of the number of unassigned information maps. Further, if two information maps are similar in regions of high information, we can estimate that they should be assigned to the same agent but branch and bound overlooks the similarity between information maps while allocating agents. To this end, we present two approaches based on clustering to reduce the branching factor in the branch and bound and thereby significantly reduce the runtime of the algorithm. Clustering before allocation sacrifices guaranteed optimality for improved computational performance.

5.1 K-means clustering

This approach is motivated by the fact that if two information maps are similar, then a single agent can cover the domain by considering both information maps together. We define the similarity (sim_{pq}) between two information maps $\phi^{(p)}$ and $\phi^{(q)}$ to be the norm of the difference in the weighted Fourier coefficients of the information maps as shown in (5.1) [29][38].

$$sim_{pq} = \sum_{k \in \mathcal{K}} \lambda^{0.5} \odot \|\phi_k^{(p)} - \phi_k^{(q)}\| \tag{5.1}$$

where $\phi_k^{(p)}$ and $\phi_k^{(q)}$ are the Fourier coefficients of $\phi^{(p)}$ and $\phi^{(q)}$ respectively, λ is the weight vector, and k is the set of coefficient indices as defined in Section 2.2. The information maps are then clustered based on this similarity metric using K-means clustering. The number of clusters is picked using the kneedle algorithm [40], which identifies the points of maximum curvature on a dataset while ensuring it is greater than or equal to the number of robots. We then follow the branch and bound algorithm as described in Algorithm 1, except now, each level corresponds to the possible allocations of clusters to one agent. The tree thus formed has the same depth but has a lower branching factor than the branch and bound approach.

Branch and bound with k-means clustering helps reduce the runtime of the algorithm by effectively reducing the number of allocations to be searched. However, the approach results in a slightly increased minmax metric as compared to the branch and bound algorithm as seen in Table 5.1 and 5.2. This can occur when the approach groups two maps into one cluster but the maps are assigned to different agents in the optimal allocation. K-means clustering attempts to reduce the sum of the squared distances of the elements in a cluster from the centroid. However, since our optimality criterion is the minmax metric, we should minimize the maximum distance of the elements in a cluster from the centroid of the cluster. This leads us to the next clustering approach.

5.2 Minimum Bounding Sphere clustering

In this approach, we aim to find N clusters, where N is the number of agents, by using the minimum bounding sphere formulation. Section 3.3 proved that minimum bounding sphere scalarization is the best scalarization to use to minimize the highest ergodic metric on any information map when a single agent is tasked to cover the domain subject to multiple information maps. It further derived that the highest ergodic metric is proportional to the square of the radius of the minimum bounding sphere of the information maps. With multiple agents, to achieve the minmax ergodicity, it is thus necessary to minimize the maximum radius across the minimum bounding spheres of each agent's assigned information maps. We thus aim to find N minimum bounding spheres in the Fourier space such that each information map is included in exactly one sphere and the maximum radius of the spheres is minimized.

Algorithm 3: Minimum bounding sphere clustering

```
Data: Information maps: $\Phi = \{\phi^{(1)}, \cdots, \phi^{(M)}\}$, Number of agents: $N$
Result: Clusters
$UB \leftarrow FLOAT\_MAX$;
$best\_clustering \leftarrow \emptyset$;
$root\_node \leftarrow \emptyset$;
$candidate\_nodes \leftarrow [root]$;
for $i \in [1, N]$ do
    $new\_nodes \leftarrow \emptyset$
    for $n \in candidate\_nodes$ do
        $groups \leftarrow$ Possible groupings of maps left for cluster $i$;
        for $group \in groups$ do
            $r \leftarrow$ getMinimalBoundingSphere($group$);
            if $r > UB$ then
                continue; // Prune
            end
            $new\_nodes.append(group)$;
            if $i == N$ then
                $new\_clustering \leftarrow$ path from root to node;
                $radii \leftarrow \emptyset$
                for $c \in new\_clustering$ do
                    $radii.append($getRadius($c$)$)$;
                end
                $new\_UB \leftarrow$ max(radii);
                if $new\_UB < UB$ then
                    $UB \leftarrow new\_UB$;
                    $best\_clustering \leftarrow new\_clustering$;
                end
            end
        end
    end
    $candidate\_nodes \leftarrow new\_nodes$;
end
return $best\_clustering$
```

This is illustrated in Algorithm 3. The algorithm begins with $FLOAT\_MAX$ as the upper bound (Step 1). The algorithm works on a tree structure where each node corresponds to a group of maps in one cluster. It first creates a root node with a null grouping (Step 3). Each subsequent level of the tree contains nodes corresponding to possible groupings of maps in one cluster. Hence, the depth of the tree generated is equal to the number of clusters which is set equal to the number of agents in the problem, and a path from the root node to a leaf node corresponds to one complete clustering.

For every node in the tree, the minimal bounding sphere for the chosen maps is computed using the function **getMinimalBoundingSphere**[1]. If the radius of the minimum bounding sphere is greater than the current upper bound, any solution containing this partial clustering will have a minmax metric greater than the current upper bound, and hence the node is pruned (Steps $11 - 13$). Whenever a complete clustering is obtained, the upper bound is updated to facilitate more node pruning (Steps $15 - 25$).

Once we get the clusters, we can use branch and bound to determine which agent assignment to clusters as described in the branch and bound with the k-means clustering approach. Note that in this approach, there are two branch and bound trees involved: for computing the clusters of information maps and for assigning these clusters to agents. In the clustering process, there is no pruning and hence the tree constructed has a full-size branching factor. However, each node's computation (fitting a minimum bounding sphere) is very fast compared to ergodic trajectory optimization.

5.3 Results: Comparison of BB with clustering against plain BB

We use a prior distance-based allocation [7] along with clustering as another comparison to our approach. For this comparison, the information maps in a test case are clustered as described in Section 5.1 but assigned based on distance as described in [7].

[1] This function utilizes the miniball library that implements the minimum bounding sphere algorithm as in [17]. The library implementation can be found at https://github.com/hbf/miniball/tree/master

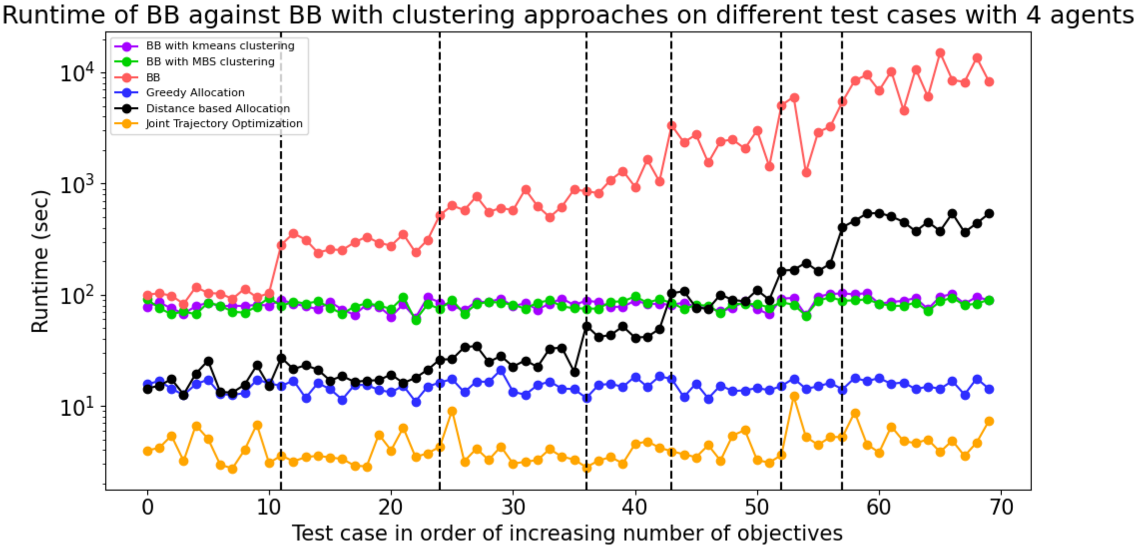


Figure 5.1: **Runtime comparison:** Runtime comparison of BB against BB with clustering approaches and clustering with distance-based allocation on test cases with 4 agents and $M \in [4, \cdots, 10]$ information maps

For each cluster, the peaks on the corresponding scalarized information map (average of the maps in a cluster) are identified. The agents are assigned to the clusters based on the distance of the agent to the centroid of these peaks.

The effectiveness of the proposed branch and bound (BB) with similarity clustering algorithms and the distance-based allocation with clustering were tested on 70 randomly generated test cases with 4 agents. Each test case contained 4 to 10 information maps (objectives) and a random start pose for each of the agents. The plain branch and bound algorithm (optimal) with weighted average and minimum bounding sphere scalarization was compared against (a) Distance-based allocation with clustering, (b)Branch and bound with k-means clustering, and (c) Branch and bound with minimum bounding sphere clustering. The results are shown in Table 5.1 and Table 5.2 respectively. The runtime comparison of the plain BB against the BB with clustering approaches is shown in Figure 5.1.

Figure 5.1 shows that the distance-based allocation has a much lower runtime than the branch and bound approaches. However, the minmax metric achieved by this method is much worse as seen in Table 5.1 and 5.2. It can be seen from Table 5.1 and 5.2 that with the minimum bounding sphere scalarization, both the branch

Table 5.1: **Comparison of BB against BB with similarity clustering:** Compares the allocation and minmax metric of the branch and bound with weighted average scalarization against the branch and bound with k-means and minimum bounding sphere clustering

	When compared to BB (optimal) with weighted average scalarization					
	Number of cases with			Among the cases when allocation didn't match		
	same allocation	equivalent allocation (same minmax metric)	optimal allocation	Maximum difference in minmax metric	Average difference in minmax metric	Std. deviation of diff. in minmax metric
Clustering with distance-based allocation	4	0	4 (5.71%)	0.6284	0.0678 (197.7%)	0.1289
K-means clustering	33	0	33 (47.14%)	0.2234	0.0277 (32.7%)	0.0473
Minimal bounding sphere clustering	38	0	38 (54.29%)	**0.1033**	**0.0210 (21.4%)**	**0.0344**

Table 5.2: **Comparison of BB against BB with similarity clustering:** Compares the allocation and minmax metric of the branch and bound with minimum bounding sphere scalarization against the branch and bound with k-means and minimum bounding sphere clustering

	When compared to BB (optimal) with minimum bounding sphere scalarization					
	Number of cases with			Among the cases when allocation didn't match		
	same allocation	equivalent allocation (same minmax metric)	optimal allocation	Maximum difference in minmax metric	Average difference in minmax metric	Std. deviation of diff. in minmax metric
Clustering with distance-based allocation	4	12	16 (22.86%)	0.6293	0.0846 (205.2%)	0.1379
K-means clustering	42	0	42 (60%)	0.2242	0.0398 (37.8%)	0.0496
Minimal bounding sphere clustering	49	7	56 (80%)	**0.1027**	0.0544 (25.9%)	**0.0285**

and bound with clustering approaches improve in terms of the number of optimal allocations in the test cases, and the minmax metric as expected.

The branch and bound with minimum bounding sphere clustering performs better than the one with k-means clustering in terms of the minmax metric achieved irrespective of the scalarization used. Further, Figure 5.1 shows that both branch and bound with clustering approaches have similar runtimes that are two orders of magnitude better than the branch and bound without clustering. Finally, the branch and bound with minimum bounding sphere clustering outputs the optimal allocation with an average deviation of 20% in the minmax ergodic metric.

Chapter 6

Conclusions

The multi-agent multi-objective ergodic search problem is an allocation problem that is NP-hard to solve optimally. An exhaustive search algorithm while optimal in allocation quickly becomes intractable in terms of runtime as the number of agents and maps increases. To tackle this, we present a branch and bound algorithm that helps reduce the average runtime by an of magnitude while still providing the optimal allocation scheme. Further, we present two approaches to cluster similar information maps to reduce the branching factor in the branch and bound algorithm. This method further reduces the runtime by two orders of magnitude and achieves good quality allocations with an average 20% deviation from the optimal minmax ergodic metric. As the runtime benefit is the result of a reduced branching factor in the branch and bound, we expect that for increasing numbers of agents, both our approaches will yield even greater benefits.

6.1 Contributions

This thesis presents three main contributions in the field of multi-agent multi-objective ergodic search:

1. Computing the best-scalarized information map when an agent is tasked to cover the domain subject to more than one information map to minimize the maximum ergodic metric on the maps

2. A branch and bound formulation to compute the minmax optimal allocation scheme for the multi-agent multi-objective ergodic search (MA-MO-ES) problem
3. Two improvements based on clustering approaches to leverage similarity between information maps to further reduce the runtime of the proposed branch and bound formulation while providing good quality solutions

6.2 Future Work

Future directions for this work include adapting the task allocation scheme to agents with heterogeneous capabilities. In this work, we have considered that one agent is enough to handle one objective characterized by an information map. Studying cases where searching a domain subject to an information map requires multiple agents working together is also part of future work. Further, additional metrics such as workload division and agent-specific task allocation can also be investigated to expand the applications of this problem. Finally, similarity metrics that consider agent position, orientation, and dynamics when clustering information maps can further improve the allocation scheme by making the estimate of the cluster of maps that should be assigned to one agent more accurate.

Bibliography

[1] Vitaly Ablavsky and Magnus Snorrason. *Optimal search for a moving target - A geometric approach*. doi: 10.2514/6.2000-4060. URL https://arc.aiaa.org/doi/abs/10.2514/6.2000-4060. 1.1

[2] Ian Abraham and Todd D. Murphey. Decentralized ergodic control: Distribution-driven sensing and exploration for multiagent systems. *IEEE Robotics and Automation Letters*, 3(4):2987–2994, oct 2018. doi: 10.1109/lra.2018.2849588. 2.2.1

[3] Ian Abraham, Ahalya Prabhakar, and Todd D. Murphey. An ergodic measure for active learning from equilibrium. *IEEE Transactions on Automation Science and Engineering*, 18(3):917–931, 2021. doi: 10.1109/TASE.2020.3043636. 2.2.1

[4] Esther M. Arkin, Sándor P. Fekete, and Joseph S.B. Mitchell. Approximation algorithms for lawn mowing and millinga preliminary version of this paper was entitled "the lawnmower problem" and appears in the proc. 5th canad. conf. comput. geom., waterloo, canada, 1993, pp. 461–466. *Computational Geometry*, 17(1):25–50, 2000. ISSN 0925-7721. doi: https://doi.org/10.1016/S0925-7721(00)00015-8. URL https://www.sciencedirect.com/science/article/pii/S0925772100000158. 1.1

[5] Elif Ayvali, Hadi Salman, and Howie Choset. Ergodic coverage in constrained environments using stochastic trajectory optimization, 2017. 2.2.1

[6] Jean Berger, Nassirou Lo, and Martin Noel. A new multi-target, multi-agent search-and-rescue path planning approach. *International Journal of Computer and Information Engineering*, 8(6):978 – 987, 2014. ISSN eISSN: 1307-6892. 1.1, 2.3

[7] Sumana Biswas, Sreenatha G. Anavatti, and Matthew A. Garratt. A time-efficient co-operative path planning model combined with task assignment for multi-agent systems. *Robotics*, 8(2), 2019. ISSN 2218-6581. doi: 10.3390/robotics8020035. 1.2, 2.3, 5.3

[8] Richard Bormann, Florian Jordan, Joshua Hampp, and Martin Hägele. Indoor coverage path planning: Survey, implementation, analysis. In *2018 IEEE In-*

ternational Conference on Robotics and Automation (ICRA), pages 1718–1725, 2018. doi: 10.1109/ICRA.2018.8460566. 2.1

[9] Lauren Bramblett, Rahul Peddi, and Nicola Bezzo. Coordinated multi-agent exploration, rendezvous, and task allocation in unknown environments with limited connectivity. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12706–12712, 2022. doi: 10.1109/IROS47612.2022.9981898. 1.2, 2.3

[10] Jonathan Butzke and Maxim Likhachev. Planning for multi-robot exploration with multiple objective utility functions. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3254–3259, 2011. doi: 10.1109/IROS.2011.6095165. 2.3

[11] Daniele Calisi, Alessando Farinelli, Luca Iocchi, and Daniele Nardi. Multi-objective exploration and search for autonomous rescue robots. *J. Field Robotics*, 24:763–777, 08 2007. doi: 10.1002/rob.20216. 2.3

[12] Wei Chen, Junjie Liu, Yang Tang, Jian Huan, and Hao Liu. Trajectory optimization of spray painting robot for complex curved surface based on exponential mean bézier method. *Mathematical Problems in Engineering*, 2017:1–10, 11 2017. doi: 10.1155/2017/4259869. 1.1

[13] Howie Choset. Coverage for robotics - a survey of recent results. *Ann. Math. Artif. Intell.*, 31:113–126, 10 2001. doi: 10.1023/A:1016639210559. 2.1

[14] Howie Choset and Philippe Pignon. Coverage path planning: The boustrophedon cellular decomposition. In Alexander Zelinsky, editor, *Field and Service Robotics*, pages 203–209, London, 1998. Springer London. ISBN 978-1-4471-1273-0. 1.1

[15] Michael Emmerich and André Deutz. A tutorial on multiobjective optimization: fundamentals and evolutionary methods. *Natural Computing*, 17, 09 2018. doi: 10.1007/s11047-018-9685-y. 3.1

[16] Osama Ennasr, Giorgos Mamakoukas, Todd Murphey, and Xiaobo Tan. Ergodic exploration for adaptive sampling of water columns using gliding robotic fish. page V003T32A016, 09 2018. doi: 10.1115/DSCC2018-9179. 1.1

[17] Kaspar Fischer, Bernd Gärtner, and Martin Kutz. Fast smallest-enclosing-ball computation in high dimensions. pages 630–641, 09 2003. ISBN 978-3-540-20064-2. doi: 10.1007/978-3-540-39658-1_57. 1

[18] Enric Galceran and Marc Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61(12):1258–1276, 2013. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2013.09.004. URL https://www.sciencedirect.com/science/article/pii/S092188901300167X. 2.1

[19] Elliot Garcia and Pablo Gonzalez-de Santos. Mobile-robot navigation with

complete coverage of unstructured environment. *Robotics and Autonomous Systems*, 46:195–204, 04 2004. doi: 10.1016/j.robot.2004.02.005. 2.1

[20] B.P. Gerkey and M.J. Mataric. Multi-robot task allocation: analyzing the complexity and optimality of key architectures. In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, volume 3, pages 3862–3868 vol.3, 2003. doi: 10.1109/ROBOT.2003.1242189. 2.3

[21] Brian P. Gerkey and Maja J. Matarić. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research*, 23:939 – 954, 2004. 1.2

[22] Changxin Huang, Li Wei, Chao Xiao, Binbin Liang, and Songchen Han. Potential field method for persistent surveillance of multiple unmanned aerial vehicle sensors. *International Journal of Distributed Sensor Networks*, 14:155014771875506, 01 2018. doi: 10.1177/1550147718755069. 2.1

[23] C. L. Hwang and K. Yoon. *Multiple Attribute Decision Making: Methods and Applications*. Springer-Verlag, New York, 1981. URL http://dx.doi.org/10.1007/978-3-642-48318-9. 3.2

[24] Hossein Karami, Antony Thomas, and Fulvio Mastrogiovanni. *Task Allocation for Multi-robot Task and Motion Planning: A Case for Object Picking in Cluttered Workspaces*, pages 3–17. 01 2022. ISBN 978-3-031-08420-1. doi: 10.1007/978-3-031-08421-8_1. 2.3

[25] G. Ayorkor Korsah, Anthony Stentz, and M. Bernardine Dias. A comprehensive taxonomy for multi-robot task allocation. *The International Journal of Robotics Research*, 32(12):1495–1512, 2013. doi: 10.1177/0278364913496484. 1.2

[26] Evelyn Kuo, Ananya Rao, Ian Abraham, and Howie Choset. On the optimality and robustness of ergodic search. 1.1

[27] Wenhao Luo and Katia Sycara. Adaptive sampling and online learning in multi-robot sensor coverage with mixture of gaussian processes. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6359–6364, 2018. doi: 10.1109/ICRA.2018.8460473. 2.1

[28] Wenhao Luo, Changjoo Nam, George A. Kantor, and Katia P. Sycara. Distributed environmental modeling and adaptive sampling for multi-robot sensor coverage. In *Adaptive Agents and Multi-Agent Systems*, 2019. 2.1

[29] George Mathew and Igor Mezić. Metrics for ergodicity and design of ergodic dynamics for multi-agent systems. *Physica D: Nonlinear Phenomena*, 240(4):432–442, 2011. ISSN 0167-2789. doi: https://doi.org/10.1016/j.physd.2010.10.010. 1.1, 2.2.1, 2.2.2, 2.2.2, 4.1, 4.2, 5.1

[30] Anastasia Mavrommati, Emmanouil Tzorakoleftherakis, Ian Abraham, and

Todd D. Murphey. Real-time area coverage and target localization using receding-horizon ergodic exploration, 2017. 2.2.1

[31] Lauren M. Miller and Todd D. Murphey. Trajectory optimization for continuous ergodic exploration. In *2013 American Control Conference*, pages 4196–4201, 2013. doi: 10.1109/ACC.2013.6580484. 2.2.1

[32] Lauren M. Miller and Todd D. Murphey. Optimal planning for target localization and coverage using range sensing. In *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 501–508, 2015. doi: 10.1109/CoASE.2015.7294129. 1.1

[33] Lauren M. Miller, Yonatan Silverman, Malcolm A. MacIver, and Todd D. Murphey. Ergodic exploration of distributed information. *IEEE Transactions on Robotics*, 32(1):36–52, 2016. doi: 10.1109/TRO.2015.2500441. 1.1, 2.2.1

[34] Robin Murphy, Satoshi Tadokoro, Daniele Nardi, Adam Jacoff, Paolo Fiorini, Howie Choset, and Aydan Erkmen. *Search and Rescue Robotics*, pages 1151–1173. 01 2008. doi: 10.1007/978-3-540-30301-5_51. 1.1

[35] Yijie Qin, Lei Fu, Dingxin He, and Zhiwei Liu. Improved optimization strategy based on region division for collaborative multi-agent coverage path planning. *Sensors*, 23(7), 2023. ISSN 1424-8220. doi: 10.3390/s23073596. URL https://www.mdpi.com/1424-8220/23/7/3596. 2.1

[36] Zhongqiang Ren, Sivakumar Rathinam, and Howie Choset. Multi-objective conflict-based search for multi-agent path finding. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8786–8791, 2021. doi: 10.1109/ICRA48506.2021.9560985. 2.3

[37] Zhongqiang Ren, Sivakumar Rathinam, and Howie Choset. Conflict-based steiner search for multi-agent combinatorial path finding. In *Proceedings of Robotics: Science and Systems (RSS '22)*, June 2022. 2.3

[38] Zhongqiang Ren, Akshaya Kesarimangalam Srinivasan, Bhaskar Vundurthy, Ian Abraham, and Howie Choset. A pareto-optimal local optimization framework for multiobjective ergodic search. *IEEE Transactions on Robotics*, pages 1–12, 2023. doi: 10.1109/TRO.2023.3284358. 1.1, 1.3, 3.1, 1, 4.2, 5.1

[39] Thomas Robinson, Guoxin Su, and Minjie Zhang. Multiagent task allocation and planning with multi-objective requirements. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '21, page 1628–1630, Richland, SC, 2021. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450383073. 2.3

[40] Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. Finding a "kneedle" in a haystack: Detecting knee points in system behavior. In *2011 31st International Conference on Distributed Computing Systems Workshops*, pages

166–171, 2011. doi: 10.1109/ICDCSW.2011.20. 5.1

[41] Yunfei Shi, Ning Wang, Jianmin Zheng, Yang Zhang, Sha Yi, Wenhao Luo, and Katia Sycara. Adaptive informative sampling with environment partitioning for heterogeneous multi-robot systems. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11718–11723, 2020. doi: 10.1109/IROS45743.2020.9341711. 2.1

[42] Vikas Shivashankar, Rajiv Jain, Ugur Kuter, and Dana Nau. Real-time planning for covering an initially-unknown spatial environment. 01 2011. 2.1

[43] Yonatan Silverman, Lauren Miller, Malcolm Maciver, and Todd Murphey. Optimal planning for information acquisition. pages 5974–5980, 11 2013. doi: 10.1109/IROS.2013.6697223. 1.1

[44] Jesus Suarez and Robin Murphy. A survey of animal foraging for directed, persistent search by rescue robotics. In *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, pages 314–320, 2011. doi: 10.1109/SSRR.2011.6106744. 1.1

[45] Jesus Suarez and Robin Murphy. A survey of animal foraging for directed, persistent search by rescue robotics. In *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, pages 314–320, 2011. doi: 10.1109/SSRR.2011.6106744. 1.1

[46] Navid Dadkhah Tehrani, Andrew Krzywosz, Igor Cherepinsky, and Sean Carlson. Multi-objective task allocation for multi-agent systems using hierarchical cost function. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12045–12050, 2022. doi: 10.1109/IROS47612.2022.9981071. 2.3

[47] Jian Xiao, Gang Wang, Ying Zhang, and Lei Cheng. A distributed multi-agent dynamic area coverage algorithm based on reinforcement learning. *IEEE Access*, 8:33511–33521, 2020. doi: 10.1109/ACCESS.2020.2967225. 2.1

[48] K. P. Yoon and C. Hwang. *Multiple attribute decision making*. 1995. 3.2

[49] Alexander Zelinsky, Ray A. Jarvis, Julian Byrne, and Shin'ichi Yuta. Planning paths of complete coverage of an unstructured environment by a mobile robot. 1993. 2.1

[50] R. Zlot, A. Stentz, M.B. Dias, and S. Thayer. Multi-robot exploration controlled by a market economy. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, volume 3, pages 3016–3023 vol.3, 2002. doi: 10.1109/ROBOT.2002.1013690. 2.3