Foundations of Epistemic Planning for Collaborative Multi-Robot Systems

---

A

Doctoral Dissertation

Presented to
the Faculty of the School of Engineering and Applied Sciences

The University of Virginia

---

in partial fulfillment
of the requirements for the degree

Doctor of Philosophy

by

Lauren M Bramblett

2024

APPROVAL SHEET

This
Dissertation
is submitted in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

Author: Lauren M Bramblett

This Dissertation has been read and approved by the examining committee:

Advisor:	Nicola Bezzo
Committee Member:	Lu Feng
Committee Member:	Laura Barnes
Committee Member:	Yen-Ling Kuo
Committee Member:	Erin Cherry

Foundations of Epistemic Planning for Collaborative Multi-Robot Systems

by

Lauren M Bramblett

M.S., Operations Research
Air Force Institute of Technology, 2019

B.S., Applied Mathematics
United States Air Force Academy, 2017

Abstract

Many robotic applications, such as search and rescue, disaster relief, and inspection operations, are often set in unstructured environments that typically have communication constraints. In such environments, a multi-robot system must either be deployed to remain constantly connected, sacrificing operational speed and efficiency, or allow disconnections considering when and how to regroup. In this thesis, we insist that the latter approach is desired to increase operational efficiency and create more robust and predictable reasoning within the multi-robot system during disconnection. However, planning in unstructured and unpredictable environments when communication is unavailable requires computing an intractable sequence of possibilities. To address these challenges, we propose a novel epistemic planning approach to propagate beliefs about the state of the system during communication loss to ensure cooperative operations. If changes occur at runtime, robots must understand the *social* aspect of a scenario to interact with the agents around them alongside achieving mission objectives and ensuring logical belief and planning updates. Planning actions socially requires a robot to infer the *intentions* and *beliefs* of other agents, *empathizing* to predict what other agents *want* and *know* about each other. The capacity to reason about the perspective of another agent is the foundation of "theory of mind" which enables the *"I know that you know that I know"* paradigm without the need for direct and constant communication among actors. Using this architecture, we can increase the operational effectiveness of multi-robot systems during disconnected operations, allowing robots to reason about the capability of other robots and plan according to local observations and simulated beliefs. The contribution of our approach includes: i) a dynamic rendezvous location and decision-making algorithm using risk estimations and multi-objective weighted sum optimization for faster information relay, ii) an epistemic planning formulation, formalizing beliefs and knowledge for consensus-based coverage while disconnected, iii) a generalized epistemic task assignment and gossiping protocol for complex multi-robot tasks with considerations for connectivity constraints and failures during operations and iv) an efficient runtime plan adaptation framework that leverages active inference to reason about the goals of others and signal to others their own knowledge and intentions in communication denied environments. We apply our contributions to both homogeneous and heterogeneous robotic systems, taking into account the various capabilities and limitations of each robot, such as failures, disturbances, dynamics, and sensing capabilities. Our contributions are validated through comprehensive simulations and experiments. Additionally, we include a discussion on incorporating reinforcement learning into our multi-robot epistemic planning framework, aiming for a more adaptable system that can handle runtime uncertainties and form dynamic optimal sub-teams, thereby making our approach scalable for large multi-robot systems.

"There is no secret ingredient. It's just you."
– Po, The Dragon Warrior

Acknowledgements

The job of PhD student has been one of the most enjoyable and painful experiences of my career. I thought that my undergraduate studies would be my last time in academia, but here I am! I would definitely not have made it without the support and guidance of my advisor Nicola Bezzo. Nicola has helped me manifest my hidden talent as a researcher and has fostered my interest in the field of robotics. I never thought in a million years that I would be writing academic papers as ~~good~~ well as I am able to after his tireless corrections and unending questions such as *how*, *what*, and most importantly *why*. His dedication to robotics and his unwavering commitment to his students have given me a foundation to develop into the researcher and individual that I am now. I will forever be grateful for the three years I have spent under his mentorship.

I would like to thank my dissertation committee members: Lu Feng, Laura Barnes, Yen-Ling Kuo, and Erin Cherry for their insightful and constructive feedback on my research, as well as for their support during the final stages of my PhD studies. I also thank Madhur Behl for serving as a committee member for my qualification exam in 2022.

I want to thank my former advisor, Dr. Lance Champagne, at the Air Force Institute of Technology for writing all of those tedious recommendation letters. I am also grateful to all the people who submitted my application and fought for me to have this opportunity even though it meant leaving my job early, especially Corrie Pecoraro and Matt Hart.

I would like to express my gratitude to current and former members of the AMR Lab – Esen, Rahul, Shijie, Jacob, Phil, Nick, Will, Patrick, Garrett – and Beatrice for being excellent lab mates and friends over the years. I have had a lot of good food and made many wonderful memories. It is an amazing team to be apart of.

To the wonderful and amazing people at Shenandoah Joe's Coffee Shop – especially Morgan: you kept me alive with your four-shot latte and Winston always enjoyed his pupper cups.

I would like to thank my parents Maureen, Jeff, and my sister Rachel for their persistent understanding, support, help, and love over the years. They have always been there for me during the highs and lows in life. Without them, I would not be the person that I have become.

Last but certainly not least, I would like to thank my husband Andy for not leaving me. I cannot thank him enough for his support and faith in me during the last three years of our separation

so that I could take advantage of this opportunity. He has endured my incessant complaints, put up with my endless ramblings, and continually offered insightful suggestions that kept me moving forward. From now on, he will have to tolerate my presence every day for the rest of our lives, but I am looking forward to it.

To fund my PhD research, I acknowledge: Northrop Grumman through the University Basic Research Program, the National Science Foundation (NSF) through grant #1816591, the Defense Advanced Research Projects Agency (DARPA) under Contract FA8750-18-C-0090, and CoStar under the Autonomous Building Condition Detection and Evaluation (ABCDE) grant.

Contents

List of Figures

List of Tables

List of Abbreviations

APF	Artificial Potential Field
DEL	Dynamic Epistemic Logic
ERX	Edge Recombination Crossover
GA	Genetic Algorithm
GAP	Generalized Assignment Problem
GS	Greedy Search
LTI	Linear Time-Invariant
MCTS	Monte Carlo Tree Search
MRS	Multi-Robot System
MRTA	Multi-Robot Task Allocation
mTSP	Multiple Traveling Salesman Problem
RHC	Receding Horizon Controller
SSI	Sequential Single-Item
ToM	Theory of the Mind
TSP	Traveling Salesman Problem
UAV	Unmanned Aerial Vehicle
UCT	Upper Confidence bound applied to Trees
UGV	Unmanned Ground Vehicle

Notation

Vectors

x_i	The ith element of a vector $\boldsymbol{x}$
$\|\boldsymbol{x}\|$	Euclidean norm of a vector $\boldsymbol{x}$

Matrices

$\boldsymbol{A}^\top$	Transpose of a matrix $\boldsymbol{A}$
$\boldsymbol{A}^{-1}$	Inverse of a matrix $\boldsymbol{A}$
$\boldsymbol{A} \otimes \boldsymbol{B}$	Tensor product of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$
$\boldsymbol{I}_n$	An $(n \times n)$–dimensional identity matrix
$\boldsymbol{A}_{ij}$	The element in the ith row and jth column of a matrix $\boldsymbol{A}$

Sets

$\mathbb{R}$	Set of real numbers
$\mathbb{R}^n$	Set of real n–dimensional vectors
$\mathbb{R}^{n \times m}$	Set of real $(n \times m)$–dimensional matrices
$\mathbb{R}_{>a}$	Set of real numbers greater than $a \in \mathbb{R}$
$\mathbb{R}_{\geq a}$	Set of real numbers greater than or equal to $a \in \mathbb{R}$
$\mathbb{N}$	Set of natural numbers (i.e., positive integers)
$\|S\|$	Cardinality of a set S
$S' \subseteq S$	Set S' is a subset of set S
$S \setminus S'$	The set S minus the elements of S'
$S \oplus_n s$	Adds element s to set S at index n

Epistemic Logic

AP	Set of atomic propositions
$K_i\phi$	Logical symbol denoting "agent i knows ϕ"
$B_i\phi$	Logical symbol denoting "agent i believes ϕ"
$s \otimes i : a$	Epistemic product update due to agent i executing an action a
$\triangleleft$	Logical notation for observation

Miscellaneous

$\mathbb{E}[x]$	Expected value of a random variable x
$\mathrm{Var}[x]$	Variance of a random variable x
$P(x)$	Probability of an event x occurring
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Graph $\mathcal{G}$ with vertices $\mathcal{V}$ and edges $\mathcal{E}$

Chapter 1

Introduction

Multi-robot systems (MRS) offer the potential to enhance efficiency, adaptability, and scalability in various tasks. With the reduction in cost of numerous robotic technologies, they are now utilized in many everyday activities such as warehouse order fulfillment and transportation, agricultural harvesting and crop monitoring, and increasingly within our homes for performing household chores or interacting with residents. Multi-robot systems can also be beneficial in critical applications such as search and rescue operations or the exploration of hazardous sites. As technology progresses, the use of multi-robot systems is becoming more widespread, although most research focuses on connected devices, assuming that all robots can stay connected or only experience brief disconnections. By deploying multi-robot systems in environments with limited communication, humans can explore remote areas, gather data from dangerous environments, and tackle complex challenges that would otherwise be impractical or hazardous.

In remote or inaccessible regions, multi-robot systems enable researchers to explore and collect data over a larger area. By distributing the workload and sharing information among robots, these systems can efficiently cover vast terrains, navigate complex obstacles, and gather diverse data sets. This collaborative approach facilitates efficient and comprehensive data collection. However, coordinating cooperation for multiple robots can be a challenging problem, particularly in dynamic and uncertain environments with limited or unavailable communication. In applications where long-range communication is often unreliable or unavailable, we note a current limitation in MRS research. Generally, MRS applications with communication constraints are high-stakes scenarios such as finding a stranded hiker in a remote location, recovering pieces of a downed aircraft in hostile territory, or rescuing survivors after a natural disaster. MRS have also been applied to scenarios with limited communication infrastructure such as subterranean pipeline inspection, marine sample collection, or extra-planetary exploration where range, terrain, and environment can inhibit signals from being sent or received by any entity [48, 89, 54].

Researchers have addressed part of the multi-robot limited communication challenge in several

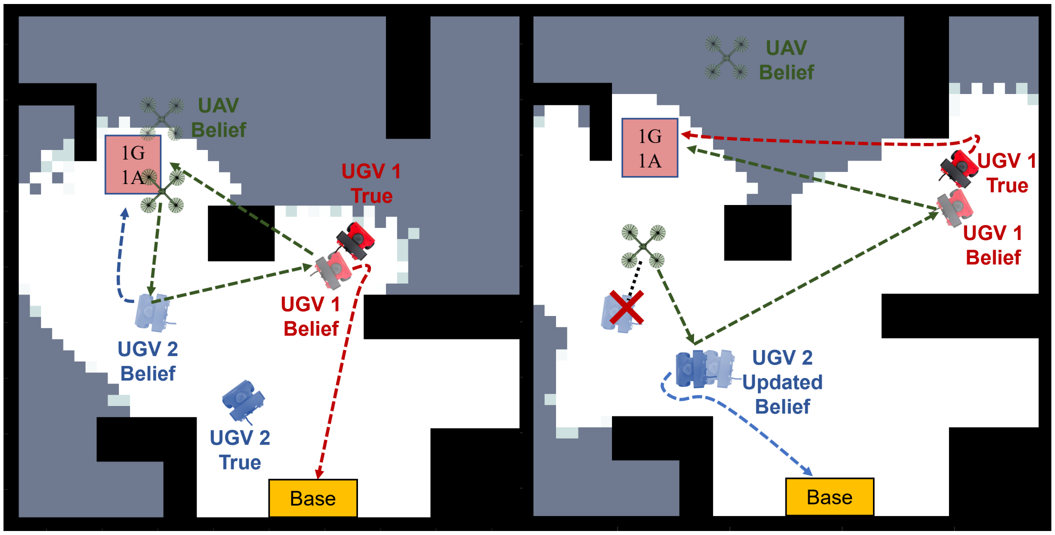


Figure 1.1: Pictorial depiction of the problem. The proposed framework enables a robot to reason from other agents' perspectives as it experiences a behavior change or observes that another robot is not where expected.

applications by predicting all possible future states. However, this method becomes impractical as the duration of disconnection increases. The widespread propagation of all potential states makes it difficult to make informed decisions. Moreover, there is limited research ensuring that the robots whose states are being predicted are cooperative and cognizant of the expectations of other robots in the system.

As humans, we cope with these constraints by implicitly reasoning about other actors' actions or beliefs while not communicating. A person may empathize with what another actor might believe in order to communicate and come to a shared understanding of the environment, as demonstrated in the Sally-Anne test [5] and [47]. Since the failures and uncertainties that occur typically happen without a priori knowledge and while disconnected, we would like to take advantage of local observations and construct a framework for robots to plan and communicate according to a set of higher-order beliefs while disconnected.

With these considerations in mind, the objectives of this work are to solve the following challenges:

- How to enable cooperative behavior of a multi-robot system in the absence of communication
- How to adapt a task allocation policy for a multi-robot system with no communication based on local information
- How to predict future states and state uncertainties of the multi-robot system with unforeseen changes

In order to accomplish these objectives, this work proposes a novel *epistemic planning* method for cooperative decision-making at runtime. Epistemic planning is an enriched automated planning with epistemic notions, including knowledge and beliefs, and allows for decision-making in multi-robot situations with distributed knowledge and capabilities, considering incomplete knowledge and beliefs about this knowledge. We couple theory of mind [71] with dynamic epistemic logic [56] to enable a multi-robot system to make informed decisions, ascribing *belief states* to other robots while disconnected, to coordinate their actions effectively. By understanding what they know and what information is unknown, robots can adapt their local plan and prioritize tasks, actions, and behaviors accordingly. This proposed work is easily extended to both homogeneous and heterogeneous robots by attributing the capabilities of a heterogeneous system (i.e., sensors, dynamics, abilities) to the epistemic plan and set of beliefs.

1.1 Related Literature

This section presents a review of related literature on multi-robot systems, particularly those addressing task allocation in scenarios with limited communication. Following this, an overview of methods for establishing a framework for deep reasoning in multi-agent systems is provided, focusing on planning based on shared beliefs and knowledge. Finally, research on multi-agent active inference is discussed, highlighting applications where intentions can be inferred without explicit communication.

1.1.1 Multi-robot Exploration & Task Allocation

Multi-robot exploration involves deploying a group of robots to systematically investigate and map an unknown or partially unknown environment. This field encompasses various strategies, algorithms, and coordination mechanisms that aim to optimize the collective efficiency and effectiveness of robots. Multi-robot exploration leverages the combined capabilities of multiple robots to achieve faster, more reliable, and more comprehensive exploration, with wide-ranging applications. Exploration and cooperation have been widely studied in multi-agent systems research [51, 25] considering many different communication strategies [1] over the years. Most works rely on constant and reliable point-to-point communication [87] or maintaining mesh networks [27] considering graph theoretical approaches to enable cooperative tasks. In contrast, [88] removes the communication expectation and accomplishes tasks with intermittent information relays.

Prior work has shown that frontier-based algorithms perform competitively with alternative approaches for exploration tasks [41]. Additionally, recent works show that using rendezvous to account for limited connectivity in multi-agent exploration can accomplish exploration as efficiently as traditional, all time connected frontier-based exploration. For example, [17] plans a team of UAVs based on a relay and explore a system which can decide to sacrifice its battery life to increase environmental information gain. Similar work was performed in [34] where a multi-agent heterogeneous team is assigned exploration or relay roles and plans meeting points based on data-gathering limitations. Tangentially, partitioning methods are used to autonomously de-conflict exploration goals amongst a team of robots in [22]. Partitioning is shown to increase the efficiency of team exploration, especially when disconnected as demonstrated in [14].

Despite the volume of work, less progress has been made on addressing the aspect of intentional disconnectivity and decision-based reconnectivity in an unknown environment that requires exploration and task relay. Authors in [35] substantiate an approach using data mules to notify a search and rescue team of a task location. The procedure assumes a single task in a bounded area and encourages connectivity as an objective in their genetic algorithm. In addition, [3] formulates a partially-observable Markov decision process that allows the robots to collaborate and make decisions based on a reward function; however, robots are penalized if moving to a disconnected location and need to remain connected to communicate upon finding a task. Other work thoroughly examines cooperative completion of one goal under limited communication [85, 39] and multi-objective approaches assume constant communication [73], [23].

A separate, but related, field to multi-robot coverage is multi-robot task allocation (MRTA). MRTA involves assigning tasks to a group of robots in a way that optimizes performance, efficiency, and coordination. The goal is to ensure that tasks are completed effectively while minimizing resource usage and time. Task allocation and planning are challenging problems that have attracted researchers from many different disciplines [43, 84, 38]. The Traveling Salesman Problem (TSP) [24], a well-studied problem in operations research, is often used to model the planning challenges encountered by a single robot. Later, this formulation was extended to include multiple vehicles (mTSP) in [6]. The mTSP is more suitable for large-scale applications but is more complex than the TSP. Several solutions have been proposed to solve this problem, such as the genetic algorithm (GA) [59, 28], which considers tasks that require specific vehicle types, and [20], which uses a consensus-based bundling algorithm for limited replanning, but few works have included communication restrictions and failures. An approach, presented in [66], utilizes an auction allocation algorithm to assign tasks but assumes enough locally connected robots to perform the assigned tasks. Other works, such as [32], address the issue of prolonged disconnections using rendezvous locations. However, this can lead to unnecessary communication and laborious backtracking. In these environments, robots may operate with outdated or incomplete information while also being aware of the possibility of misinformation. A robot may act on information that it believes to be accurate, only to discover later that it is outdated or incorrect. This can have significant consequences, particularly in critical applications such as disaster response or military operations. In [40], system failures are considered in the multi-agent policy search, but it is assumed that robots can communicate these disruptions. This dissertation explores an MRS framework designed to accommodate intentional disconnections while ensuring efficient task completion and exploration. Our approach is resilient to failures and adaptable to dynamic task additions, thus enhancing contributions to the field of multi-robot task allocation.

1.1.2 Theory of Mind & Epistemic Planning

Epistemic planning is a type of automated planning in artificial intelligence that deals with knowledge and beliefs of agents [7]. It involves creating plans that not only achieve physical goals but also ensure that agents acquire, maintain, or reason about knowledge and beliefs. As humans, we cope with communication constraints by implicitly reasoning about the actions of other actors while not communicating. A person may empathize with what another actor might believe in order to communicate and come to a shared understanding of the environment. Theory of Mind (ToM) and epistemic planning are closely related concepts in artificial intelligence and cognitive science [36]. ToM refers to the ability of an agent to attribute mental states, such as beliefs, desires, and intentions, to others, enabling it to predict and interpret their actions [2]. In psychology, theory of mind refers to the ability for one to "put themselves in another person's shoes" [15]. Integrating ToM into epistemic planning allows agents to anticipate and respond to the knowledge and beliefs of other agents, leading to more effective coordination and decision-making in multi-agent systems. The authors in [61] and [90] show that nested beliefs and reasoning in multi-agent planning can better equip agents to work in teams and show that this integration is crucial for applications requiring sophisticated interaction and collaboration among multiple intelligent agents. This dissertation characterizes theory of mind for a multi-robot system similar to [26] and employs epistemic planning as a logical mechanism to account for the system's knowledge and beliefs. Epistemic planning can adopt the perspectives of other robots within the system, reasoning about their knowledge and uncertainties, thereby preventing first-order reasoning deadlock. Previous multi-agent planners typically maintain separate knowledge bases for each agent in the scenario [78, 13]. However, these static first-order representations lack the expressiveness required for more complex scenarios involving nested perspective-taking [49] and when the environment or the system changes over time.

Dynamic Epistemic Logic (DEL) extends the concepts of epistemic planning and Theory of Mind by providing a formal framework to model and reason about changes in knowledge and beliefs over time [83]. While epistemic planning focuses on devising plans that consider the current epistemic states of agents, DEL specifically addresses how these states evolve through actions and observations [82]. DEL incorporates both epistemic actions, which affect what agents know, and public announcements, which can change the common knowledge among agents. DEL, as a result, is a more flexible representation of the dynamics of knowledge and belief, enabling adaptive planning in scenarios where agents must continuously update their understanding of the world and each other's mental states [21]. This makes DEL particularly useful for applications in multi-agent systems, communication protocols, and interactive environments where knowledge and belief are in constant flux. In multi-robot systems dynamic epistemic logic (DEL) [83], allows each robot in the MRS to reason and plan using its beliefs of other robots in the system while disconnected, updating its beliefs and policy if new events are observed, and routing to communicate when necessary. DEL has recently been integrated into robotics applications. The method presented in [8] recreates the Sally-Anne psychological test for human-robot interactions. Typical DEL-based multi-agent research uses epistemic planning for game theory-based policies [56]. In contrast to existing work, we evaluate the use of DEL for an MRS application, equipping each robot with the ability to reason about the state of the system, considering local observations. Our approach expands on previous mTSP research and shows that intermittent rendezvous and gossiping protocols will allow the MRS to reason about the system's state and share any new knowledge, updating its beliefs at runtime utilizing an epistemic planning framework in environments where communication is limited.

1.1.3 Active Inference

Active inference provides a probabilistic framework for agents to make decisions and update beliefs by minimizing uncertainty and surprise [29]. Using Bayesian inference, agents predict sensory inputs and select actions aligned with their goals, incorporating their own and others' knowledge and beliefs. This inherently involves Theory of Mind (ToM), as agents model and anticipate others' mental states for effective interaction [70]. Integrating active inference with ToM and epistemic planning enables sophisticated planning and decision-making, allowing agents to refine their understanding and actions for optimal outcomes in complex, multi-agent environments. This connection is essential for developing intelligent systems capable of adaptive and cooperative behavior in uncertain and dynamic settings. This dissertation explores a method of active inference that aims to improve reasoning about the environment to reduce uncertainty about the operational environment by using higher-order reasoning.

Previous work on active inference in [80] utilized Policy Belief Learning (PBL) to improve conveyance of intent through action, paired with a reward system, which incentivized actions which improve overall system understanding of the operational environment. Additionally, [75] evaluated how active inference could be performed in robot teams which can typically communicate and how the lack of communication can be exploited to better decrease uncertainty in an agent's beliefs of the world. Previous work was also performed in both leader-follower and leaderless models in [53] of which the similar approach to their leaderless work was used in our approach. Few works have incorporated realistic applications, including the authors in [68] who use active inference and behavior trees to improve the robustness of plans for a mobile manipulator. Additionally, authors in [16] showed that perception, path generation, localization, and mapping naturally emerge from using active inference and minimizing free energy.

Recently, a new perspective on the theory of mind (ToM) called the Bayesian brain has attracted attention, being suggested as a feedforward model for decision-making [69]. In this dissertation, we extend the concept of the Bayesian mind using active inference as in [72] and [53], and incorporate dynamic epistemic tasking with higher-order reasoning. We demonstrate that while first-order reasoning can yield good results, higher-order reasoning provides more robust outcomes even in the absence of communication and in the presence of noisy observations.

1.2 Overview of Research

The research presented in this dissertation consists of three successive Parts that include: I) multi-robot exploration and task allocation in communication restricted environments, II) epistemic planning for multi-robot rendezvous and task allocation, and III) online epistemic planning and inference for multi-robot systems. A final Part IV summarizes what we have gained and learned from in the first three Parts. Figure 1.2 provides an overview of the research presented in this dissertation.

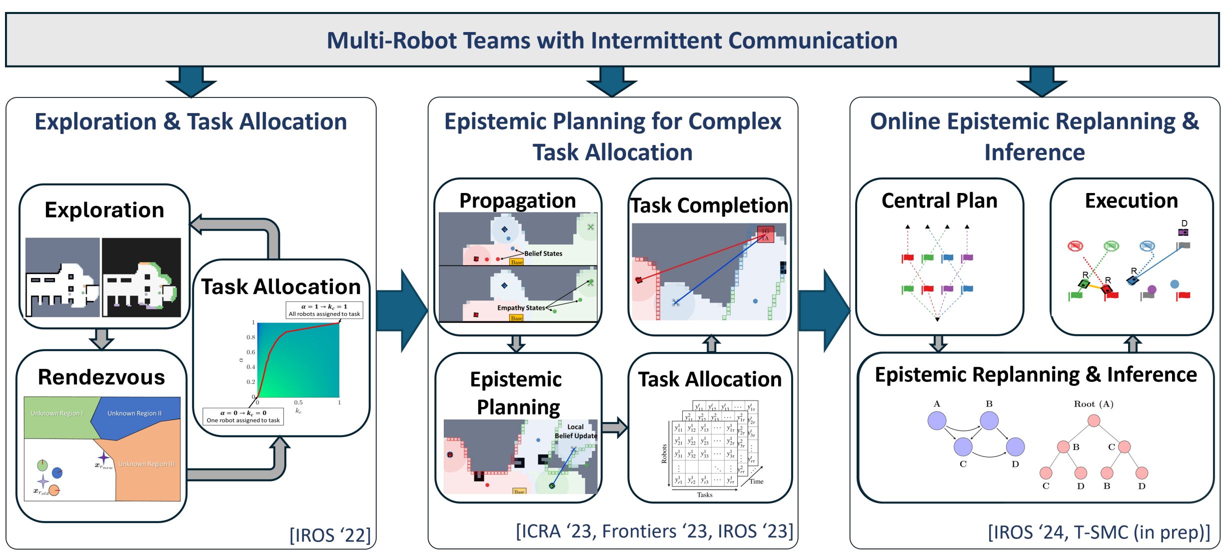


Figure 1.2: Overview of the presented research in this dissertation.

Beginning with **Part I**, we present a framework for efficiently accomplishing all tasks in the environment considering dynamic task discovery and intermittent communication. Exploration is achieved via a Sobel edge detection frontier algorithm that enables navigation of unknown complex (both convex and non-convex) environments. Once a task is discovered, a multi-objective weighted sum optimization method is proposed for allocating tasks based on prioritization and expectation estimation. The multi-robot system is able to effectively handle rendezvous and task allocation even in a limited communication environment. In **Part II**, we present a solution to the backtracking problem in Part I, where previously explored regions were used as rendezvous points. In this Part, we formulate a framework for a multi-robot system to explore, allocate tasks, and rendezvous in a partially unknown environment. In particular, we include an epistemic notion in planning, a robot may enact depth-of-reasoning about the state of the system, analyzing its beliefs about each robot in the system. In this method, a set of possible beliefs about other robots in the system is propagated using a frontier-based planner to achieve the coverage objective. As disconnections occur, each robot tracks beliefs about the system state and reasons about multiple objectives: i) coverage of the environment, ii) dissemination of new observations, and iii) possible information sharing from other robots. In **Part III** we focus on the following question: *How can we ensure cooperative and efficient behavior for task allocation when a centralized predefined plan must change at runtime?* This question is an expansion of our work in Part II, allowing for the elimination of certain limiting suppositions and making the work more suitable for practical scenarios where tasks are known but unforeseen changes in the environment or MRS may occur. Our proposed solution includes a centralized mission planner that accounts for intermittent rendezvous, promoting the discovery of failures and inefficiencies in the MRS if something does not go according to plan, and an efficient runtime plan adaptation that leverages our epistemic planning framework to reason about the likely knowledge and intentions of others based on the current epistemic state and dynamically reassign tasks. Our proposed framework enables MRS to cooperate, given limited communication and an uncertain operating environment. Before concluding Part III, we also include our preliminary work on higher-order reasoning for multi-robot systems without explicit communication. In this work, we introduce the concept of epistemic planning and deep reasoning in an active inference framework to allow a multi-robot system to empathize with the perception of other robots and infer task assignments. Throughout Parts I-III in this dissertation, we feature our contributions with simulation results using realistic system dynamical models and lab experiments on real unmanned ground and aerial vehicles to validate our frameworks. Finally, in **Part IV**, we conclude the dissertation by providing insight in what we have accomplished and learned, followed by a discussion on possible directions we could take in the future that we could take.

1.3 Dissertation Organization and Contributions

In this section, we outline the structure of this dissertation by summarizing each chapter and detailing the contributions within them. To summarize this dissertation, Part I includes Chapter 2, which emphasizes multi-robot exploration in unknown environments with limited communication. Part II encompasses Chapters 3 and 4, in which we delve into epistemic planning and decentralized task allocation for multi-robot systems. Part III is represented in Chapters 5 and 6, which discuss online adaptations using higher-order reasoning to reallocate tasks and infer beliefs and task assignments. Finally, Part IV concludes this dissertation by summarizing our results and discussing potential future work. Let us now outline each chapter and highlight our contributions.

Chapter 2: Coordinated Multi-Agent Exploration, Rendezvous, & Task Allocation

In this chapter, we introduce a comprehensive planning framework designed to facilitate cooperative behavior among robots without requiring continuous communication. This approach shows significant enhancements in task completion and coverage time when compared to fully connected robotic networks and commonly used frontier-based exploration techniques. The proposed framework specifically addresses exploration by encouraging separation and disconnection, rendezvous for reconnection and information sharing, and task allocation based on prioritized objectives. Exploration is facilitated through a Sobel edge detection frontier algorithm, which allows for navigation in unknown and complex environments, whether convex or non-convex. Upon discovering a task, a multi-objective weighted sum optimization method is employed to allocate tasks based on their prioritization and expected outcomes. The key contributions of our approach are twofold: i) a dynamic rendezvous location and decision-making algorithm that utilizes a risk variable to expedite information relay, and ii) a multi-objective weighted sum optimization method for task allocation that prioritizes both exploration and task completion. The aim of this work is to efficiently coordinate efforts to complete all tasks and thoroughly explore the entire environment. This chapter is based on the publication:

- L. Bramblett, R. Peddi, and N. Bezzo, "Coordinated multi-agent exploration, rendezvous, & task allocation in unknown environments with limited connectivity," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2022, pp. 12 706–12 712.

Chapter 3: Implicit Coordination for Multi-Robot Teams in Communication Restricted Environments

In this chapter, we consider that in communication restricted environments, a multi-robot system can be deployed to either: i) maintain constant communication but potentially sacrifice operational efficiency due to proximity constraints or ii) allow disconnections to increase environmental coverage efficiency, challenges on how, when, and where to reconnect (rendezvous problem). Specifically, we address the second issue and observe that the majority of advanced methods presume that robots can follow a prearranged plan; nevertheless, system malfunctions and alterations in environmental conditions can lead to deviations from the plan, resulting in cascading impacts throughout the multi-robot system. This chapterintroduces a coordinated framework for epistemic prediction and planning aimed at achieving consensus without direct communication for purposes such as exploration, coverage, task discovery, completion, and rendezvous. The core element of this framework is dynamic epistemic logic, which enables robots to share belief states and understand other agents' perspectives. The propagation of belief states and subsequent environmental coverage are accomplished using a frontier-based method within an artificial physics-based framework. The effectiveness of the proposed framework is demonstrated through both simulations and practical experiments involving unmanned ground vehicles in various cluttered settings. The work in this chapter is based on the following publication:

- L. Bramblett, S. Gao, and N. Bezzo, "Epistemic prediction and planning with implicit coordination for multi-robot teams in communication restricted environment," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023.

Chapter 4: Epistemic Planning for Complex Task Allocation

This chapter addresses the addition of dynamic tasks and the occurrence of failures during multi-robot system operations. In scenarios such as search and rescue or disaster relief, unreliable communication often leads to inefficiencies or complete failures in most MRS algorithms. Failures and environmental uncertainties can have cascading effects throughout the system, particularly when the mission objective is complex or time-sensitive. To address this, we introduce an epistemic planning framework in this chapter that enables robots to reason about the system state, utilize heterogeneous system configurations, and optimize information dissemination to disconnected neighbors. Dynamic epistemic logic formalizes the propagation of belief states, and epistemic task allocation and gossip are achieved through a mixed integer program that uses the belief states for utility predictions and planning. The proposed framework is validated using simulations and experiments with heterogeneous vehicles. This chapter is based on the following publications:

- L. Bramblett and N. Bezzo, "Epistemic planning for multi-robot systems in communication restricted environments," *Frontiers in Robotics and AI*, vol. 10, p. 67, 2023.
- L. Bramblett and N. Bezzo, "Epistemic planning for heterogeneous robotic systems," in *2023 IEEE International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2023.

Chapter 5: Online Epistemic Replanning for Multi-Robot Systems

Chapter 5 extends our previous work on epistemic planning to the multiple traveling salesman problem under uncertainty and limited connectivity. A large part of MRS research assumes that the system can maintain communication through proximity rules and formation control, or by establishing a framework that permits separation and adherence to a prearranged plan during long periods of disconnection. If robots cannot communicate intermittently during operations, any failures in the MRS may go unnoticed, which can be harmful to a complex or urgent mission. To address this issue, our proposed framework consists of two primary phases: i) a centralized planner that allocates mission tasks by incentivizing periodic rendezvous between robots to mitigate the impact of unforeseen events during mission execution, and ii) a decentralized replanning approach using epistemic planning to formalize belief propagation and a Monte Carlo tree search for policy optimization based on distributed rational belief updates. The proposed framework surpasses a baseline heuristic and is validated through simulations and experiments with aerial vehicles. This chapter is based on the following publication:

- L. Bramblett, B. Miloradovic, P. Sherman, A.V. Papadopoulos, and N. Bezzo, "Robust Online Epistemic Replanning of Multi-Robot Missions," *2024 IEEE International Conference on Intelligent Robots and Systems (IROS).*

Chapter 6: Active Epistemic Inference for Task Allocation in Multi-Robot Systems

In this chapter, we design a frameworkwhere a team of autonomous robots must complete tasks in the environment without communication. The robots are equipped with the ability to use higher-order reasoning and infer the goals of other agents while signaling their own intent to the other robots in the system. Some methods have addressed this problem by utilizing a theory of mind (ToM) framework, but typically only allow agents to use first-order reasoning about observations. In contrast, to deal with this problem, our proposed framework has two main phases: i) efficient runtime plan adaptation using active inference to signal intentions and reason about a robot's own belief and the beliefs of others in the system, and ii) hierarchical epistemic planning framework to iteratively reason about the current MRS mission state. The proposed framework outperforms a baseline heuristic and is validated using simulations and virtual experiments with unmanned ground vehicles. This chapter is based on the following current work:

- L. Bramblett, J. Reasoner, and N. Bezzo, "Active Epistemic Inference for Task Allocation in Multi-Robot Systems," *in preparation for submission to the IEEE Transactions on Systems, Man, and Cybernetics: Systems.*

Chapter 7: Conclusions and Future Work

In this chapter, we conclude the dissertation by summarizing the results from all the aforementioned works and discuss potential future directions to build on.

1.4 Summary of Contributions

To summarize, the work presented in this dissertation will contribute to the existing state-of-the-art research of multi-robot systems in communication restricted environments and presents new ideas utilizing epistemic planning in multi-robot applications:

- A dynamic rendezvous location and decision-making algorithm using a risk variable for faster relaying of information and a multi-objective weighted sum optimization method for task allocation based on weighted prioritization of exploration and task completion.
- An epistemic planning formulation using dynamic epistemic logic providing generalized assignment and belief propagation for coverage of an environment with considerations for connectivity constraints and team member dynamics.
- A formal homogeneous and heterogeneous task assignment and gossiping procedure using epistemic planning for complex tasks considering disturbances and uncertainties in a communication restricted environment.
- An epistemic planning-based framework applied to the multiple traveling salesman problem under communication constraints, adapting at runtime for system disturbances and failures.
- A continuous epistemic planning approach for multi-robot systems that uses active inference to inform a task assignment at runtime and adapts to observations in the environment.

Part I

Multi-Agent Systems in Unknown Environments with Limited Connectivity

Chapter 2

Coordinated Multi-Agent Exploration, Rendezvous, & Task Allocation

This chapter introduces an innovative framework for coordinated exploration, rendezvous, and task allocation. We utilize a Sobel edge detection frontier-based approach, assigning each robot a unique exploration task. As robots explore, they may discover new tasks and are capable of reasoning about these tasks using a weighted multi-objective optimization algorithm. Robots that do not find new tasks meet to update their exploration assignments and are also given new tasks communicated by other agents or are assigned to find agents that do not rendezvous. We show that our method outperforms those lacking a reasoning step or where agents explored the environment in formation. The material covered in this chapter was published in the following:

- L. Bramblett, R. Peddi, and N. Bezzo, "Coordinated Multi-Agent Exploration, Rendezvous, & Task Allocation in Unknown Environments with Limited Connectivity," *in 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).*

2.1 Introduction

The presence of a robust communication infrastructure within the multi-robot system is typically considered the key enabler for such cooperative behavior. However, we notice that maintaining reliable communication may prevent the rapid exploration of an environment. As human beings we are constantly dealing with this trade-off and do not rely on constant communication with other actors to perform tasks. Instead, we typically take a "divide and conquer" approach, splitting to better cover an area, and coming back to rendezvous points if/when needed to share information. In this chapter, we insist that to enable efficient multi-robot operations, it is necessary to balance priorities for high-stakes scenarios and locally adapt based on the current environment while assuming limited communication within teams.

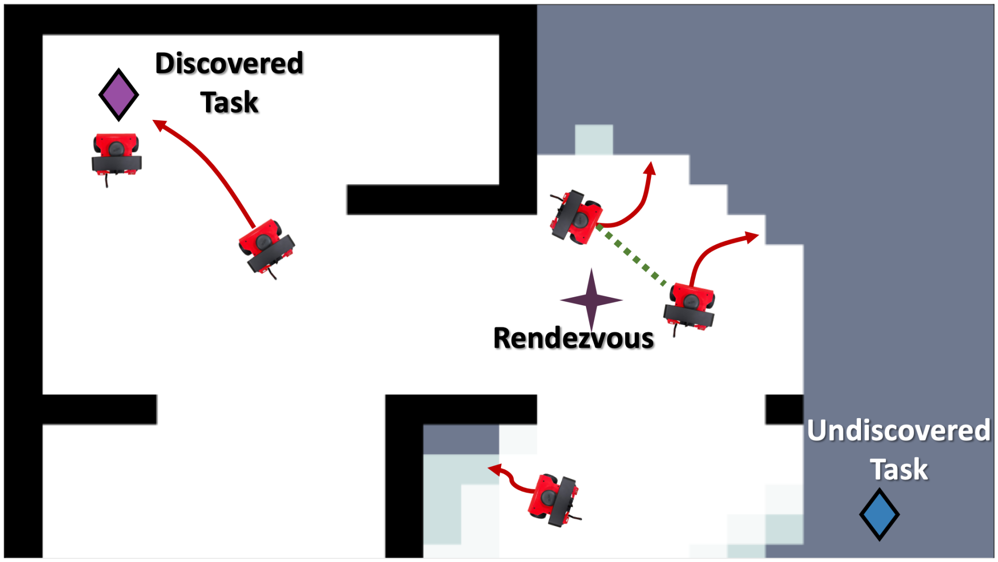
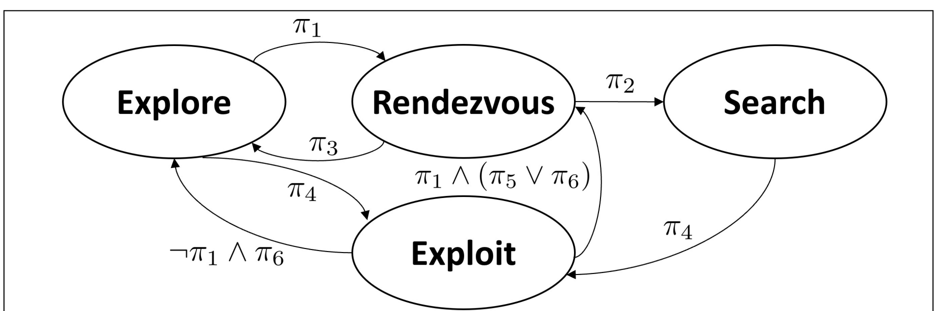


Figure 2.1: Pictorial depiction of the problem in this chapter in which a group of robots are tasked to explore an unknown environment while discovering and accomplishing tasks at unknown locations. Robots intentionally disconnect to improve exploration, reconnecting at rendezvous locations, dynamically decided at runtime, to share information and seek assistance.

Consider the scenario shown in Fig. 2.1 where robots are canvasing an area for tasks. The trajectory of maximum information gain is for all agents to move inverse to each other. As traversal to decrease entropy continues, communication becomes less reliable and tasks may be encountered. An intelligent agent will assess the utility of actions such as meeting at a specific point to recruit additional help and share information or staying to work on the task. The decision of one agent in the system will affect how other agents perceive the utility of any action in the environment.

Translating these intuitions into an algorithmic framework results in more efficiently coordinated AMRs that make independent decisions to cooperatively complete tasks.



$\boldsymbol{\pi}_1$: exploring time expired	$\boldsymbol{\pi}_4$: task found
$\boldsymbol{\pi}_2$: agent allocated to task	$\boldsymbol{\pi}_5$: locally optimal to rendezvous
$\boldsymbol{\pi}_3$: agent allocated to explore	$\boldsymbol{\pi}_6$: task complete

Figure 2.2: The proposed framework with event definitions.

Our framework, depicted in Fig.2.2, consists of four behaviors: *explore, rendezvous, search, and exploit*. All robots begin in the *explore* state and are initially all connected, hence in communication range with each others. Specifically, a robot in:

1. **Explore:** decreases environmental entropy by moving to unmapped locations

2. **Rendezvous:** meets the team at a common dynamically selected location to share information

3. **Search:** locates a robot who did not rendezvous or travels to a found task

4. **Exploit:** completes discovered tasks.

Tasks discovered by any member of the team will impact the assigned objectives after rendezvous of a robot in the search or exploit states.

Transition between each behavior is defined by several key events. First, robots are initially tasked to explore the environment, maximizing information gain while promoting disconnection. Once a task is discovered, an agent can either stay at the task location and exploit the task or rendezvous to seek help from team members. Rendezvous occurs after a predetermined time and rendezvous points are computed dynamically to promote efficient information sharing. The rendezvoused team will allocate tasks based on a task allocation policy. After allocation, the robots move to their assigned objectives of either exploration of the environment or completion of tasks. The scenario is complete once there are no more available frontier points and all discovered tasks are complete.

2.2 Preliminary Modeling

This section introduces the dynamical, system set notation, communication, and task models used throughout this chapter. Let us consider K robots identified by a unique value in the set $\mathcal{K} = \{1, \ldots, K\}$. In addition, we assume N stationary targets identified by a unique value in the set $\mathcal{N} = \{1, \ldots, N\}$, placed at an unknown location. Tasks are completed at a pace proportional to the number of robots present at the task location with an a priori expected completion time, μ_L. Real-world applications of such task include for example manipulating an item or inspecting a complex object. The environment, $\mathcal{W}$, is partitioned by m square cells as an occupancy map, M, where $M \subseteq \mathbb{R}^2$.

To represent a communication constrained environment, we utilize a disc constraint for sharing information between robots. A robot can communicate with any other robot within a fixed maximum distance, independently of the presence of obstacles. The robots employ multi-hop connectivity such that any robots not within range connected by another member of a team that is within range may share information. Robots connected to any k robot are evaluated through connected components in graph $G = (\mathcal{K}, E)$.

$$E \subseteq \{\{i,j\}|\ i,j \in \mathcal{K} \text{ and } i \neq j \text{ and } d_{ij} \leq d_c\} \tag{2.1}$$

where E is the set of connected edges between robots i and j where the distance between i and j, d_{ij}, is less than or equal to the communication range, d_c. Robots i and j can also communicate when there exists a path from i to j in G.

For ease of discussion, each robot is modeled as a double integrator; however, the proposed strategy is independent on the dynamics of the robots, and, although terrain, ground clutter, and elevation may vary from cell to cell, we assume they do not have any significant impact on motion, communication, or search.

2.3 Problem Formulation

In this chapter, we consider a scenario in which robots in a multi-robot system coordinate their motion in a decentralized fashion to explore and cover an unknown, cluttered environment while finding and accomplishing tasks at unknown locations scattered throughout the environment. There are several challenges that arise to allow such cooperative behavior. In particular, the agents in the team need to: 1) explore an unknown environment quickly and efficiently, disconnecting from each other at times, and 2) once a task is found, decide if and how to seek help from other team members not in communication range. A successful framework is applicable to any autonomous mobile system, any number of robots, any number of tasks, and any environment. We decouple the research question into the following three sub-problems.

Problem 1: ***Exploring unknown environments:*** Consider the task of exploring an unknown, bounded area using a multi-robot system with K agents, find a policy for each robot to maximize information gain by mapping unique portions of the environment in the minimum amount of time. Assuming all cells in M are traversable, we can cast this problem as:

$$\min \quad \left| \bigcap_{k \in \mathcal{K}} \mathcal{V}_k \right| \tag{2.2}$$

$$\text{s.t.} \quad \bigcup_{k \in \mathcal{K}} \mathcal{V}_k = M \tag{2.3}$$

where $\mathcal{V}_k$ is the set of visited cells for robot k.

While exploring, tasks may be encountered when a robot is outside communication range. Considering this constraint, we introduce the following problem statement.

Problem 2: ***Task relaying:*** Given that a robot has found a task and is disconnected from the rest of the team, find a rendezvous policy, $R_\phi(t)$, that minimizes task completion over a time interval.

$$R_\phi(t) = \arg\min_{\phi} \sum_{t_l=t}^{t+\mu_n} C_n(t_l, \phi) \tag{2.4}$$

where μ_n is the task length for task n and $C_n(t_l, \phi)$ is the remaining task length at time t_l given the rendezvous decision, ϕ. A robot may stay and accomplish the task instead of rendezvous. It is assumed that any robot that does not rendezvous has discovered a task. Under these assumptions, we introduce the third problem.

Problem 3: ***Task allocation:*** To generalize task allocation and environmental exploration, a policy, A, must be implemented that can address any combination of rendezvous results. Given prioritization of exploration and task completion while considering that a subset of robots, $K_a \subseteq K$, have rendezvoused, find a strategy to allocate robots to any discovered tasks such that

$$A_\psi = \underset{\psi}{\operatorname{argmin}}\ \alpha f(\psi) + \beta g(|K_a| - \psi, N_d) \tag{2.5}$$

where N_d is the number of discovered tasks, f is the reward function for allocating ψ robots, and g is the reward function for assigning the remaining robots, $|K_a| - \psi$, to task completion. The prioritization parameters are α and β for exploration and task completion, respectively.

2.4 Approach

In this section, we present a framework for exploration, rendezvous, and task allocation that enables robots to process local observations in a decentralized way before exchanging information with the rest of the multi-robot system. We include scheduled rendezvous, a frontier-based exploration technique, and a weighted sum optimization function for task distribution.

2.4.1 Cooperative Exploration

At the core of this framework, we require a distributed method for exploration. Tasks of the exploration strategy must include: a mapping procedure to characterize elements of the map and a method for motion planning to areas that decrease environmental entropy. Any deterministic exploration algorithm can be implemented in the proposed framework; in this chapter we implement a frontier-based edge detection algorithm which is complete, simple, and efficient.

To address the first element of a multi-robot exploration strategy, we leverage an occupancy grid to partition the map into cells where each cell, c, in the occupancy map m retains a probability of occupation. Given that all robots are equipped with some range sensor (e.g., lidar, sonar, cameras) a measurement h is a member of the set $\{0, 0.5, 1\}$ to denote a cell as free, unknown, or occupied, respectively. Recursive Bayesian estimation is employed to update each cell for a given measurement [4]. We also employ Sobel edge detection which allows efficient, simple processing of edges in noisy images [31] that results from the map update. To organize motion to the frontier of free and unknown cells, we apply such algorithm with two processing steps. First, the local approximated gradient of each cell in the occupancy map is found using the Sobel operator. This calculates the rate of change between the known and unknown regions where any value greater than a threshold is considered a frontier point. Second, we remove frontier points that coincide with a known obstacle.

In tandem with the exploration procedure, to encourage distinctive exploration assignments, k-means clustering is implemented to partition the unknown environment into K clusters (one for each robot) each time all agents are able to communicate. The clusters are auctioned in a centralized manner. Each robot submits its bid based on its distance to the j^{th} cluster centroid location, $\boldsymbol{c}_j$. Ties are broken by the robot's index value.

Noting that every unknown cell in the occupancy map has a label associated with one of the K centroids, we denote the centroid coordinate belonging to robot k as $\boldsymbol{c}_k$ and the set of cells belonging to that centroid as ζ_k. The cost associated with each frontier point is defined as:

$$F_{kp} = \begin{cases} ||\boldsymbol{f}_p - \boldsymbol{x}_k|| + \Delta||\boldsymbol{f}_p - \boldsymbol{c}_k|| & \boldsymbol{f}_p \notin \zeta_k \\ ||\boldsymbol{f}_p - \boldsymbol{x}_k|| & \boldsymbol{f}_p \in \zeta_k. \end{cases} \tag{2.6}$$

where Δ is a constant multiplier for a frontier point chosen outside of a robot's partition. We incentivize travel toward any robot's cluster by penalizing a frontier choice far away from its assigned partition. This encourages motion toward a robot's assignment if no frontier point is available within the partition. The index of the best frontier point is shown as:

$$\boldsymbol{f}_{k,p^*} = \underset{p}{\operatorname{argmin}}\ F_{k,p} \tag{2.7}$$

where $\boldsymbol{f}_{k,p^*}$ is the waypoint for robot k.

Fig. 2.3 shows the steps for the frontier-based approach for the map shown in (a), (b) shows the result of the Sobel edge detection algorithm, (c) plots the set of available frontier points, and (d) shows the resultant path to the chosen frontier point using an A* path planning algorithm.

While exploring, agents may experience fortuitous interactions if frontier points are similarly located. To ensure effective exploration, we allow for robots to share occupancy maps when within communication range and for agents to reevaluate current motion plans if an area has already been explored. Robots also communicate the last shared locations and occupancy map for all robots in set $\mathcal{K}$. For reduction in complexity, we assume that robot position uncertainties are negligible when

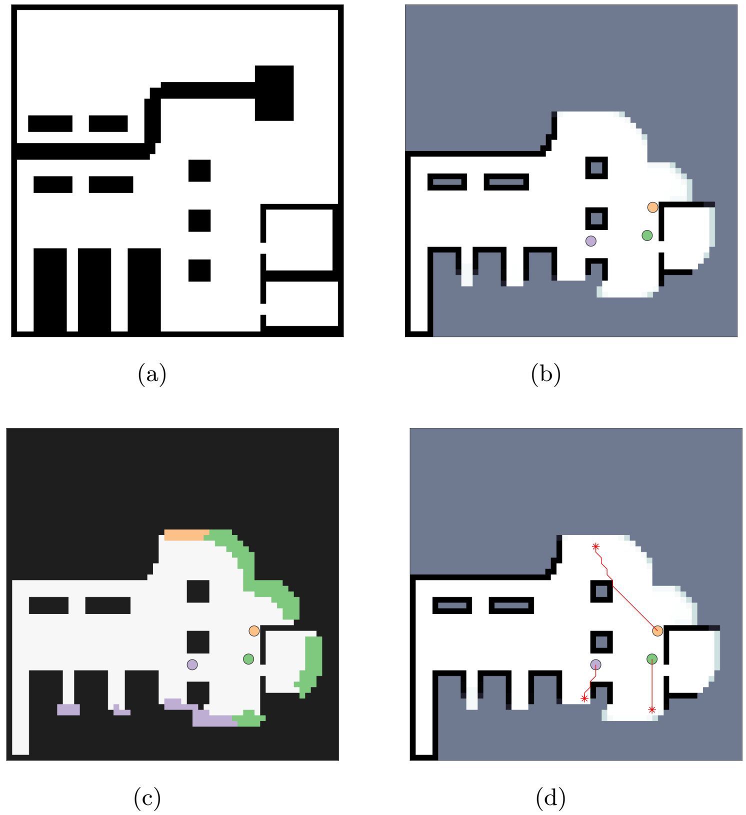


Figure 2.3: Overview of the exploration procedure with: (a) the warehouse occupancy map, (b) the portion of the map that has been explored (white), (c) the filtered frontier result using Sobel edge detection and colored by each agents partition, and (d) the resultant path to the frontier point chosen using the frontier point selection equation, (2.6).

building a map; alternatively, distributed SLAM algorithms can be used to account for uncertainties when map sharing [33].

Similarly, intentional sharing is desirable to decrease redundant coverage of a map. We introduce the rendezvous behavior as a deliberate method for sharing information and optimizing coordination amongst team members.

2.4.2 Rendezvous

To share information and any discovered tasks, the team will rendezvous. A rendezvous point is constructed when all agents are connected. The goal of the rendezvous point, $\boldsymbol{x}_r$, is to minimize a weighted travel distance for all robots such that

$$\boldsymbol{x}_r = \underset{r}{\operatorname{argmin}} \frac{\sum_{j=1}^{k} |\zeta_j| \; ||\boldsymbol{x}_r - \boldsymbol{c}_j||}{\sum_{j=1}^{k} |\zeta_j|}. \tag{2.8}$$

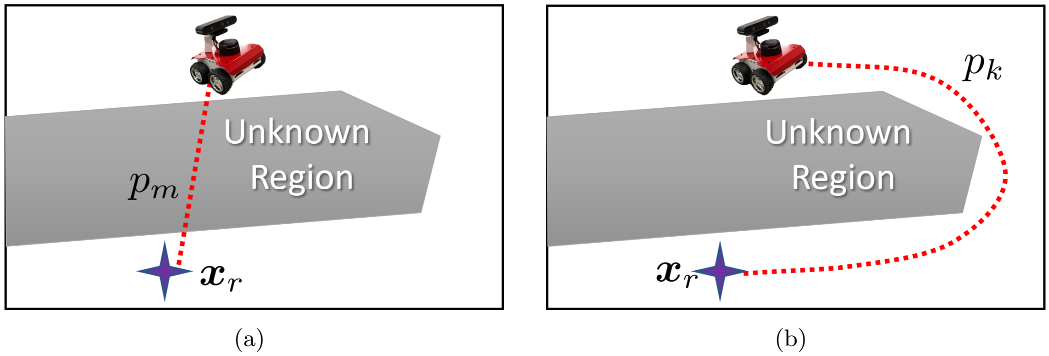


Figure 2.4: (a) shows R_ϕ through an unknown region when $\gamma \to 0 \Rightarrow \phi = 1$ and (b) shows R_ϕ through a known region when $\gamma \to 1 \Rightarrow \phi = 0$

The location is weighted according to the cardinality of the centroid's partition and is validated given the current known occupancy map. If the location is occupied or unknown, Moore's neighborhood [18] is used to check surrounding cells until a valid rendezvous point is found.

The point, $\boldsymbol{x}_r$, serves as the meeting location after a constant, pre-defined number of discrete steps, τ_e, have executed in the exploration state or the event denoted as π_1 in Fig. 2.2. Time-optimal rendezvous is not addressed in this chapter, but is considered for future work.

The safest rendezvous procedure is through known portions of the map where there are no spanning obstacles or dead-ends; however, the fastest route back to the rendezvous location may be through unknown portions of the occupancy map. To encourage economical rendezvous behavior, we introduce the following rendezvous policy:

$$R_\phi = \min_{\phi} \phi \left(p_m + \gamma n_{unk}\right) + (1-\phi) p_k \tag{2.9}$$

where R_ϕ is the rendezvous policy decision, $\gamma \in [0, 1]$ is a user-defined risk coefficient for traveling into unknown parts of the environment, p_m is the length of the potential shortest path to the rendezvous point, n_{unk} are the unknown segments of p_m, p_k is the length of planned path where all segments are known and unoccupied, and ϕ is the binary decision variable to rendezvous using p_m or p_k. Fig. 2.4 shows an example of two different paths depending on the risk coefficient γ.

Any robot who path plans using p_m may discover an expansive obstacle and risk not being able to rendezvous if the time to meet threshold has been exceeded. The robot that fails to rendezvous, continues exploring, but likely executes redundant exploration without knowing what portions of the map have already been explored by the rest of the robots.

2.4.3 Coordinated Task Allocation

Once a robot has localized a task (π_4), it must decide if and how to seek help from others (π_5). Similar to (2.9), the quickest route to rendezvous and share information may traverse unknown portions of the map, so it may be prudent to stay and complete the task if no help is necessary. The corresponding policy representation is an expectation estimation of the opportunity cost of rendezvousing versus remaining at the task.

$$R_\psi = \min_{\psi} \psi\,(p_m + \gamma n_{unk}) + (1-\psi)(p_e + \mu_\nu - \mu_L) \tag{2.10}$$

where, besides the elements already defined in (2.9), μ_L is the global expected path length, μ_ν is the actual task length, p_e is the length of path originally taken in the exploration state, and ψ is the binary decision variable to stay at the task or rendezvous.

In parallel, the rendezvoused team must decide how many robots to allocate to the found tasks given a prioritization for entropy versus task completion. As a baseline, all agents share their current occupancy maps, and any agent who localize a task and rendezvous will share the current task status and task location.

Tasks associated with agents who did not rendezvous are still considered in the policy optimization. Their task length is assumed to be the expected task length (μ_L) minus the time the robot has spent accomplishing the task while others were rendezvousing. The set L contains all expected tasks lengths for each $n \in N_d$ discovered tasks.

Coupled with task completion, we also consider that the policy must balance the entropy objective or the change in map entropy over the allotted exploration time. The amount of information gained over a path y given map m with position $\boldsymbol{x}$ and measurements h is represented as

$$\Delta H(y, m|\boldsymbol{x}, h) = \int_y p(y|\boldsymbol{x}, h) H(m|y, \boldsymbol{x}, h) dx \tag{2.11}$$

Since the environment is unknown, we make the simplifying assumption that one unit of information will be gained for every time step that an agent is exploring while accounting for travel time to the frontier. We denote $\mathcal{K}_a$ as the set of robots that rendezvoused and each element in Q as the path length for each robot in $\mathcal{K}_a$ to their next frontier point, f_{k,p^*}. Re-indexing the sequence to be monotonically decreasing, we designate q as the first element of the sequence and the furthest agent from the frontier.

For determining the optimal number of robots for each task, let Ψ represent the set of all possible permutations of $N_d + 1$ numbers such that the sum of each permutation equals $|\mathcal{K}_a|$. The first element represents the number of robots assigned to exploration and elements in the set

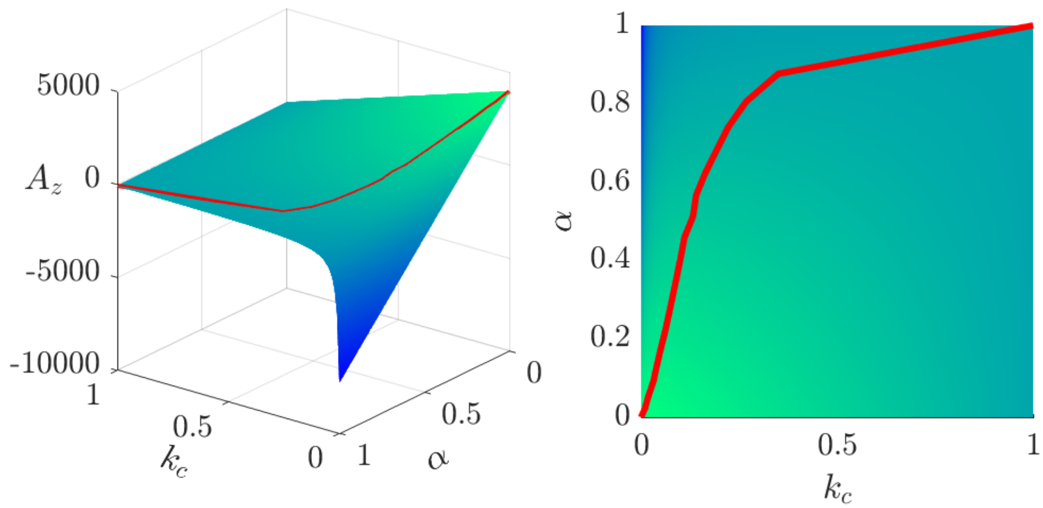


Figure 2.5: A graphical representation of (2.12). The z-axis in the left figure is calculated objective value. A_z denoted by the red line is the optimal mix of robots allocated to the task depending on α.

$\{2, ..., N_d + 1\}$ represent the number of robots assigned to each n task. The team minimizes the following optimization problem given K_a available robots and N_d discovered tasks:

$$A_z = \max_z \;\; \beta(\tau_e - q)\Psi_z(1) - \alpha \sum_{n=2}^{N_d+1} \frac{L_n}{\Psi_z(n) + 1}. \tag{2.12}$$

The separable equations weighted by α and β represent task completion and exploration, respectively. For task completion, the more agents present at a task, the less time a task will take. For exploration, we assume all agents will be able to decrease environmental entropy in the exploration state after traveling to the frontier in q steps. The variable z is the element of the set in Ψ that maximizes the value A_z. In most cases, N_d and $|\mathcal{K}_a|$ will be a small fixed number so that exhaustive search is practical.

Fig. 2.5 shows an example trade-off space assuming that $\alpha + \beta = 1$. The surface of this plot displays the available solutions and the red line shows the normalized optimal number of robots assigned to task completion given the prioritization, α. This showcases that, based on our policy optimization, the number of robots allocated to a task (π_2) or exploration (π_3) is dependent on the prioritization of objectives.

Robots bid on task completion (π_4) or exploration (π_3) based on distance to either assignment. Ties are broken by a robot's index value. We also point out that if a robot is assigned to search for another robot who did not rendezvous and is assumed to have found a task, it uses the last known location, centroid assignment, target, and occupancy map to recreate the exploration steps.

Fig. 2.6 shows an example case study for a robot who decides not to rendezvous after discovering a task. Fig. 2.6(a) is the initialization of the example environment, and, in Fig. 2.6(b), the agents start to explore. Fig. 2.6(c) shows two agents rendezvousing and the agent that found the task

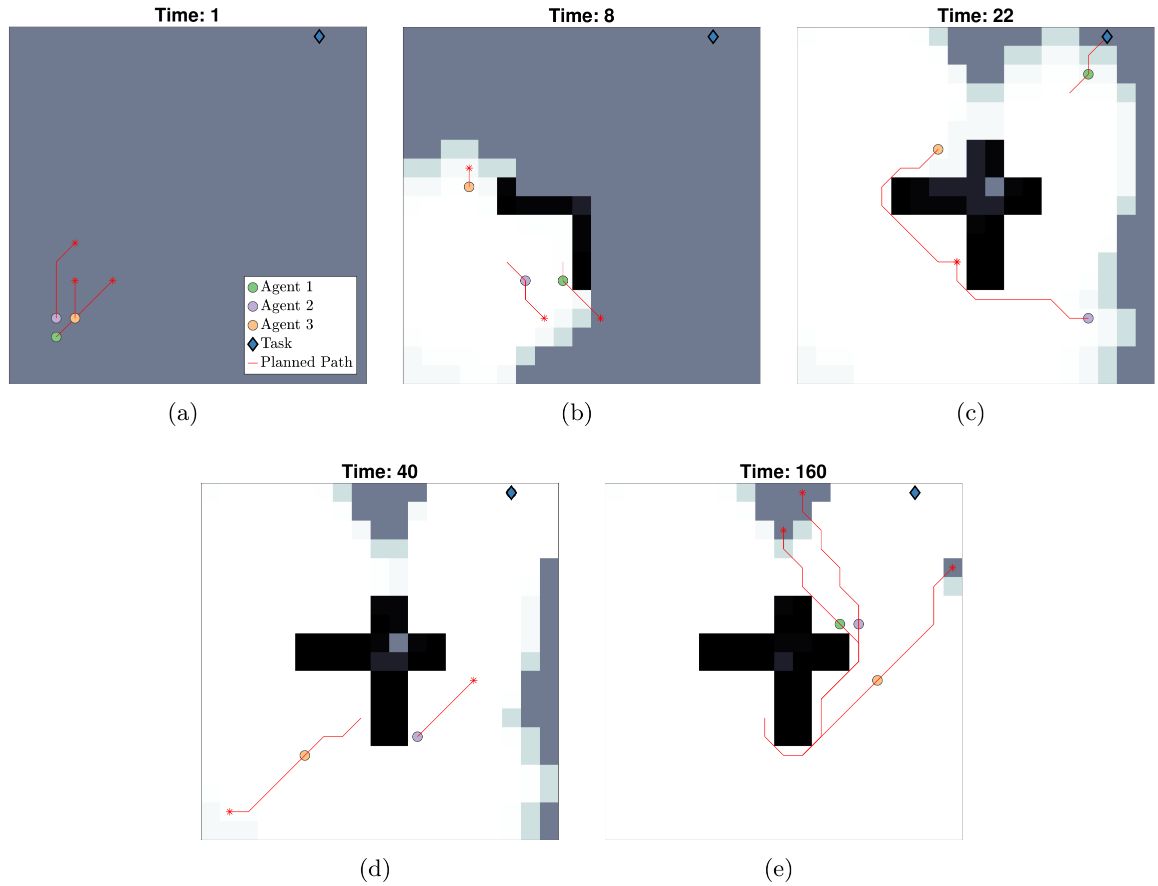


Figure 2.6: Snapshots of a case study where the robot does not to rendezvous after finding a task. (a) shows the initialization of the environment. (b) shows the initial exploration. (c) shows agent 1 finding the task and choosing not to rendezvous with agents 2 and 3, (d) shows agent 2 tasked with finding agent 1 and agent 3 tasked with exploration. (e) shows final steps of exploration and final rendezvous

choosing to remain at the task using (2.10). After rendezvous and based on the prioritization of exploration over exploitation such that $\beta > \alpha$, Fig. 2.6(c) shows that Agent 2 is assigned to find Agent 1 and the task by retracing its exploration path while Agent 3 is assigned to exploration. In Fig. 2.6(e), the task has been completed and all robots rendezvous to share information and explore the last unknown portions of the map.

2.5 Simulation and Experiment Results

In this section, we provide the results from MATLAB simulations used to evaluate and compare our approach with other methods. Simulations were performed on 20 randomly generated cluttered environments with convex and non-convex obstacles to demonstrate the robustness of the approach. Fig. 2.7 shows some examples of these maps. A single task is placed at a random, unobstructed location. Robots begin with the expectation that tasks should take 200 time steps (μ_L), but the single task has an actual task length of 1000 (μ_ν). γ is set to one and exploration and task completion is equally weighted. Task completion is achieved when the remaining task length reaches zero. Tasks are completed linearly based on the number of robots at the task. Exploration is considered complete when the approximate gradient across all traversable cells is less than 10^{-2}.

All robots begin within communication range in the explore state. Collision between agents is not addressed in the simulation, but local path deconfliction is employed in the experiments based on prioritization to next goal.

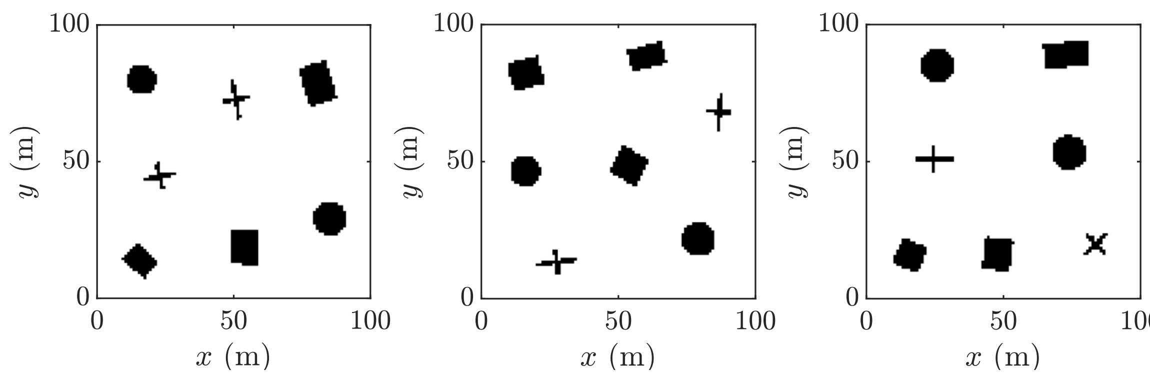


Figure 2.7: Example environments for multi-robot trials

The proposed method is compared against two other exploration techniques. The first is a rendezvous method using the procedure discussed in Sec. 2.4.2 to intentionally share map information without the task allocation procedure from Sec. 2.4.3. In the second method, all robots remain within communication range, and explore using the frontier-based method from Sec. 2.4.1.

Fig. 2.8 shows the results of the three comparisons. Our proposed approach outperforms the others in both median exploration time and median task completion time. In addition, the spread

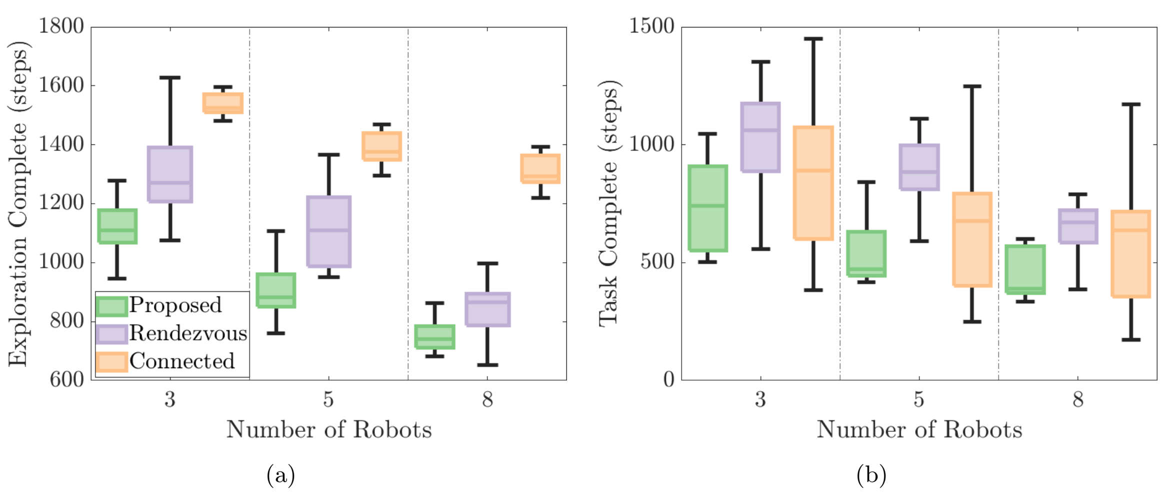
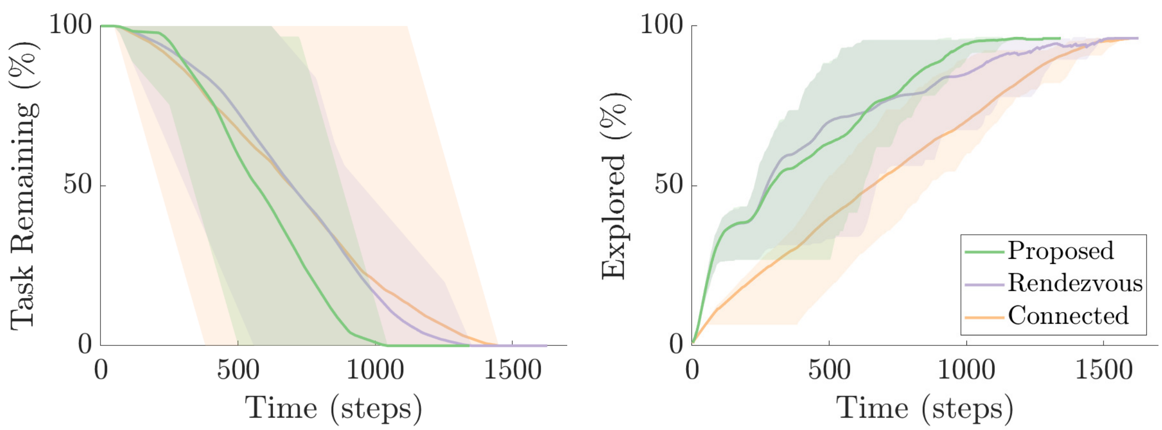


Figure 2.8: Method comparison. (a) shows exploration completion time and (b) shows task completion time across methods with 3, 5, and 8 robots.

of the results is greatly decreased for task completion when compared to the other methods. The connected method completes the task more efficiently if the task is found close to the starting coordinates. However, the team sacrifices exploration and completes the task slower if the task is further from the initial coordinates of the vehicles. The rendezvous method performs worse on task completion since agents complete tasks alone until the other agents have completed their respective partitions. Exploration in the rendezvous method showcases improvement over the connected swarm method, but performs approximately 14% to 18% slower coverage than our proposed method.

Results shown in Fig. 2.9 display a perspective over time for three agents. We observe that our method outperforms both rendezvous and connected methods and decreases the variance on average. We show in Fig. 2.9(b) that for exploration, at best, our method is equivalent to the rendezvous method when setting task completion prioritization (α) to zero. In contrast to these naive methods, ours allows the agents to communicate necessary information to members of the team and is able to complete both exploration and tasks faster on average.

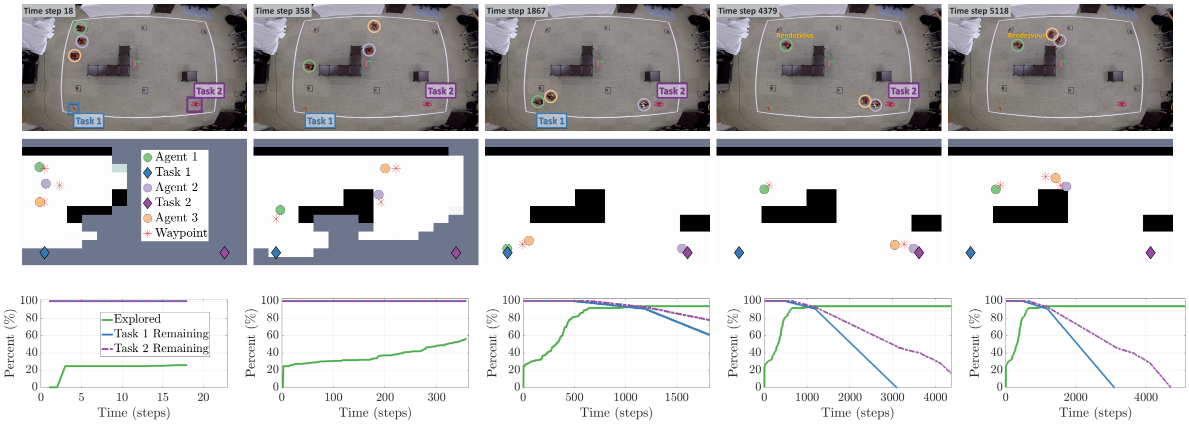


(a) (b)

Figure 2.9: A comparison over time across methods for three agents. In (a) and (b) the spread and speed for exploration and task completion is shown to be smaller and faster than other methods.

The proposed approach was also validated through laboratory experiments with teams of Husarion ROSbot 2.0 UGVs using a Vicon motion capture system and through Gazebo experiments using a team of Clearpath Jackal UGVs in a larger environment. Experiments were performed with at most a three-robot team in order to effectively demonstrate all parts of the proposed approach, including intentional disconnections, searching, and rendezvousing behaviors. In all experiments, the UGVs start within communication range of each other and are tasked to explore the environment and complete any discovered tasks.

Experiments were performed in a 4m$\times$5.5m space containing one convex and one non-convex obstacle considering, as a proof of concept, a sensing and communication range for each robot of 1m. Displayed in Fig. 2.10 are the results from the three-robot experiment in which the vehicles are



(a) (b) (c) (d) (e)

Figure 2.10: Snapshots and results of the three-agent experiment. All agents successfully explore the environment, coordinate to complete tasks, and finally, reconvene at the rendezvous point.

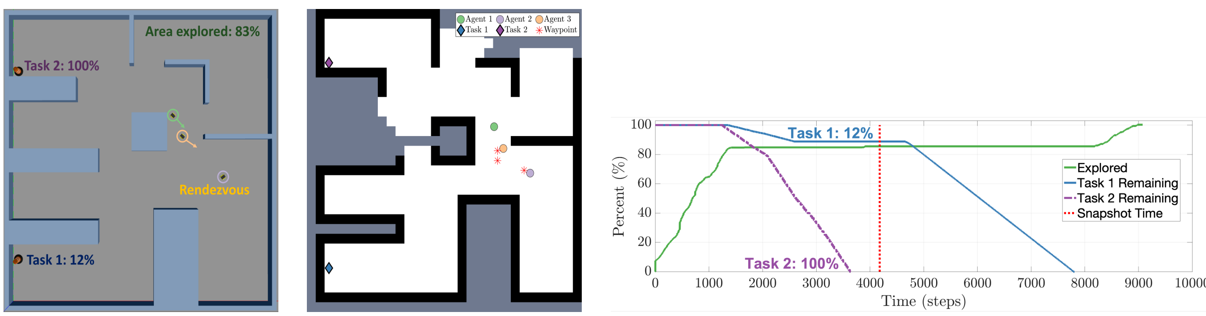


Figure 2.11: Gazebo experiment snapshot at step 4180 and results.

required to search for and complete two tasks unknown a priori. The columns of Fig. 2.10 correspond to different instances within the experiment, and each row from top to bottom shows snapshots of the robots at different times throughout the experiment, the current map of the environment covered by the team, and the map-coverage and task completion over time, respectively. Experiments were run at 10hz, and thus, each time step corresponds to 0.1s. In Figs. 2.10(a-b), the UGVs explore the map in search for tasks. After robots 1 and 2 locates tasks, robot 3 completed exploration and assisted each of the other robots in completing the tasks (Figs. 2.10(c-d)). Finally, once all tasks were completed the robots reconvened at the rendezvous point of the fully explored environment, as shown in Fig. 2.10(e). More lab experiments with two-agents are included in the supplementary material.

To further reinforce these results, we also performed Gazebo experiments that allows for larger environments and longer missions. In Fig. 2.11 we show one of these experiments with 3 UGVs having a 5m sensing and communication range exploring a 30m×30m warehouse-like environment, containing multiple obstacles and two tasks not immediately visible to the UGVs. In Fig. 2.11 we show a snapshot of this experiment and the corresponding results at the time the snapshot was taken.

2.6 Discussion

In this chapter, we have presented a novel framework for multi-robot systems to balance exploration and exploitation in an unknown environment under limited connectivity. The proposed method promotes disconnection through a frontier-based exploration and includes an optimal rendezvous approach to reconnect and share data among the multi-robot system. The extensive simulations and experiment results show the validity, applicability, generality, and scalability of the proposed method. Using this framework, we also demonstrate improved task completion and exploration of unknown environments with respect to standard exploration methods.

The framework presented in this chapter is beneficial for scenarios with an unknown environment. However, when the environment is partially known, it is not required to revisit explored locations. Instead, robots can dynamically meet at previously unexplored areas as needed. Therefore, in the next chapter, we enhance this framework by introducing and utilizing epistemic planning for multi-robot systems which allows the MRS to be robust to failures while disconnected.

Part II

Epistemic Planning for Multi-robot Rendezvous & Task Allocation

Chapter 3

Implicit Coordination for Multi-Robot Teams in Communication Restricted Environments

In the previous chapter, we noted that significant backtracking was essential for rendezvous in unknown environments, a necessity not present in mostly known environments. To extend our approach to partially known environments, we now address the challenge of multi-robot exploration and rendezvous in such environments, further complicating the scenario by introducing potential failures or disturbances. In these cases, robots may not act as previously agreed upon at an unknown time while disconnected from the rest of the team. To mitigate this, we propose an epistemic planning framework for a multi-robot system, which limits uncertainty propagation to a finite set of particles. This enables robots to update their beliefs upon observing the particles implicitly without needing communication. We show that our method surpasses those requiring robots to remain within communication range, and performance times are closely aligned with scenarios where robots can communicate continuously. The material discussed in this chapter was published in the following:

- L. Bramblett, S. Gao, and N. Bezzo, "Epistemic prediction and planning with implicit coordination for multi-robot teams in communication restricted environment," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023.

3.1 Introduction

One natural extension of the work presented in Chapter 2 and in [12] is to consider what humans would do in a similar scenario. Consider two children, Jimmy and Timmy, trying to meet at a playground at 10am. Jimmy arrives at 10am, but does not see Timmy. Jimmy might travel to places where Timmy might be such as his house or school. Timmy might just be running late, but takes a path to intercept Timmy if he came looking for him. We noticed that in such scenarios, humans adequately cope with these problems, performing these tasks collaboratively by extrapolating and empathizing with what other actors might believe if the local plan must change at run-time. This subconscious process can be modally represented as epistemic planning, computing and reasoning about multiple predictions and actions while accounting for apriori beliefs, current observations, and other actors' sensing and mobility capabilities. In this chapter, we insist that if the robots in a team could perform similar reasoning without communication then we could relax the typical connectivity constraints, while increasing autonomy (i.e., decrease human intervention) and mission performance (i.e., more coverage, faster task discovery and completion).

As a reader can note, calculating a distributed plan for coverage while accounting for any combination of robot system failures, changes in the environments, or deviations is intractable. Instead, constructing a finite set of possibilities and implementing a reasoning framework for each robot can reduce computational complexity and allow for more efficient operations. Thus, we introduce a coordinated epistemic prediction and planning method in which a robot propagates a finite set of *belief* states representing possible states of other agents in the system and *empathy* states representing a finite set of possible states from other agents' perspectives. Subsequently, using epistemic planning, we can formulate a consensus strategy such that every distributed belief in the system achieves consensus. For example, consider Fig. 3.1 where two robots are canvassing an environment. During disconnection, Robot 1 maintains a set of possible (belief) states for Robot 2 and also a set of (empathy) states that Robot 2 might believe about Robot 1. Once Robot 2 experiences a failure, it tracks another state in its empathy set. We reason that though Robot 1 holds a false belief about Robot 2's state, there exists an epistemic strategy that can allow robot 1 to find robot 2 (i.e., updating its belief after observing robot 1's believed state).

The contribution of our approach is two-fold: i) an epistemic planning formulation using dynamic epistemic logic, formalizing beliefs and knowledge for robot control and ii) a generalized task assignment and artificial potential field-based model for belief propagation and coverage of an environment with considerations for connectivity constraints and team member dynamics.

3.2 Epistemic Planning Preliminaries

Let us consider a multi-robot system of N_r robots in the set $\mathcal{A}$. We note that initial positions of the robots are known. The system's connectivity graph is denoted as $G = (\mathcal{A}, \mathcal{E})$ where the set $\mathcal{E} \subset \mathcal{A} \times \mathcal{A}$ represents edge connections between robots. An edge $(i, j) \in \mathcal{E}$ indicates that robots i and j are within communication range (i.e. connected). For ease, motivated by most wireless communication modules with a limited range such as WiFi, LoRa, Bluetooth, we abstract communication range as a disk centered on the robot. Robots i and j are considered connected if they are within communication range, r_c.

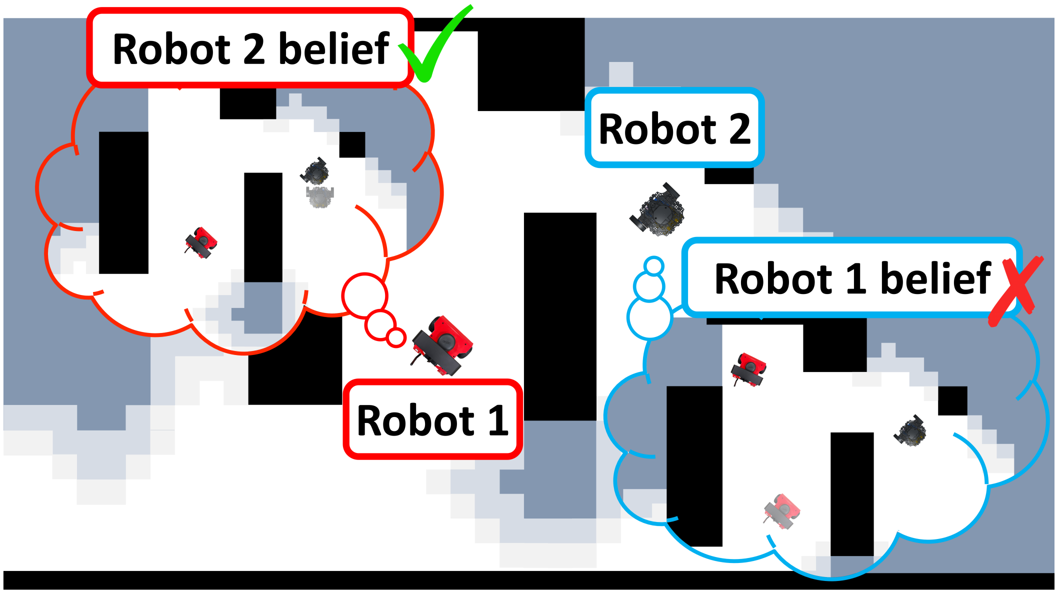


Figure 3.1: Pictorial depiction of the problem. The proposed framework enables a robot to reason from other agents' perspectives as it experiences a behavior change or observes that another robot is not where expected.

Additionally, a number of tasks N_t in the set $\mathcal{T}$ are located in unknown positions within the operating environment. Initially, N_t may be known or unknown. An element τ in $\mathcal{T}$ is defined by the tuple identifying the location, number of required robots, and reward: $(x_\tau, y_\tau, r_\tau, \lambda_\tau)$. We assume the tasks are stationary and completed once a subset of robots navigate within a radius $r_t > 0$.

The robots are assigned to search for the tasks in an environment, $\mathcal{W}$, that is partitioned into N_m cells, which we define as an occupancy map $\mathcal{M} \subseteq \mathbb{R}^2$. When robots navigate to observe unexplored cells $\mathcal{M}_u \subseteq \mathcal{M}$, $\mathcal{M}$ is updated using Recursive Bayesian estimation, though any method can be used. Subsequently, we define the frontier set $\mathcal{F} \subseteq \mathcal{M} \setminus \mathcal{M}_u$ as the set of explored cells adjacent to unknown cells. We assume that the entirety of the exploration area is partially unknown.

Without loss of generality, each of the robots is modeled as a linear time-invariant (LTI) dynamical agent such that

$$\dot{\boldsymbol{x}}_i = \boldsymbol{A}\boldsymbol{x}_i + \boldsymbol{B}\boldsymbol{u}_i + \boldsymbol{\nu}_i, \ \forall i \in \mathcal{A} \tag{3.1}$$

where $\boldsymbol{x}_i \in \mathbb{R}^n$ is the robot i's state vector, $\boldsymbol{u}_i \in \mathbb{R}^m$ is the control input, and $\boldsymbol{A}$ and $\boldsymbol{B}$ are state and input matrices. The variable $\boldsymbol{\nu}_i \in \mathbb{R}^n$ denotes zero-mean Gaussian process uncertainty. We let a state of robot i, $\boldsymbol{x}_i$, represent not only the location and dynamics of the robot, but also its local occupancy map and status. *Status* is defined as a robot's current objective such as covering the environment, communicating, or completing a task. We let a robot i's status be denoted as proposition σ_i and represents which objective a robot is executing.

In this chapter, epistemic and doxastic logic [74] is used to model distributed knowledge and reasoning for system changes during disconnectivity. We define an epistemic state with the following definition.

Definition 1 *An epistemic state is classically described using a tuple $s = (W, R_i, V, W_d)$ for a countable set of atomic propositions, AP, where*

- *W is a non-empty, finite set of possible worlds*
- *$R_i \subseteq W \times W$ is an accessibility relation for robot i*
- *$V \rightarrow 2^{AP}$ is a valuation function.*
- *$W_d \subseteq W$ is the set of designated worlds from which all worlds in W are reachable*

The formula vR_iw means that though the actual world is w, robot $i \in \mathcal{A}$ believes the world is v. We also define s as the epistemic state and set the initial epistemic state to $s_0 = (W, R, V, w_0)$ where $W_d = \{w_0\}$ means that s_0 is the global epistemic state. A world, w, signifies the set of true propositions which in our application is the status of each robot $w = \{\sigma_i \forall i \in \mathcal{A}\}$.

To propagate states of robots, we define beliefs as the set of estimated locations of all robots in the system from each robots' perspective. The set $\mathcal{P} = \{\mathcal{P}_1, \ldots, \mathcal{P}_{N_a}\}$ holds the distributed beliefs of all agents, where an element in $\mathcal{P}_i$ represents possible states from an agent i's perspective of robots $j \in \mathcal{A}$. Ψ is a set of functions that describe the current state of the system. For this application, the epistemic language, $\mathcal{L}(\Psi, \mathcal{P}, \mathcal{A})$ is obtained as follows in Backus-Naur form [45]:

$$\phi ::= H(\omega) \mid \phi \wedge \phi \mid \neg\phi \mid K_i\phi \mid B_i\phi$$

where $i, j \in \mathcal{A}$, $H \in \Psi$ is a function to describe a system state, and ω broadly indicates function arguments. $\neg\phi$ and $\phi \wedge \phi$ denote that propositions can be negated and form logical conjunctions. $B_i\phi$ and $K_i\phi$ are interpreted as "agent i believes ϕ" and "agent i knows ϕ."

Dynamic epistemic logic is expanded from epistemic logic through action models. These models affect how robots perceive an event and its affects on the world.

Definition 2 *An action model $L = (A, R_i^L, pre, post)$ is a tuple with the following definitions*

- *A is a non-empty, finite set of possible actions*
- *$R_i^L \subseteq A \times A$ is an accessibility relation for agent i in the action model*
- *pre is a precondition for an action to be performed*
- *post is a post-condition or effects of an action.*

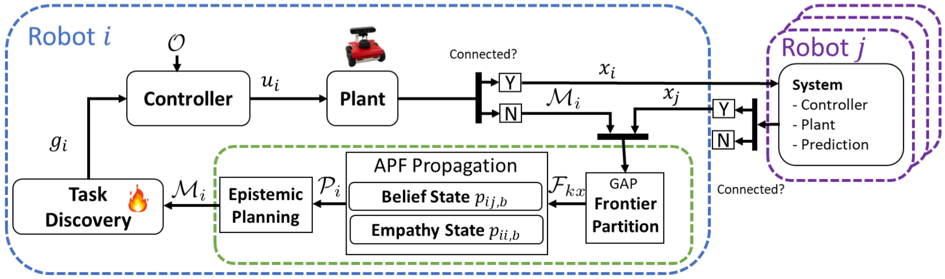


Figure 3.2: Diagram of the proposed approach. The contributions of this section are within the green box.

As such, the epistemic product model is formally introduced as $s \otimes i : a = (W', R'_i, V', W'_d)$ where $i : a$ indicates that an action a has been executed by robot i. In this chapter, we describe a robot's main actions that can occur as: *perceive* a robot or task and *announce* a proposition or system state. The worlds that the system can be in are described by the combinations of all possible statuses of each robot in the multi-robot system.

3.3 Belief & Empathy Propagation

In this section, we present the approach for the coordinated epistemic prediction and planning framework which propagates belief and empathy states to inform frontier assignment and robot control, all while considering failures, task discovery, and unknown obstacles. For ease of discussion let us consider two robots i and j. From robot i's perspective, a *belief state*, $p_{ij,b} \in \mathcal{P}_i$, represents a possible state of a robot j and an *empathy state*, $p_{ii,b} \in \mathcal{P}_i$, describes robot i's belief of robot j's belief about robot i's state. With this knowledge, robot i predicts and tracks empathy states to ensure that a robot j holds one true belief of the state of robot i. The diagram in Fig. 3.2 summarizes this architecture.

As shown in Fig. 3.2, the robot i initially assesses whether communication is successful with a robot j. If communication is successful, robot i uses its current state x_i and the state of robot j, x_j, to partition its frontiers using a generalized assignment problem (GAP) [65] and to predict future states of robot j using an APF method. When connected, epistemic planning is reduced to direct communication of states. If the robots disconnect, a common belief set, $\mathcal{C}_i$, acts as the state for any robot $j \in \mathcal{A}$ from i's perspective. Predictions for these belief and empathy states are accomplished using the same GAP and APF methods. A robot i then uses these predicted states to plan considering its belief about robot j. In both connected and disconnected conditions, the robot's objective is to search for tasks. If connected and a task is discovered, the robots bid on and accomplish tasks. If disconnected, the robots will deviate to accomplish the task and subsequently continue to track its empathy state.

In our coordinated epistemic prediction and planning framework, the robots propagate belief and empathy states for all robots in the multi-robot system. This allows a robot i to plan according to its belief of other robots and reason about what other robots' expect robot i to accomplish while disconnected. As previously noted, to account for uncertainties over long periods of disconnection, it is important to have a finite number of these states. With this goal in mind, we define a finite set of particles, $\mathcal{P}_i$, to represent these belief and empathy states for the i^{th} robot:

$$\mathcal{P}_i = \{p_{ij,b} \ \forall j \in \mathcal{A}, \forall b \in \mathcal{B}\}. \tag{3.2}$$

The i^{th} robot defines its empathy particles as $\mathcal{P}_i^e = \{p_{ii,b} \ \forall b \in \mathcal{B}\}$ and its belief particles about other robots as $\mathcal{P}_i^r = \{p_{ij,b} \ \forall j \in \mathcal{A} \setminus \{i\}, \forall b \in \mathcal{B}\}$ where $\mathcal{P}_i = \mathcal{P}_i^e \cup \mathcal{P}_i^r$. For each robot $j \in \mathcal{A}$, the robot i orders its belief and empathy particles 1 through N_b by likelihood of occurrence (i.e., from largest to smallest). The order is initialized prior to deployment and each robot i initially tracks its first empathy particles.

While not in communication range of other robots, each robot i has a common belief about each robot j and itself. We define robot i's common belief as $\mathcal{C}_i \subseteq \mathcal{P}_i$ and refer to it as the *common belief set*. All robots track their first empathy particle upon disconnection, $\mathcal{C}_i = \{p_{ij,1} \ \forall j \in \mathcal{A}\}$.

If a robot experiences a failure, choosing which of its next empathy particles to track is nontrivial, but we assume each robot i is capable of computing the set of empathy states that are suitable to track, denoted by $\mathcal{P}_i^t \subseteq \mathcal{P}_i^e$. The robot chooses to track the particle in $\mathcal{P}_i^t$ with the highest likelihood. If all robots are within communication range, the first particle becomes the robot's current state and subsequent particles are propagated based on the updated common belief.

Since the robot will be tracking an empathy particle, these states must propagate in a manner that allows the robot to safely and efficiently accomplish the coverage objective with considerations for intentional information sharing. Thus, we propagate the particles using an artificial potential field (APF) that leverages four main objectives: 1) attraction to frontier, 2) cooperative rendezvous, 3) obstacle avoidance, and 4) task completion. In this propagation method, the total force acting on particle $p_{ij,b}$ is formulated generally as:

$$F_{ij,b}^{total} = \beta_1 F_{ij,b}^1 + \beta_2 F_{ij,b}^2 + \beta_3 F_{ij,b}^3 + \beta_4 F_{ij,b}^4 \tag{3.3}$$

considering β_n is a weighting coefficient for force $F_{ij,b}^n$ where each $F_{ij,b}^n$ corresponds to the n^{th} objective listed previously and will be discussed in detail. Local minima is avoided using an A* path planner.

3.4 Frontier Selection

A frontier-based exploration method is proposed here due to its completeness and simplicity. To begin, the force $F^1_{ij,b}$ in (3.3) is an attraction to a frontier set $\mathcal{F} \subset \mathcal{M}$. However, a robot should only traverse unique portions of the environment to reduce redundancy and minimize completion time. So, a decentralized GAP assigns particles to a unique subset of $\mathcal{F}$ using its belief of each robots' capabilities.

For particle $p_{ij,b}$, we allocate frontiers based on the cost of assigning the particle or any other robot. Cost and binary assignment are denoted as Λ and Γ, respectively. The corresponding GAP is formulated as:

$$\begin{aligned} \mathcal{F}_{ij,b} = \min & \sum_{k\in\mathcal{A}} \sum_{z\in\mathcal{F}} \lambda_{zk}\gamma_{zk} \\ \text{s.t.} & \sum_{k\in\mathcal{A}} \gamma_{zk} = 1,\ \forall z \in \mathcal{F} \\ & \sum_{z\in\mathcal{F}} \gamma_{zk} \leq u,\ \forall k \in \mathcal{A} \\ & \sum_{z\in\mathcal{F}} \gamma_{zk} \geq \ell,\ \forall k \in \mathcal{A} \\ & \gamma_{zk} \in \{0,1\},\ \forall k \in \mathcal{A},\ \forall z \in \mathcal{F} \end{aligned} \tag{3.4}$$

where the elements $\lambda_{zk} \in \mathbb{R}_{\geq 0}$ and γ_{zk} represent the z^{th} frontier and k^{th} particle in the matrices Λ and Γ, respectively. Cost generally refers to any traversal metric (i.e., energy, time, etc.) to a frontier point $z \in \mathcal{F}$. The variables $u \in \mathbb{N}$ and $\ell \in \mathbb{N}$ are the upper and lower bounds on the number of frontier points that can be assigned to any particle. Each robot i calculates the frontier assignment for each particle in the set $\mathcal{P}_i$ and the assigned frontier set is denoted as $\mathcal{F}_{ij,b}$.

Subsequently, to utilize the GAP solution, the force $F^1_{ij,b}$ controls the b^{th} particle $p_{ij,b}$ towards its assigned frontiers

$$F^1_{ij,b} = \frac{1}{|\mathcal{F}_{ij,b}|} \sum_{z\in\mathcal{F}_{ij,b}} \frac{s_z - p_{ij,b}}{||s_z - p_{ij,b}||^3} \tag{3.5}$$

where $|\cdot|$ indicates the set's cardinality and the coordinate of a z^{th} frontier is designated as s_z. The force computed in (3.5) encourages particle motion to the unexplored regions of the environment $\mathcal{M}_u$ based on their frontier assignment in (3.4).

3.5 Epistemic Planning

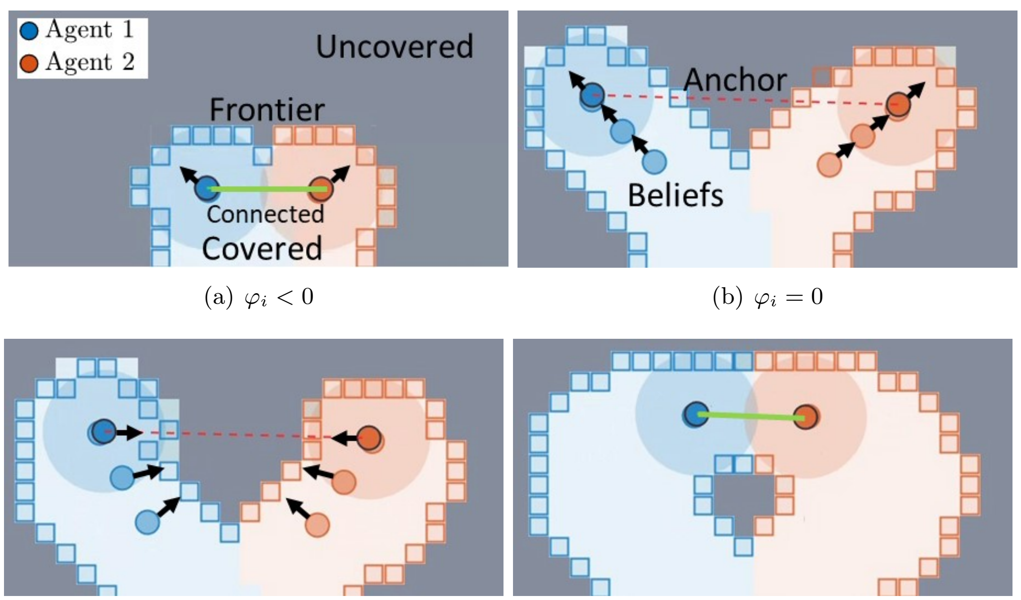
Intentional information sharing allows an agent to communicate any environmental or capability changes with other robots. For this purpose, we introduce $F^2_{ij,b}$ in (3.3) to control each particle $p_{ij,b}$ according to robot i's common belief set:

$$F^2_{ij,b} = \varphi_i \sum_{k \in \mathcal{A}} \frac{c_k - p_{ij,b}}{||c_k - p_{ij,b}||^3} \tag{3.6}$$

$$\varphi_i = \begin{cases} -h(t_r, \tau), & t_r < \tau \\ h(t_r, \tau), & t_r \geq \tau \end{cases} \tag{3.7}$$

where c_k denotes the k^{th} element in $\mathcal{C}_i$. Variable τ is a time-based threshold for rendezvous and $h : (t_r, \tau) \mapsto \mathbb{R}_{\geq 0}$ where t_r is the time lapsed since the last successful communication.

Fig. 3.3 shows the effect of φ_i. Given the partitioned frontier from (3.5), the robots' particles are incentivized to travel i) away from c_k when $\varphi_i < 0$, ii) towards its assigned frontier when $\varphi_i = 0$, or iii) towards c_k when $\varphi_i > 0$. We denote the line between common belief particles as the *anchor line*. In this way, (3.6) controls all of robot i's particles to all beliefs in $\mathcal{C}_i$ when $t_r > \tau$. This is



(a) $\varphi_i < 0$

(b) $\varphi_i = 0$

(c) $\varphi_i > 0$

(d) Reconnected and reset

Figure 3.3: Depiction of APF forces for each particle given different φ_i over time. The color of the anchor line indicates communication (green) or no communication (red).

formalized in the following lemma:

Lemma 3.1 *If $F^2_{ij,b} = 0$, as $t \to \infty$ all particles in the set $\{p_{ij,b} \mid p_{ij,b} \in \mathcal{P}\}$ will converge to c_k, $\forall k \in \mathcal{A}$.*

Proof: Given the $\lim_{t\to\infty} F^1_{ij,b} + F^4_{ij,b} = 0$ once all area has been covered, the only force acting on each particle will be $F^2_{ij,b}$. Also, $\varphi_i > 0$ since $t_r > \tau$ when $t \to \infty$, all particles in the set $\{p_{ij,b} \mid p_{ij,b} \in \mathcal{P}_i\}$ will converge to $c_k \in \mathcal{C}_i$. ■

Considering that the $c_k \in \mathcal{P}_i$ is controlled via (3.6), all common belief states in $\mathcal{C}_i$ converge and so, all particles in $\mathcal{P}_i$ converge to the same rendezvous location. Thus, it is imperative to ensure that while the robots are not communicating they can reach consensus such that

$$\mathcal{C}_i \equiv \mathcal{C}_j, \quad \forall i, j \in \mathcal{A}. \tag{3.8}$$

We coordinate this consensus using dynamic epistemic logic.

Referring to the previously established semantics for DEL in Sec. 3.2, we introduce the set Ψ consisting of a binary and tertiary function, `track` and `anchor`, noting the argument for the i^{th} robot is denoted as A_i for readability in the epistemic model. The function $\texttt{track}(A_i, p_{ii,b})$ is read as "robot i is tracking empathy particle b" and $\texttt{anchor}(A_i, p_{ii,b}, p_{ij,b})$ is read as "robot i is using belief particle $p_{ij,b}$ and empathy particle $p_{ii,b}$ as its common belief."

Since a robot's failures can affect a robot's capabilities in disparate ways and at any time, we formulate a strategy for rendezvous that accounts for all possible combinations of failures. Given that robots are connected given a communication graph G, we also assume particles can observe each other similarly (e.g., based on range, line-of-sight), denoted logically as $p_{ij,b} \triangleleft p_{ik,b}$. If two common belief particles have observed each other, we know that one of four events have occurred for two robots: i) A_i has observed that A_j is not tracking the common belief particle, ii) A_j has observed that A_i is not tracking the common belief particle, iii) neither agents observe either common belief particle, or iv) both agents communicate. In events (i)-(iii), neither robot knows the true system state since the robots did not communicate.

Given that our current common belief is the b^{th} particle and all particles are within observation range, we define the consensus-based policy sequence formally as:

$$\begin{aligned}
a_0 &= p_{ij,b} \triangleleft p_{ji,b} \ \forall i, j \in \mathcal{A} \\
a_1 &= \texttt{anchor}(A_j, p_{ij,b+1} \ \forall j \in \mathcal{A}) \\
a_s &= a_0 \otimes a_1
\end{aligned}$$

such that the frame-policy update is:

$$f_{b+1} = f_b \otimes a_s \tag{3.9}$$

where $b \in \mathcal{B} \setminus \{N_b\}$ and $\otimes$ is a modal product. This strategy is used until all robots communicate at which time the particles are reset to the robots' poses and dynamics are updated.

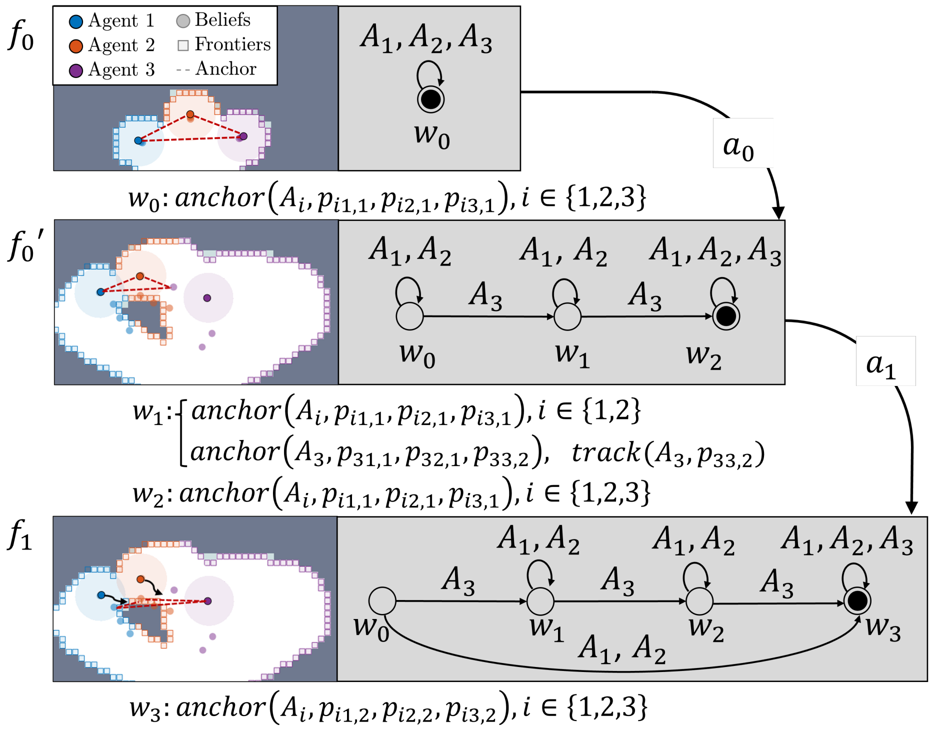


Figure 3.4: Example scenario where A_3 experiences a failure. The Kripke models are shown in the gray boxes with frame transitions denoted as a_n.

Example: Consider f_0, f_0', f_1 in the Kripke model shown in Fig. 3.4 where the true world is denoted by the black vertex, edges represent accessibility relation (R), and new propositions (V) are denoted in each worlds' sets. From the perspective of robot A_1, each robot initially tracks particles $p_{11,1}$, $p_{12,1}$ and $p_{13,1}$. All robots propagates four possible belief states for each teammate and four empathy states of itself. We denote robot A_1's knowledge and belief of his own state as K_1 $\texttt{track}(A_1, p_{11,1})$. Similarly, robot A_1's knowledge and belief about his teammate's state is represented as $B_1\texttt{track}(A_j, p_{1j,1}), j \in \{2,3\}$ and empathy from robot A_1's perspective is shown as $B_1 B_j\texttt{track}(A_1, p_{j1,1}), j \in \{2,3\}$ and can be read as "robot A_1 believes that robot j believes that robot A_1 is tracking particle 1". We denote the initial common belief propositions for all three agents as $\texttt{anchor}(A_i, p_{i1,1}, p_{i2,1}, p_{i3,1}),\ i \in 1,2,3$.

After disconnection (as shown in f_0' in Fig. 3.4), the robots can no longer assume to have knowledge about the true state of the system. In this scenario, A_3 experiences a failure such that $\texttt{track}(A_3, p_{33,2})$ and updates its common belief (w_1). However, A_3 subsequently reasons that A_1 and A_2 hold a false belief that A_3 is still tracking the first particle. Thus, A_3 reverts its common belief in w_2 to mirror the initial proposition such that $\texttt{anchor}(A_i, p_{i1,1}, p_{i2,1}, p_{i3,1})$.

When $A_1 \lhd p_{13,1}$ in f_0', A_1 and A_2 are are connected and so A_1 relays that $A_1 \lhd p_{13,1} \wedge A_1 \lhd \neg A_3$, reasoning that A_3 does not know that both agents are still tracking their respective first particle. A_3 also knows that all three first particles are within communication range, but cannot observe any additional information. Thus, all three agents update their common belief such that $\texttt{anchor}(A_i, p_{i1,2}, p_{i2,2}, p_{i3,2})$ and all particles begin converging to the updated common belief set.

3.6 Obstacle Avoidance & Task Allocation

Finally, we consider the last two forces in (3.3). To avoid obstacles, force $F^3_{ij,b}$ is formulated as:

$$F^3_{ij,b} = \frac{1}{|\mathcal{O}|} \sum_{o \in \mathcal{O}} \frac{s_o - p_{ij,b}}{||s_o - p_{ij,b}||^3} \tag{3.10}$$

where $s_o \in \mathcal{C}$ is the coordinate for an obstacle $o \in \mathcal{O}$ in the environment. We note that $\mathcal{O}$ is a set of the commonly known obstacles for all agents. For example, if a robot individually encounters an obstacle, but has not communicated its location to all teammates, the particles' motions are not affected by this new obstacle which is unknown to other robots.

For attraction to tasks, the force $F^4_{ij,b}$ is only active when all agents are connected to propagate particles towards any commonly known tasks. These tasks are centrally auctioned and each robot submits a bid based on estimated traversal time to the task. Assigned tasks placed in a queue set $\mathcal{Q}_i$, and distributed to robot i's particle task queues, $\mathcal{Q}_{ij,b}$. Attraction to the first task, $\mathcal{Q}_{ij,b}[1]$, in a particle's queue is formulated as:

$$F^4_{ij,b} = \mathcal{Q}_{ij,b}[1] - p_{ij,b} \tag{3.11}$$

where the coefficients for $F^1_{ij,b}$ and $F^4_{ij,b}$ depend on the particle's queue $\mathcal{Q}_i$, such that:

$$\begin{cases} \beta_1 = 0 \text{ and } \beta_4 > 0, & \text{if } \mathcal{Q}_i \neq \emptyset, \\ \beta_1 > 0 \text{ and } \beta_4 = 0, & \text{otherwise.} \end{cases} \tag{3.12}$$

Once the task queue is empty, the particle force for tasks is set to zero and coverage resumes.

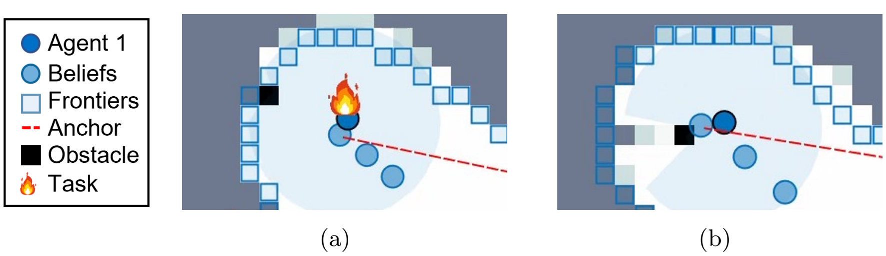


Figure 3.5: Examples of necessary deviation. In (a) the robot spots a task and updates its RHC goal. In (b) the robot must avoid a discovered obstacle while minimizing distance to the particle.

3.7 Particle Tracking

After particle propagation has occurred, a robot must predict and track its empathy particle considering the possibility of a new obstacle or discovering a task while disconnected. Though any constrained tracking algorithm can be employed, we use a nonlinear receding horizon controller (RHC) to minimize the distance to the particle while maintaining a radius from any obstacle [57]. Additionally, if the team is disconnected and a robot discovers a task within its observing radius r_o, the tasks are re-indexed by monotonically increasing cost and placed in the set $\mathcal{Q}_i$. The controller's goal is updated to the first queued task location. Once the robot has traveled within the completion radius, r_t, the robot continues tracking its respective particle and the task is removed from the robot's queue. An example for both obstacle avoidance and task discovery are shown in Fig. 3.5.

3.8 Simulation and Experiment Results

We provide results and comparisons from MATLAB simulations with our approach implemented on a two robot team. Simulations were performed on 15 random 50m $\times$ 50m environments with 5-15 initially unknown obstacles and a maximum of seven tasks. The robots start by assuming that the environment has no obstacles and do not know the location of the tasks. We compare the results between: 1) no failures, 2) one failure by one robot, and 3) one failure by each robot at random times.

Each robot propagates three particles traveling at 2m/s, 1m/s and 0.5m/s. A failure can cause the robot to track either the second or third particle. The particles propagate according to these failure velocities and the maximum communication range is 10m from the center of the robot.

The proposed approach is compared against two other methods. The first method applies a constant connectivity constraint, not allowing agents to travel outside of a 10m communication range. The second method assumes ideal conditions, where robots can communicate across the entire environment. Both methods use an artificial potential field technique for controlling the robots towards uncovered regions and away from obstacles. In all methods, the initial maximum velocity is 2m/s, the simulated LiDAR range is 5m, and the robots' motion is modeled using unicycle dynamics. Fig. 3.6 shows an example at various time steps for the three methods, displaying the coverage disparity of the connected method versus our proposed method and showing the similarities between our method and the ideal method.

As shown in Fig. 3.7, the proposed method outperforms the fully connected method in all scenarios. Additionally, the median coverage time for the proposed method is similar to the ideal method, even with the communication limitation.

The proposed approach was also validated through laboratory experiments with a two-robot team. The team consists of a Husarion ROSbot 2.0 UGV and a Turtlebot3 Burger UGV using a Vicon motion capture system. The two-robot experiments effectively demonstrate all parts of the proposed approach, including intentional disconnections, searching, and rendezvous behaviors. In all experiments, the UGVs start within communication range and are tasked to cover the environment and complete any discovered tasks.

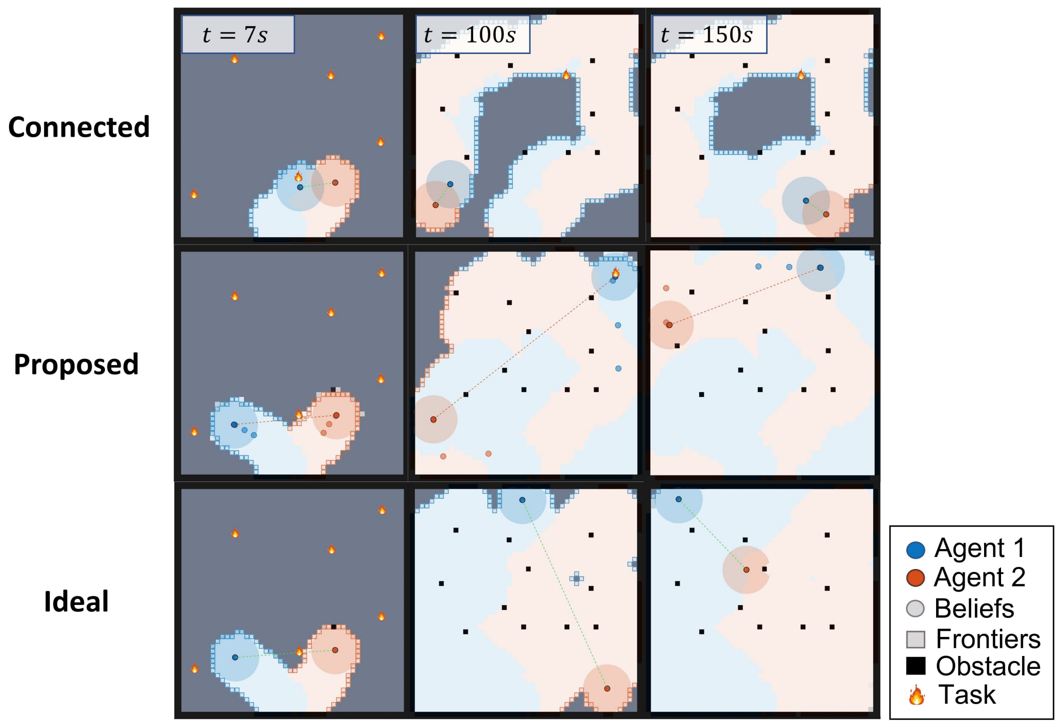


Figure 3.6: Snapshots of example simulations. The robots experience a failure each at random times with 11 unknown obstacles and 7 tasks.

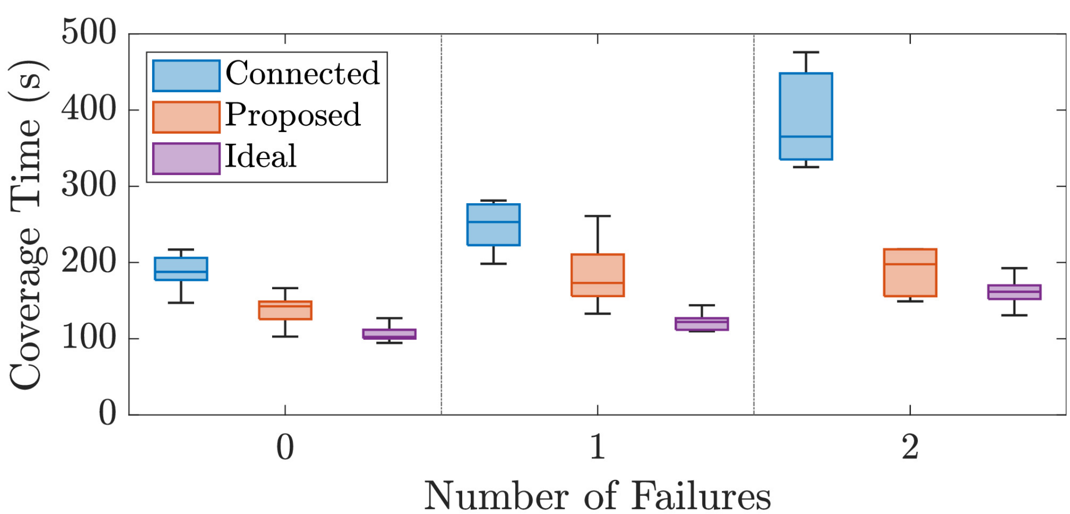
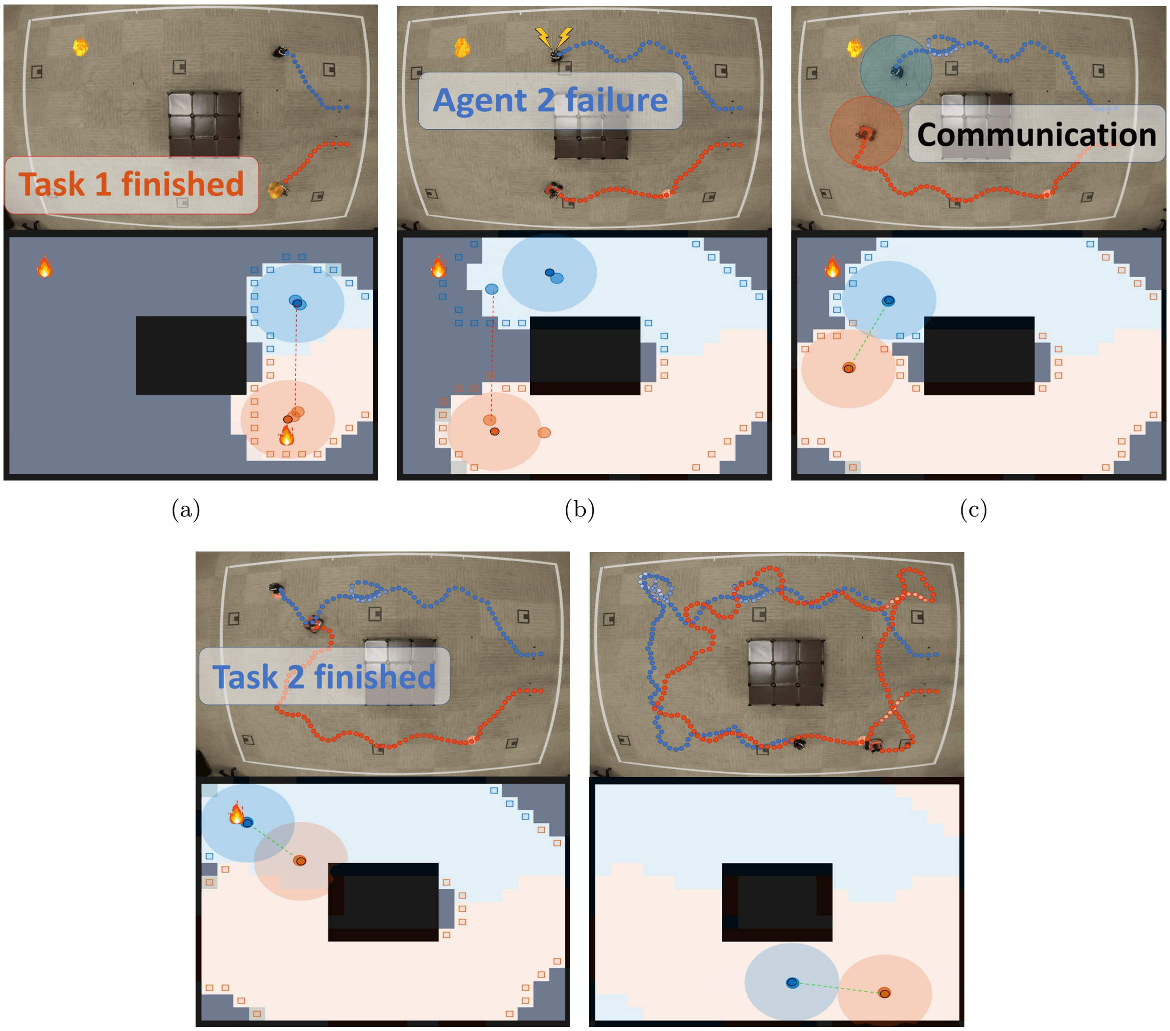


Figure 3.7: Comparison between methods given 0, 1, and 2 failures.



(a) (b) (c) (d) (e)

Figure 3.8: Snapshots and results of an experimental case study.

Experiments were performed in a 4m×5.5m space containing convex obstacles considering, as a proof of concept, a sensing and communication range for each robot of 1m. Displayed in Fig. 3.8 are the results from the two-robot experiment in which the vehicles are required to search for and complete two tasks unknown apriori. The columns of Fig. 3.8 correspond to different instances within the experiment, and each row from top to bottom shows snapshots of the robots at different times throughout the experiment and the current map of the environment covered by the team. In Fig. 3.8(a), the UGVs start to cover the map in search of tasks and disconnect. Robot 1 (ROSbot) finds a task and completes it. In Fig. 3.8(b), robot 2 (Turtlebot) experiences a fault and begins following the second empathy particle. In Fig. 4.16(c-d) the robots connect, share fault information, and bid on the discovered task. Robot 1 receives a larger share of the frontier as a result of Robot 2's failure and Robot 1 is assigned the task based on proximity. Finally, once all tasks are completed

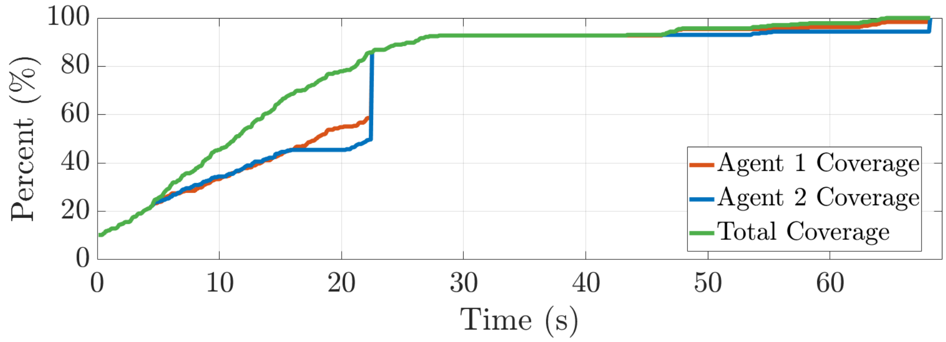


Figure 3.9: Graphical coverage over time for the experimental case study.

and no frontiers remain, the agents rendezvous and the final results are shown in Fig. 3.8(e). Final coverage over time for each robot is shown in Fig. 3.9. More lab experiments with two-robots are included in the supplementary material.

3.9 Discussion

In this chapter, we have presented a novel framework for multi-robot systems to use epistemic planning to propagate and behave according to a set of beliefs. The proposed method promotes disconnection using frontier-based exploration and includes a dynamic rendezvous approach to reconnect and share data among the multi-robot system. The extensive simulations and experiment results show the validity, applicability, generality of the proposed method. Using this framework, we also demonstrate improved task completion and coverage time of partially known environments with respect to standard coverage methods.

While this framework provides a comprehensive solution for multi-robot systems operating in communication-limited, partially-known environments, it assumes that tasks are designed for individual robots and that the system is homogeneous. For complex tasks requiring multiple robots or diverse robot capabilities, the system must include a way to share information among the team. In the following chapter, we extend this work to address heterogeneous teams tackling complex dynamic tasks within the environment.

Chapter 4

Epistemic Planning for Complex Task Allocation

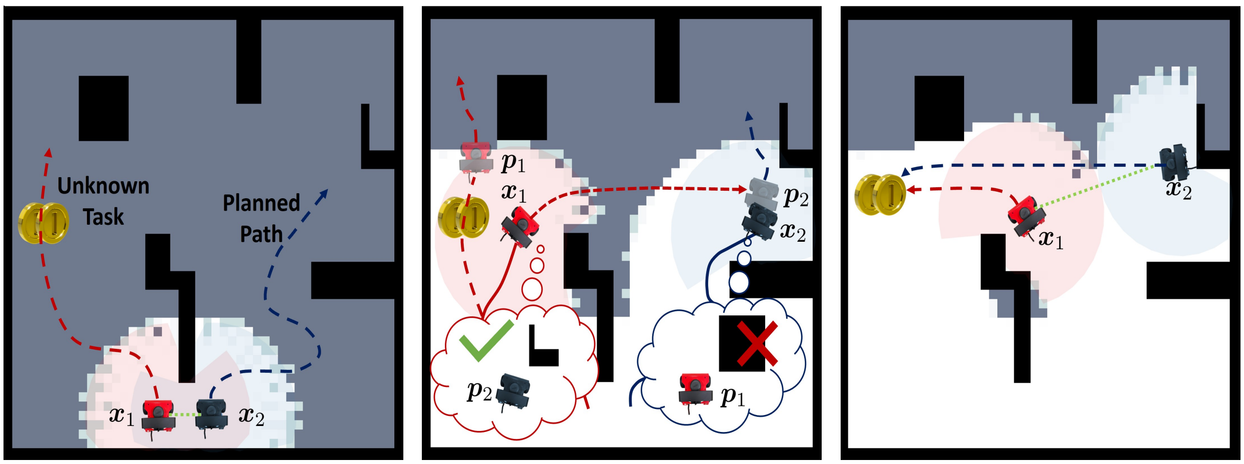
Building on this epistemic planning framework, we extend the method from the previous chapter for heterogeneous multi-robot teams to execute complex tasks in environments that are partially known and where communication is limited. We introduce a novel dynamic task assignment and gossiping protocol, enabling agents to communicate with other robots as necessary to accomplish tasks utilizing updates to their epistemic state to reason about actions to take. The work in this chapter is based on the following publications:

- L. Bramblett and N. Bezzo, "Epistemic planning for multi-robot systems in communication restricted environments," *Frontiers in Robotics and AI*, vol. 10, p. 67, 2023.
- L. Bramblett and N. Bezzo, "Epistemic planning for heterogeneous robotic systems," in *2023 IEEE International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2023.

4.1 Introduction

In this chapter, we build on the framework from Chapter 3 and formalize a problem in which the goal is to cooperatively explore, find, and accomplish tasks in the environment; however, the scenario is further complicated by tasks at unknown locations that may require multiple agents (e.g. lifting a heavy object or inspecting a large structure). Since the locations of these tasks is initially unknown, calculating a distributed plan for coverage while accounting for any combinations of a robot system's actions, changes in the environment, or deviations is intractable over long periods of disconnection. Alternatively, establishing a reasoning framework for a finite set of possibilities for each robot can reduce computational complexity and increase the mission efficiency. Thus, we introduce an epistemic prediction and planning method with gossip protocol in which a robot propagates a finite set of *belief* states representing possible states of other agents in the system and *empathy* states representing a finite set of possible states from other agents' perspectives. Each agent may attempt to communicate and allocate found tasks by traveling to the believed location of another agent.

For task assignment, we focus on a subcategory of the MRTA problems known as the single-task, multi-robot, time-extended allocation problem [ST-MR-TA], meaning that each robot can only execute one task at a time and tasks may require multiple robots. There are several challenges that arise to allow efficient and cooperative behavior given limited communication including: 1) how to efficiently cover a partially unknown environment for tasks, 2) upon discovery, how should tasks be ideally allocated to a subset of robots, and 3) how to communicate necessary information to robots in the system if disconnected.



(a) (b) (c)

Figure 4.1: Pictorial depiction of the problem presented in this chapter. The proposed framework enables a robot to reason from other robots' perspectives as it experiences a behavior change or observes that another robot is not where expected.

Consider Fig. 4.1 where two robots are canvassing an environment. During disconnection, robot 1 maintains a set of belief states for robot 2 ($\boldsymbol{p}_2$) and also a set of empathy states that robot 2 might believe about robot 1 ($\boldsymbol{p}_1$). Once robot 1 finds a task that also requires robot 2, it attempts to communicate by routing to robot 2's belief state shown in Fig. 4.1(b). Robot 1 travels to the believed location of robot 2 and is able to communicate if $\boldsymbol{p}_2$ is a close approximation of $\boldsymbol{x}_2$, illustrated in Fig. 4.1(c). We reason that though robot 2 holds a false belief about robot 1's state, there exists an epistemic strategy that allows robot 1 to communicate with robot 2 (i.e., robot 1 propagating and checking the belief state for robot 2 and by robot 2 empathizing with robot 1's belief).

4.2 Propagation & Coverage

Similar to Sec. 3.3, we propagate a finite set of particles for the i^{th} robot. In this case, since all robots may have different knowledge of each other, we consider a second-order depth of reasoning representation for this section, defined as:

$$\mathcal{P}_i = \{p_{ij,k}\ \forall j \in \mathcal{A}, \forall k \in \mathcal{A}\}. \tag{4.1}$$

The i^{th} robot defines its empathy particles as $\mathcal{P}_i^e = \{p_{ij,i}\ \forall j \in \mathcal{A}\}$ and its belief particles about other robots as $\mathcal{P}_i^r = \{p_{ij,k}\ \forall j \in \mathcal{A}, \forall k \in \mathcal{A} \setminus \{i\}\}$ where $\mathcal{P}_i = \mathcal{P}_i^e \cup \mathcal{P}_i^r$ for a multi-robot system of N_r robots in the set $\mathcal{A}$. The particle $p_{ij,k}$ is interpreted as a second-order belief (a belief about beliefs) and represents robot i's belief about robot j's belief about robot k's state. We note that initial positions of the robots are known and thatthe system's connectivity graph is denoted as $G = (\mathcal{A}, \mathcal{E})$ where the set $\mathcal{E} \subset \mathcal{A} \times \mathcal{A}$ represents edge connections between robots. An edge $(i, j) \in \mathcal{E}$ indicates that robots i and j are within communication range (i.e. connected). A number of tasks N_t in the set $\mathcal{T}$ are located in unknown positions within the operating environment. Initially, N_t may be known or unknown. An element τ in $\mathcal{T}$ is defined by the tuple identifying the location, number of required robots, and reward: $(x_\tau, y_\tau, r_\tau, \lambda_\tau)$. We assume the tasks are stationary and completed once a subset of robots navigate within a radius $r_t > 0$.

To start, all particles are set as the robots' initial state. While not in communication range of other robots, each robot i propagates a subset of belief particles from the last globally communicated state between robot i and robot j. We define this set of particles as $\mathcal{C}_i \subseteq \mathcal{P}_i$ and refer to it as robot i's *common belief* set. All robots track a second order belief or empathy particle, $p_{ij,i}$, upon disconnection whose motion is planned using the common belief set, $\mathcal{C}_i = \{c_{ij}\ \forall j \in \mathcal{A}\}$. Each particle $c_{ij} \in \mathcal{C}_i$ propagates according to the last global epistemic state. The common belief is reset when all robots are within communication range and new knowledge is shared (i.e. coverage, unknown obstacles, tasks). Each particle, $p_{ij,k}$, is propagated towards its goal state, $g_{ij,k}$, using the given vehicle dynamics and a smoothed A* path planning algorithm [60]. The goal selection is dependent on a particles' status. Within this chapter, there are four main statuses that each particle can be in: *exploring*, *gossiping*, *completing a goal*, or *going home*, noting that these statuses are pre-defined and mission dependent. The go-to-goal behavior for each particle is accomplished via Artificial Potential Field (APF) [44] because of its simplicity and calculation speed. When the APF is coupled with the A* path planning algorithm, local minima are avoided.

We also define two main actions that may occur as: *perceive* a robot or task and *announce* a proposition or system state, but these actions are not triggered until task discovery. To find these tasks, each robot is initially responsible for exploring a portion of the environment. We introduce a partitioning and coverage mechanism using the common belief set, $\mathcal{C}$, for cooperative robots given a partially unknown environment while disconnected.

To begin, an i^{th} robot updates its local map and estimated coverage from robot j at the location in described by the j^{th} particle in the set $\mathcal{C}_i$ using recursive Bayesian estimation. Using this common belief map, an i^{th} robot determines its frontier set, $\mathcal{F}_i$, by assessing which explored cells are adjacent to unknown cells. Additionally, the optimal partition of $\mathcal{F}_i$ is the tessellation $\mathcal{V}_i(\mathcal{C}_i) = \{\mathcal{V}_{i1}, \mathcal{V}_{i2}, \ldots, \mathcal{V}_{iN_r}\}$ generated by common belief particles in $\mathcal{C}_i$ denoted as the points $(c_{i1}, c_{i2}, \ldots, c_{iN_r})$ and weighted by a constant factor ψ_j based on a j^{th} robot's capability:

$$\mathcal{V}_{ij} = \{f \in \mathcal{F}_i | \quad \psi_j ||f - c_{ij}|| \leq \psi_k ||f - c_{ik}||, \forall k \neq j\}. \tag{4.2}$$

Using the common belief set versus the communicated location of robots allows for decentralized coverage while disconnected by implicitly reasoning about the assignments of other robots and their individual motion plans.

After determining each common belief particles' frontier partition, the utility of each frontier point is assessed. The utility of a frontier point is user-defined (e.g. distance to frontier point, distance to other robots, heading difference) while incorporating a penalty for frontier points outside of a particles' partition such that the utility of each frontier point is defined as follows

$$\upsilon_{ij,z} = \begin{cases} u(f_z, \alpha_j) + \Delta & f_z \notin \mathcal{V}_{ij} \\ u(f_z, \alpha_j) & f_z \in \mathcal{V}_{ij} \end{cases} \tag{4.3}$$

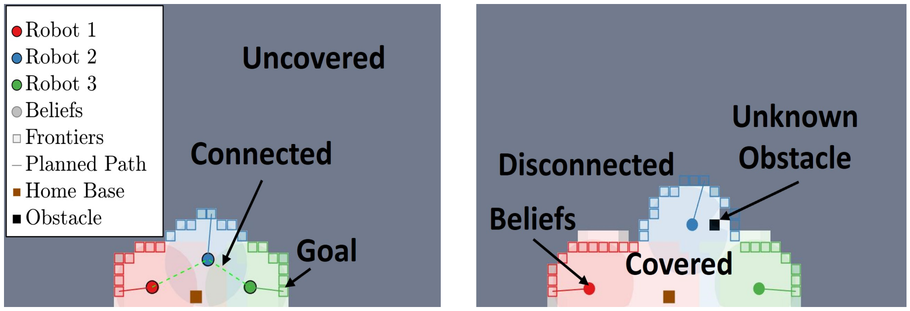
where Δ is a penalty for frontier points outside of a particles' partition and $u(\cdot)$ is the utility function for assigning c_{ij} to $f \in \mathcal{F}_i$. Subsequently, the frontier point that minimizes the utility from (4.3) is defined as

$$z^* = \underset{z}{\operatorname{argmin}} \ \upsilon_{ij,z} \tag{4.4}$$

and

$$g^c_{ij} = f_{z^*}. \tag{4.5}$$

The variable g^c_{ij} is the frontier point goal for the common belief particle, c_{ij}, which encourages the common belief to propagate towards unique, uncovered portions of the environment. If a particle's status is *exploring*, it also shares the same goal as its respective common belief particle: $g_{ij,j} = g^c_{ij} \forall j \in \mathcal{A}$. Otherwise, the goal for each particle depends on the particle's status such as going to a task, communicating with another robot, or traveling to home base. Fig. 4.2 shows an example of particles propagating in a partially unknown environment. In Fig. 4.2(a), the robots begin in communication range and establish goals along the frontier using (4.3). Fig. 4.2(b) shows the robots disconnect as they move towards their respective frontier goals and establish belief states. The plotted belief states for an i^{th} robot are the belief states of all other robots and an empathy state from robot i's perspective. The covered area is shaded by the robot color that accomplished coverage and the plotted frontier points are the frontier points from each belief states' perspective,



(a) (b)

Figure 4.2: This figure shows the initial stages of coverage for three robots using the proposed epistemic coverage method. **(A)** shows three robots connected and partitioning the environment based on known states. **(B)** shows coverage using the epistemic belief to allocate frontiers in the environment. The actual coverage accomplished by each robot is represented by the light shaded region.

dynamically allocated using (4.2). As the robot is traveling, unknown obstacles may appear and the robot avoids these obstacles while continuing to follow its main empathy particle.

4.3 Epistemic Updates & Planning

Epistemic planning is a modal representation of planning about knowledge and beliefs when the environment changes. Under the assumption that robots have limited communication capabilities, the problem we are solving can be considered a game with imperfect information. [55] points out that multi-player games with imperfect information are undecidable, but using epistemic planning and assuming cooperative robots, we can tame the complexity of the problem to achieve consensus in most disconnected scenarios.

A belief update is the process of accepting new information that may contradict initial beliefs. When robots communicate, any necessary belief updates must take place rationally to ensure global consensus is still retained. Thus, there are four cases in which belief update occurs in this work: i) when globally connected to all robots, ii) when locally connected to another robot, iii) when expecting to connect with another robot, and iv) upon task discovery. Referring to the previously established semantics for DEL in Sec. 3.2, we introduce our action library A that can transform the epistemic state. We let $A = \{perceive(\phi), announce(\phi)\}$. The action *perceive* is when a robot perceives a generic proposition ϕ in the environment, such as a task or robot, and the action *announce* is when a robot communicates with its locally connected team. Also, we introduce the set Ψ with one element such that $\Psi = \{present\}$ which is interpreted functionally in our application for $K_i\, present(\tau)$ as robot i knows the location, required robots, and value of task τ.

The global belief update is relatively simple. All new information is centrally known and so all particle states can be updated to known robot states instantaneously. We assume because robots are cooperative, all belief updates are accepted and do not become outdated unless an event occurs in the environment such as discovering a task; however, each robot may not know when/if the information of the system becomes outdated when disconnected. We formulate the logic for this framework using a series of worlds, w_t, which is the set of propositions of each robot's status, σ_i^t $\forall i \in \mathcal{A}$. Additionally, there exists one true world, w_t^*, at time t and only exists if

$$w_t^* R_i w_t^*, \ \forall i \in \mathcal{A}. \tag{4.6}$$

In order for all robots to know with certainty the true world, all robots' states, $\sigma_t^i \in w_t^*$ must be common knowledge and announced such that the epistemic state from robot i's perspective at time t is:

$$s_{t-1}^i \otimes announce(\boldsymbol{x}) = s_t^i \models K_i \sigma_t^i \bigwedge_{j \in \mathcal{A}} K_i K_j \sigma_t^j \bigwedge_{(j,k) \in \mathcal{E}} K_i K_j K_k \sigma_t^k, \ \forall i \in \mathcal{A}. \tag{4.7}$$

where $announce(\boldsymbol{x})$ is an action symbolizing the announcement of all robots' states. The common belief particles are updated from the announcement of all states to the multi-robot system such that

$$p_{ij,k} \leftarrow \boldsymbol{x}_k, \ \forall (i,j,k) \in \mathcal{A}^3. \tag{4.8}$$

Similarly, all particles are updated according to the most recent public announcement, the common belief set is updated so that

$$p_{ij,i}^* \leftarrow \boldsymbol{x}_i, \ \forall (i,j) \in \mathcal{E}. \tag{4.9}$$

Since the common belief is updated to the world w_t shared according to (4.7), the particles in this set are propagated based on each robot's status propositions. For example, in a two-robot team, if robot 1 communicates with robot 2 that it has found a task and will complete this task, robot 1 and robot 2 would propagate a common belief particle that moved to complete the task before continuing to cover the environment.

The local belief update is more complicated as all robots must also retain the common belief, $\mathcal{C}_i$, for partition consensus among disconnected robots. As such, the common belief is not updated upon receiving new information, but rather the second order belief about each robot. Given that a robot has a belief about the current world, this belief is revised if an action changes the robot i's knowledge of the world

$$s_{t-1}^i \otimes i : announce(\boldsymbol{x}_i) \implies K_i \sigma_t^i \tag{4.10}$$

noting that knowledge and belief are equivalent ($B_i\sigma_t^i \equiv K_i\sigma_t^i$). In turn, a robot may communicate this action to only its connected neighbors:

$$s_{t-1}^i \otimes i : announce(\boldsymbol{x}_i) = s_t \models \bigwedge_{(i,j)\in\mathcal{E}} B_iB_jq_i' \bigwedge_{(i,j)\in\mathcal{E}} B_iB_jB_iq_i' \tag{4.11}$$

noticing that disconnected robots' knowledge is not impacted, nor does robot i update its belief of the overall system and robot j updates its belief about robot i such that

$$p_{ji,i} \leftarrow \boldsymbol{x}_i. \tag{4.12}$$

In this way, the system is able to maintain both local and global beliefs, even while disconnected using this announcement protocol. Also, the set $q_{ij,k} \in \mathcal{Q}_i$ holds the timestamp that information was last shared between robot i and all other robots. Each particle in a connected team is assessed and revised if another robot has a more recent belief to ensure we can plan with the last shared belief. For example, if robot j finds a task and shares a new state with robot i, robot i will set all timestamps in the set $q_{ij,j}$ to the current time and update its particle propagation for particle $p_{ij,j}$ according to the new status of robot j, σ_t^j, until assumed task completion, then the particle will propagate towards its common belief, $c_{ij} \in \mathcal{C}_j$.

The maximum number of worlds in this epistemic model is the combination of all possible statuses in the system or $n \leq 4^{N_r}$. Even this number is too large to track for a small multi-robot system, but, using dynamic epistemic logic, each $p_{ij,k}$ is only updated upon an action in the action library, A.

Fig. 4.3 illustrates the example of local belief update when a task is found and two robots are communicating while a third robot remains disconnected. For ease, in the figure, we display every robot's belief about only robot A. Originally, robot A planned to follow the common belief, but upon discovering a task, replans to complete the task before continuing to track the common belief particle. Robot A and robot B are within communication range and so robot A communicates that it will travel to complete the task before continuing to track the common belief. Robot B updates its belief about robot A (and vice versa). Robot C is not able to receive the updated information and continues to plan according to the common belief. Robots A and B propagate robot C's belief and robot A will eventually continue to track the common belief particle after completing the discovered task.

With our epistemic states and actions defined, we now move to describe how these concepts can be used for planning. A planning task for a robot i is defined by the tuple $\Pi = (s_t^i, A, \gamma)$ where γ is a goal formula. In plain language, the goal formula is completion of all tasks in the environment. The goal formulas are considered to be common knowledge as each robot will act according to the

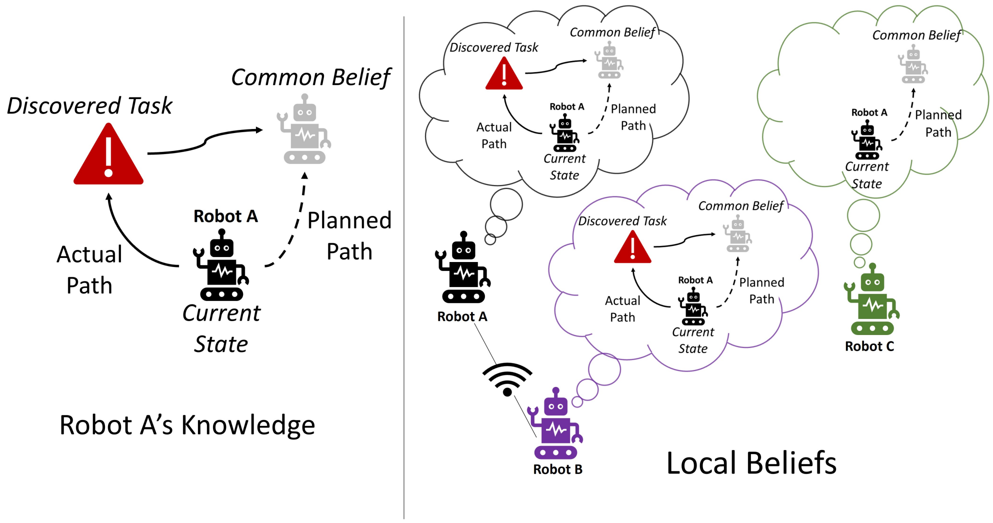


Figure 4.3: Illustration of a local belief update. Both robot A and robot B are connected and update their local shared belief, but retain the global common belief that achieves consensus with robot C.

same policies under the same conditions. Thus, we seek the following joint policy implementation, π to ensure the completion of all tasks in the environment. The reason we use joint policies is that robot needs to map indistinguishable epistemic states to the same actions. Therefore, we define the following important rulesets.

First, a robot i may discover a task requiring two robots and seek to communicate with a robot j by traveling to its last shared belief. Consequently, σ_t^i becomes *gossiping* and robot i travels to the particle with the most recent timestamp in the set $\{q_{ij,k} \forall k \in \mathcal{A}\}$. If robot j is not at its last shared belief, robot i's belief about j is incorrect and so three additional worlds are possible and indistinguishable given robot i's current knowledge. Except for exhaustively searching for robot j, robot i does not have any way to find j. The ruleset is to exclude j from its policy options π if its belief is false. This policy ruleset will cause the policy to potentially fail if more tasks are present in the environment than available robots such that $\sum_{\tau \in \mathcal{T}} N_\tau \geq N_r$.

Second, a robot i may discover a task, but believes robot j has also discovered the task first based on robot j's coverage assignments. Therefore, robot i assumes the task has been accomplished by j. Upon communication, this assumption is verified and the task is designated for completion if it has not been accomplished. Since this policy execution is finite and the last element satisfies the goal formula, γ.

Thus, our acceptable common knowledge policy rulesets are established. The execution of π is defined as a maximal sequence satisfying the global formula γ. The algorithm for this sequence is defined in the following section.

4.4 Epistemic Task Allocation & Gossiping

At the core of this framework is an epistemic-based multi-robot information dissemination and task allocation algorithm. As previously mentioned, in this chapter we focus on a subcategory of the MRTA problems known as the single-task multi-robot time-extended allocation problem. There are few mathematical models from combinatorial optimization research that tackle this further generalization of the assignment problem; however, the assignment problem can be modeled with joint, rather than per-robot constraints for each task such that the utility, $u(\cdot)$, is maximized. The solution to the following assignment problem is the execution sequence of policy π satisfying the epistemic goal formula γ for completing all discovered tasks in the environment.

$$\max \quad \sum_{i\in\mathcal{A}}\sum_{\tau\in\mathcal{T}} u_{i\tau}(t_{i\tau}(\boldsymbol{b}_i(\mathbf{y}_i)))y_{i\tau} \tag{4.13}$$

$$\begin{aligned} \text{s.t.} \quad & \sum_{i\in\mathcal{A}}\sum_{\tau\in\mathcal{T}} y_{i\tau} \geq N_\tau, \quad \forall\tau\in\mathcal{T} \\ & t_{i\tau}(\boldsymbol{b}_i(\mathbf{y}_i)) \geq t_{i\varsigma}(\boldsymbol{b}_i(\boldsymbol{y}_i)) + \delta_{\varsigma\tau} \ \forall(\varsigma,\tau)\in S_i \\ & t_{i\tau}(\boldsymbol{b}_i(\mathbf{y}_i)) \geq 0 \ \forall\tau\in\mathcal{T} \\ & x_{i\tau}\in\{0,1\} \end{aligned} \tag{4.14}$$

where $y_{i\tau} = 1$ if robot i is assigned to task τ and $\boldsymbol{y}_i = \{y_{i1}, \ldots, y_{iN_t}\}$. The arrival time for the i^{th} robot is a unique function, $t_{i\tau}$, that accounts for the arrival time of N_τ necessary robots for task τ. The variable $\delta_{\varsigma\tau}$ is the duration between tasks ς and τ. The order of tasks is represented by a directed graph, S_i, created by the order of robot i's path, $\boldsymbol{b}_i$, where an edge in S_i is (ς, τ) indicates that task ς is performed before task τ.

Additionally, when a task is discovered, a robot must consider if any assistance is required to complete a task, any tasks that are already in its queue, and prior communicated allocations of tasks to other robots. If assistance is required, the robot must disseminate the new information to neighboring robots, acting as an ad hoc network by visiting a neighboring robots belief state.

To account for these considerations, the following section describes each of three steps involved in our proposed algorithm: i) initial task bundling to assign each task to an robot, ii) makespan minimization to minimize the expected time to complete all tasks, and iii) a gossip protocol algorithm to optimize the assignment of information dissemination.

First, we require a valid initial solution for the task allocation problem. We define a robot's bundle as an ordered list of tasks to complete. Given that each task may require more than one robot, the allocation order requires that one task must be executed in the bundle order before another is

Algorithm 1 Initial Task Bundling Algorithm

Require: N_τ ▷ number required for task $\tau \in \mathcal{D}$
1: $B_j = \emptyset, \forall j \in \mathcal{A}$ ▷ initial bundle
2: **for each** $\tau \in \mathcal{D}$ **do**
3:     **for each** $j \in \mathcal{A}$ **do**
4:         Bid on task with utility $h(x_j, \tau)$
5:     **end for**
6:     $W_\tau = \{j \in \mathcal{A} : \quad |\{j' \in \mathcal{A} : h(x_{j'}) < h(x_j)\}| \leq N_\tau\}$
7:     **for each** $j \in \mathcal{A}$ **do**
8:         **if** $j \in W_\tau$ **then**
9:             $B_j \leftarrow B_j \bigoplus_{end} \tau$
10:         **end if**
11:     **end for**
12: **end for**

assigned. Thus, to accommodate this temporal constraint, we use a modified sequential-single item (SSI) auction for initial bundling as shown in Algorithm 1. The task bundling algorithm initializes an empty bundle for each robot and each robot bids on the first task in the set of locally discovered tasks, $\mathcal{D} \subseteq \mathcal{T}$. The highest N_τ bidders incorporate the task at the end of their bundle (lines 6-10).

If a robot is not connected to make a bid, the locally connected team member with the highest confidence (i.e. most recent information documented by the set $\mathcal{Q}$) of the state of the disconnected robot submits a bid on their behalf. The bid for adding task τ to robot j is defined by marginal improvement of robot j's bundle score. As such, the bid is defined as

$$h(x_j, \tau) = \lambda_\tau - S_j^{B_j \oplus_{end} \{\tau\}} \tag{4.15}$$

where $S_j^{B_j}$ is initialized to $S_j^{\emptyset} = 0$ and denotes the cost of travelling given the original bundle, B_j, and the added task. The operator $\bigoplus_{end}$ adds the antecedent task τ to the end of its precedent bundle, B_j. This decentralized algorithm allows connected robots to quickly create a valid task allocation, but does not account for the completion time of every task. Thus, minimizing the makespan of the bundle order will reduce the task allocation's estimated completion time.

Makespan is the time it takes for all robots to finish all of their assigned tasks [63]. Attempting to minimize the makespan of the bundled tasks accounts for a scenario where a robot can complete a task "on the way" to another task and complete all assigned tasks faster. Algorithm 2 gives an overview of the makespan minimization algorithm.

The algorithm iterates through all tasks in each robot's bundle and places each task in each available path segment. Then the makespan is calculated for the robot's new bundle order, B_{tmp}. If the new makespan, m_{tmp} is smaller than the best makespan, m_{best}, the bundle B_j is replaced with

Algorithm 2 Makespan Minimization

Require: $B_j \ \forall j \in \mathcal{A}$; $m_{best} = makespan(B_j)$

1: **for each** $j \in \mathcal{A}$ **do**
2:     $B_{tmp} \leftarrow B_j$
3:     **for each** $j' \in B_j$ **do**
4:         $B_{tmp} \leftarrow B_j \setminus j'$
5:         **for each** $n \in \text{len}(B_j)$ **do**
6:             $B_{tmp} \leftarrow B_{tmp} \bigoplus_n j'$
7:             $m_{tmp} = makespan(B_{tmp})$
8:             **if** $m_{tmp} < m_{best}$ **then**
9:                 $B_j \leftarrow B_{tmp}$
10:                 $m_{best} \leftarrow m_{tmp}$
11:             **end if**
12:         **end for**
13:     **end for**
14: **end for**

Algorithm 3 Gossip Protocol Auction

Require: robots $j_g \in \{G_j : B_j \neq \emptyset\}$

$D = r_c$ where r_c are the connected robots
$G_{B_j} = \emptyset$ is the gossiping assignments for robot j given bundle B_j
while $D \neq G_j$ **do**
    for each $j_g \notin D$ **do**
        for each $j_v \in D$ **do**
            $b_{vg} = bid(j_v, j_g)$
        end for
    end for
    for each $j_v \in D$ **do**
        $g^* = \text{argmax}_g(b_{vg})$
        if $j_{g^*} \notin D$ **then**
            $G_{B_{j_v}} = G_{B_{j_v}} \bigoplus_{end} j_{g^*}$
            $D \leftarrow D \bigoplus_{end} j_g$
        end if
    end for
end while

B_{tmp}. Note that the order of tasks that were previously communicated to now disconnected robots must be maintained in Algorithm 2 by not reordering these tasks in the makespan minimization (lines 3-13).

The ordered bundle for each robot would typically be the execution sequence for policy π to complete the NP-Hard problem defined in 4.13, but given the communication restriction, communication assignments must also be allocated for every robot to perform its sequence of tasks. For this reason, we introduce the gossip protocol assignment algorithm.

If a robot is assigned to a task, but is not aware of the new information, robots in charge of the allocation must deliver the information, acting as an ad hoc network and informing the necessary team of robots through a gossip protocol based algorithm, accounting for the cascading effect of communication and adding nodes to the ad hoc network. Algorithm 3 steps through the allocation of peer-to-peer communication tasks based on the resulting task allocation from Algorithm 2. Similar to bids in Algorithm 1, the robots with the most recent state information for a disconnected robot will submit bids on their behalf.

First, the set G_j is defined as the robots who are assigned a task in the bundle. The variable D represents the set of robots who either know the information to be disseminated or a robot has been assigned to communicate with them. The set G_{B_j} is initialized as the currently connected robots in G_j and a new empty gossip bundle is established for all robots (lines 1-2). Next, each required robot, j_g, is bid on by a robot, j_v in the D set (lines 4-8). The highest bid for robot $j_g \notin D$ is added to robot j_v's bundle and j_g is added to the D set (lines 9-15). The while loop repeats until all necessary robots for B_j have been assigned and accounts for the cascading effects of communication (i.e. when a robot has communicated with another robot, two robots are now available to gossip to other members).

After execution of these algorithms, the execution policy for a robot i is represented as a sequence that is defined by the concatenation of its gossip bundle G_{B_i} and task bundle B_i. A robot is responsible for its communication assignments before continuing to its ordered task execution. The ordered sequence for every robot is the execution of π satisfying the goal formula γ given its current epistemic state s_t^i.

Task allocation, makespan minimization, and coverage of an environment are all NP-Hard problems; however the worst-case computational complexity of the above algorithms are $O(N_a^2 N_t^3)$. By using these algorithms we can create a feasible solution for single-robot multi-task allocation and subsequent gossip protocol.

4.5 Epistemic Planning for Heterogeneous Robot Teams

Lastly, we build on the work in the previous sections of this chapter and consider fault and disturbance tolerant task allocation for heterogeneous robotic systems. This method requires a solution that considers the heterogeneous makeup of robots in the system and the required capabilities to complete certain tasks in the environment. Due to the heterogeneous nature of the system, we first introduce a common meeting place that allows robots with lesser capabilities to converge with the location of all other robots once the exploration task is complete. The robot i's particles converge to a common meeting place using the mean location of the particles in the set $\mathcal{C}_i$:

$$\sum_{c_{ij} \in \mathcal{C}_i} c_{ij} / |\mathcal{C}_i| \tag{4.16}$$

where $|\cdot|$ signifies the cardinality of a set.

In addition, the epistemic planning task, Π, demands an execution policy, π that satisfies the mission objective. We consider the solution to a planning task as a joint policy so that each robot is responsible for subtasks within the mission objective. The solution to such a policy is solved using the below nonlinear integer program where the utility, $u(\cdot)$, is maximized and we note the addition of the first constraint that considers the robots capabilities.

$$\begin{aligned}
\max \quad & \sum_{i \in \mathcal{A}} \sum_{\tau \in \mathcal{T}} u_{i\tau}(t_{i\tau}(\boldsymbol{b}_i(\boldsymbol{y}_i), s_t^i)) y_{i\tau} && (4.17)\\
\text{s.t.} \quad & \sum_{i \in \mathcal{A}} \sum_{\tau \in \mathcal{T}} y_{i\tau} \mathbf{I}_i^k \geq r_\tau^k, \quad \forall \tau \in \mathcal{T},\ \forall k \in \{1, \ldots, N_k\} \\
& t_{i\tau}(\boldsymbol{b}_i(\boldsymbol{y}_i), s_t^i) \geq t_{i\varsigma}(\boldsymbol{b}_i(\boldsymbol{y}_i), s_i^t) + \delta_{\varsigma\tau} \ \forall (\varsigma, \tau) \in S_i && (4.18)\\
& t_{i\tau}(\boldsymbol{b}_i(\boldsymbol{y}_i), s_t^i) \geq 0 \ \forall \tau \in \mathcal{T} \\
& y_{i\tau} \in \{0, 1\}
\end{aligned}$$

where $y_{i\tau} = 1$ if robot i is assigned to task τ and $\boldsymbol{y}_i = \{y_{i1}, \ldots, y_{iN_t}\}$. $\mathbf{I}_i^k$ is an indicator function for the robot i that has the k^{th} capability. The arrival time for the i^{th} robot is a unique function, $t_{i\tau}$, that accounts for the arrival time of r_τ^k necessary robots with capability k for task τ. Additionally, the function $t_{i\tau}$ considers the current belief using the epistemic state s_t^i. The variable $\delta_{\varsigma\tau}$ is the duration between tasks ς and τ. The order of tasks is represented by a directed graph, S_i, created by the order of robot i's path, $\boldsymbol{b}_i$, where an edge $(\varsigma, \tau) \in S_i$ is indicates that task ς must be performed before task τ.

We use the matrix representation shown in Fig. 4.4 to represent the solution space from (4.17). We refer to the time of a task as an epoch in which any robot is available to perform an additional task. Using this representation, we can check precedence constraints such as gossiping to a robot before assigning it tasks to accomplish or simultaneous tasks requiring multiple robots at the task location (e.g. a UGV opening a door for a UAV to fly through). All matrix representations for one iteration of the task allocation problem are of static depth along the time axis and initialized to be the number of tasks possible to complete (i.e. gossiping tasks plus discovered tasks).

Though any method for solving a constrained nonlinear assignment problem can be used, we apply a genetic algorithm (GA) due to the sparsity of the decision space and the binary assignment constraints. Specifically, we generate an initial feasible population for the task allocation problem

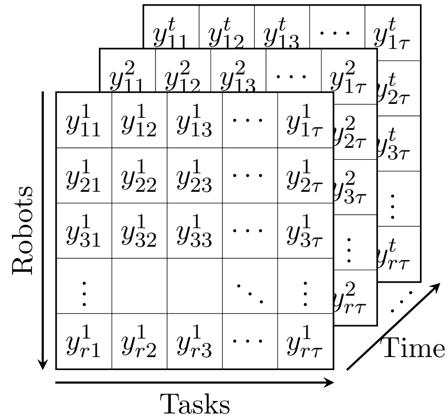


Figure 4.4: Pictorial representation of binary decision matrix.

Algorithm 4 Feasible Solution Generation

Input: $\Pi = (s_t^i, A, \gamma)$

Output: Solution representing policy, π for Π

v_0 is the set of connected robots and $v_r = v_0$
while $\tau \in \mathcal{T}_D$ not complete **do**
    for each $\tau \in \mathcal{T}_D$ **do**
        if v_r can perform the task **then**
            if $v_r \equiv \mathcal{A}$ or rand() >threshold **then**
                $v_t \in_R v_r$ for each task capability
                $\boldsymbol{b}_j \leftarrow \boldsymbol{b}_j \oplus$ task $\forall j \in v_t$
            end if
        end if
    end for
    if $\tau \in \mathcal{T}_D$ is complete **then**
        break
    end if
    for each $v_r \notin v_t$ **do**
        $v_g \in_R \mathcal{A} \setminus v_r$
        $\boldsymbol{b}_{v_r} \leftarrow b_{v_r} \oplus v_g$
    end for
    $v_r = v_r \oplus v_g$
end while
$\pi \leftarrow \bigcup_{j \in \mathcal{A}} \boldsymbol{b}_j$

---

and then utilize a GA to efficiently sample the decision space.

Since the assignment problem must be solved at runtime, the initial population is generated using the method in Algorithm 4 to warm-start the GA where $\in_R$ indicates a uniformly selected element. The operator $\oplus$ appends the antecedent set to the precedent set. The output policy π is a sequence defined by the joint execution order $\{\boldsymbol{b}_j \ \forall j \in \mathcal{A}\}$ and represented as a sequence of epistemic states and robot action pairs, $\pi = (s_t^i, (j : a), s_{t+1}^i, \ldots)$. The algorithm is run for the desired size of the initial GA population.

y_{11}^1	y_{12}^1	$\cdots$	$y_{1\tau}^1$	y_{11}^2	$\cdots$	$y_{1\tau}^t$	y_{21}^1	$\cdots$	$y_{r\tau}^t$

$\longrightarrow$

Robots, Time, & Tasks

Figure 4.5: GA solution representation

Additionally, the constrained optimization function in (4.17) is transformed into an unconstrained, penalty-based function such that

$$val = \sum_{i \in \mathcal{A}} makespan(\boldsymbol{b}_i) + V_i \tag{4.19}$$

where $makespan(\boldsymbol{b}_i)$ is the estimated length of time for robot i to complete its assigned tasks and V_i is the penalty for violated constraints in (4.18). Since only policies where no constraints are violated are valid for the goal formula γ, V_i must be set at a high value to ensure that the selected solution of the GA is feasible. Algorithm 4 ensures that the initial population of solutions are all valid policies for the goal formula γ, but we use the GA to iterate our solution and attempt to achieve a higher fitness value.

The resultant solution or gene is shown generically in Fig. 4.5 where the highest fitness value is the execution policy for the robots connected locally. To achieve the highest fitness value at the lowest computational complexity, the chromosome is formatted as a single row sparse matrix. By using this representation we can utilize point mutation, one-point crossover, and roulette wheel selection [19] such that the computational complexity of our task allocation algorithm is $O(nm^2)$ with n being the number of robots and m being the number of tasks.

Example. To reinforce the proposed approach in the reader's mind, consider the scenario in Fig. 4.6 where all robots know there is one task at an unknown location (e.g., a search and rescue mission). Fig. 4.6(a) shows the robots disconnecting and beginning to propagate belief states. UGV 2 experiences a failure and moves to the second particle. In Fig. 4.6(b), UAV 1 finds a task that requires one aerial vehicle and one ground vehicle and uses the first particle of UGV 2 in the allocation optimization. In Fig. 4.6(c), UAV 1 observes that UGV 2 is not tracking its first particle and reallocates using UGV 2's second particle. UAV 1 is able to communicate with UGV 2 at its second belief particle and re-optimizes assignments using the new information. This sends UGV 2 back to base and assigns UGV 1 to perform the task shown in Fig. 4.6(d). In Fig. 4.6(e) UAV 1 and UGV 1 plan to perform the task and complete the task in Fig. 4.6(f) before relocating to the base.

In general, employing this framework allows a heterogeneous MRS to propagate beliefs, reason about environmental changes, and plan according to local observations while disconnected. The product of our approach is a sequential policy that is formulated using belief states to reason about other robots location and used to accomplish any tasks discovered in the environment.

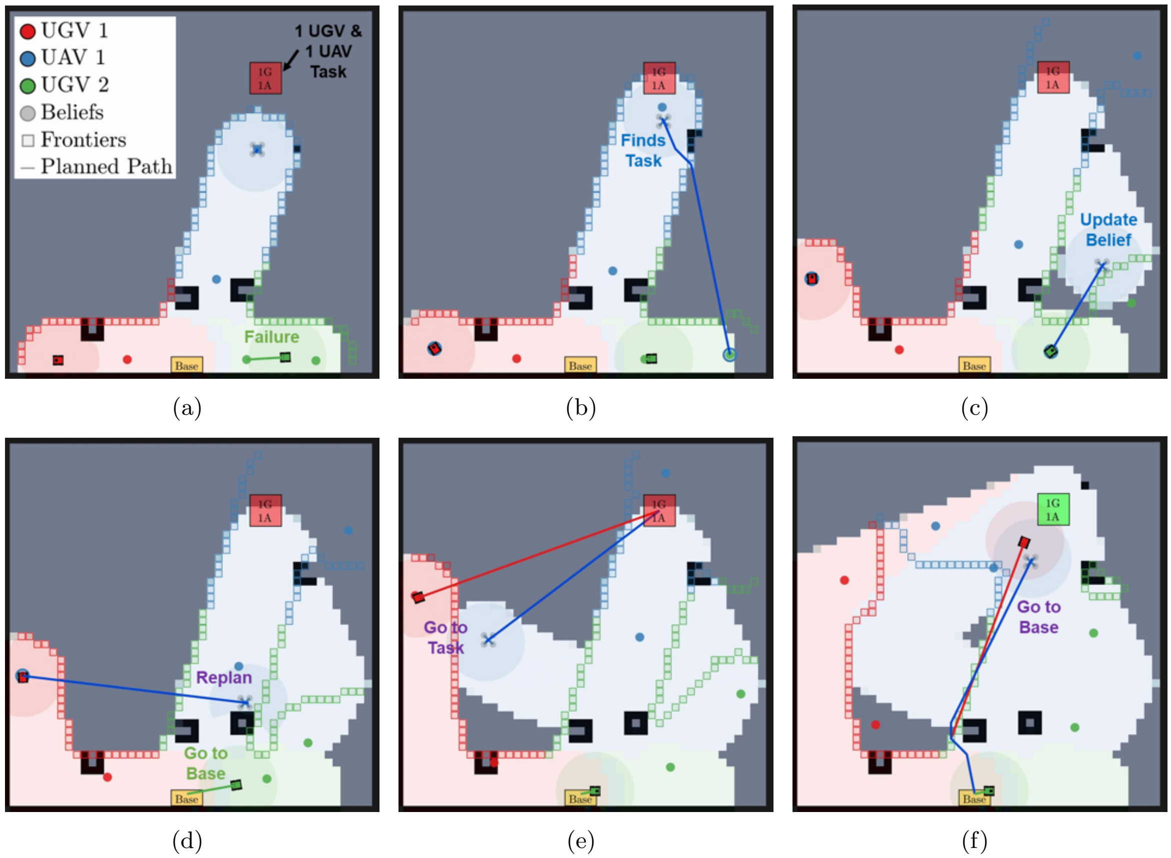


Figure 4.6: Example simulation with one known task at an unknown location.

4.6 Simulation and Experiment Results

In this section, we provide comparisons from MATLAB simulations with our approach implemented on two case studies. Case Study I is a simulated scenario where all robots know that only one task exists in the partially known environment requiring an unknown number of robots at an unknown location. Case Study II is a simulated scenario similar to Case Study I, but there are an unknown number of tasks that, in total, do not require more robots than are available, $\sum_{\tau \in N_t} N_\tau \leq N_a$. Simulations were performed on 15 randomly generated cluttered 50m $\times$ 50m environments with 10-20 initially unknown obstacles.

The proposed approach is compared against two other methods. The first method applies a constant connectivity constraint, not allowing agents to travel outside of a 10m communication range. The second method assumes ideal conditions, where robots can communicate across the entire environment. In the following section, we refer to the first method as the "flock" method and the second method as the "ideal" method. Both methods use smooth A* path planning and an artificial potential field technique for controlling the robots towards uncovered regions and away from obstacles. In all methods, the maximum velocity is 3m/s, the simulated LiDAR range is 5m,

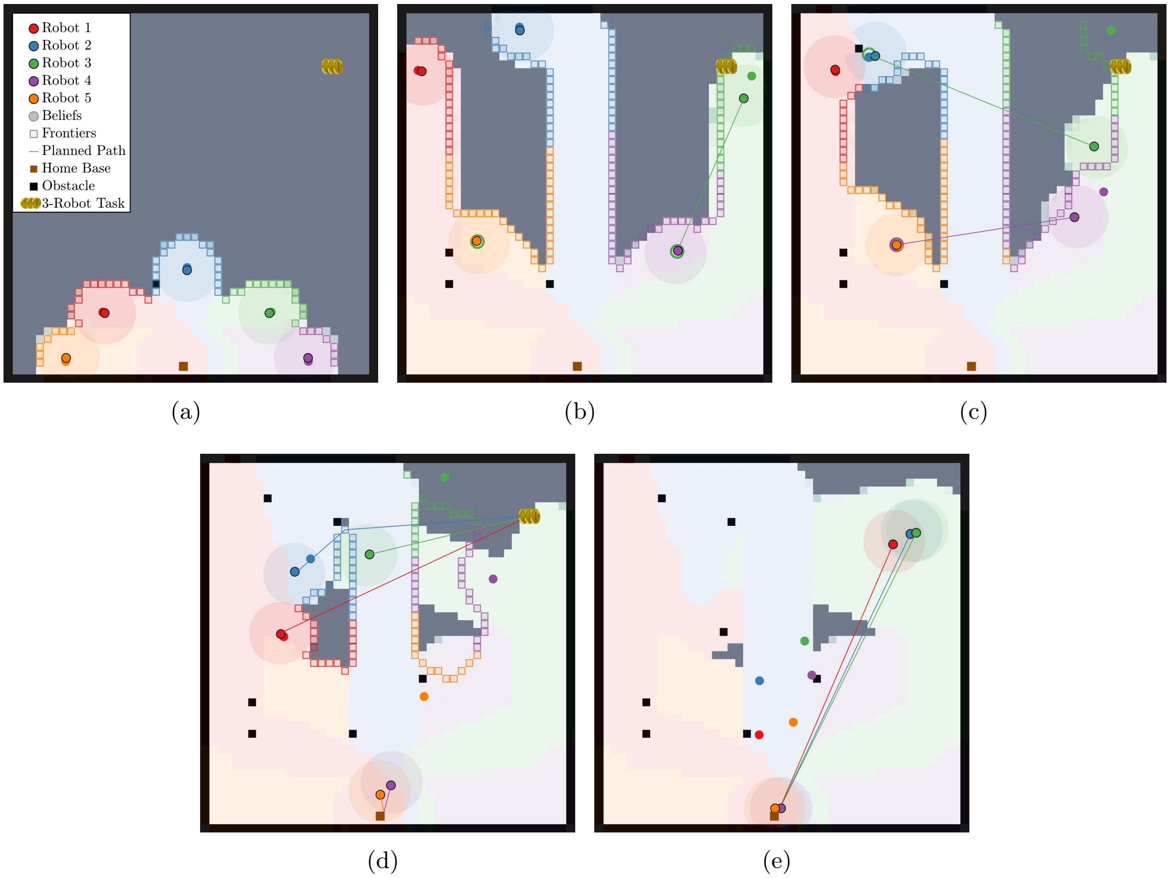


Figure 4.7: This figure depicts the progression of the Case Study I simulation where there is a single task in the environment. (**A**) shows the starting state of the robots after initial disconnection. (**B**) shows robot 3 finding the 3-robot task and deciding to communicate with robot 4. (**C**) shows a successful communication and replanning with robot 4 tasked with communicating to robot 5 and robot 3 to robot 2. In (**D**), all robots have their final assignment with robots 1, 2, and 3 assigned to the task while robots 4 and 5 route to home base. (**E**) shows that with the task completed, all robots are routed home.

and the robots' motion is modeled with a single-integrator model with $\boldsymbol{\nu} \sim N(0, 0.2)$ from (3.1).

4.6.1 Case Study I: *Multi-Robot Single Task*

In Case Study I, all robots are aware that only one task is located in the environment. This simulated scenario is similar to a search and rescue mission where the goal is to locate and rescue an individual at an unknown location in a large environment. The robots begin by covering the area and, upon discovery, calculate how many robots are necessary for the rescue operation. Then, the robot disseminates the information to the rest of the robot team who either are tasked with returning to home base or assisting in the rescue. An example scenario is shown in Fig. 4.7. The Case Study I example showcases the method when only one task is in the environment. The robots begin by disconnecting to more efficiently cover the environment. Robot 3 finds a task in Fig. 4.7(b) and plans using its knowledge of each robot's epistemic state. The result is for robot 3 to communicate with robot 4 via robot 4's belief state. After communication is successful in Fig. 4.7(c), robots 3 and 4 similarly plan to communicate with robots 2 and 5 via their respective belief states. Lastly, all robots are assigned to the final task in Fig. 4.7(d) and complete the task in Fig. 4.7(e) before routing to home base.

This scenario was implemented using two, three, five, and eight robot teams in 15 varying environments. The results of the simulated method comparisons are shown in Fig. 4.8.

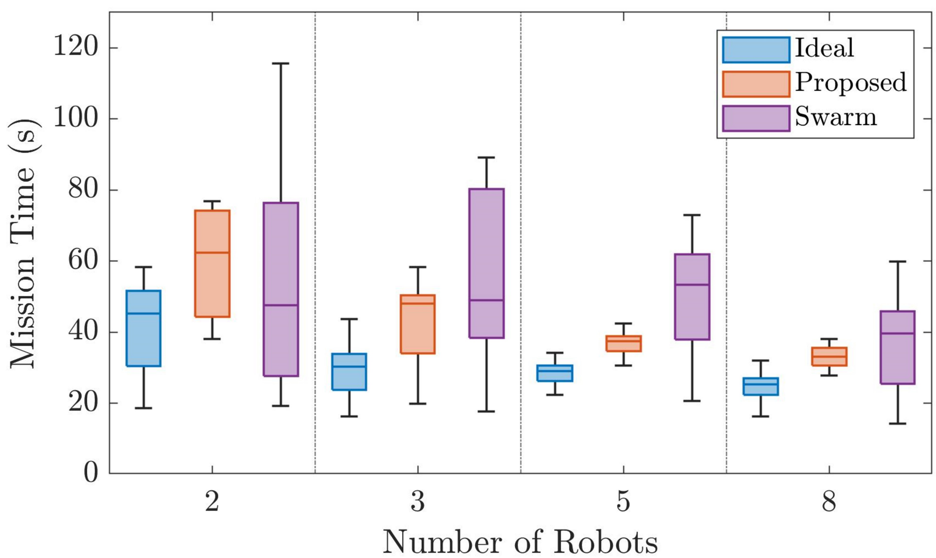


Figure 4.8: Figure comparing the results of the simulated scenarios for Case Study I. The proposed framework is measured against two other methods and shown to decrease the variance over random environments and decrease mission time as the robot team grows in size when compared to the always connected flock.

The figure illustrates the proposed framework's performance given a variety of environments and team sizes. The proposed method decreases the variance in the mission time with a two robot team, but is outperformed by the flock method since the robots remain together and can become lucky, finding the task and completing the mission. This method even outperforms our ideal scenario in some cases since the robots must potentially travel a longer distance to the task once found by a team member. However, as the robot teams become larger, the flock method is outclassed by more efficient coverage of the environment, represented by the ideal and proposed case. We also notice that the variance in mission times of the proposed and ideal methods are similar with a standard deviation of 11s and 14s, respectively across all robot team sizes. In comparison, the standard deviation of the flock method is 48s. Additionally, though initially outperformed with a team size

(a) (b) (c) (d)

Figure 4.9: This figure illustrates the progression of a Case Study I experiment where there is only one task. (**A**) shows the starting state of the robots after initial disconnection. (**B**) shows robot 1 finding the 2-robot task and deciding to communicate with robot 2. (**C**) shows a successful communication and (**D**) shows task completion and robots' route to their home base.

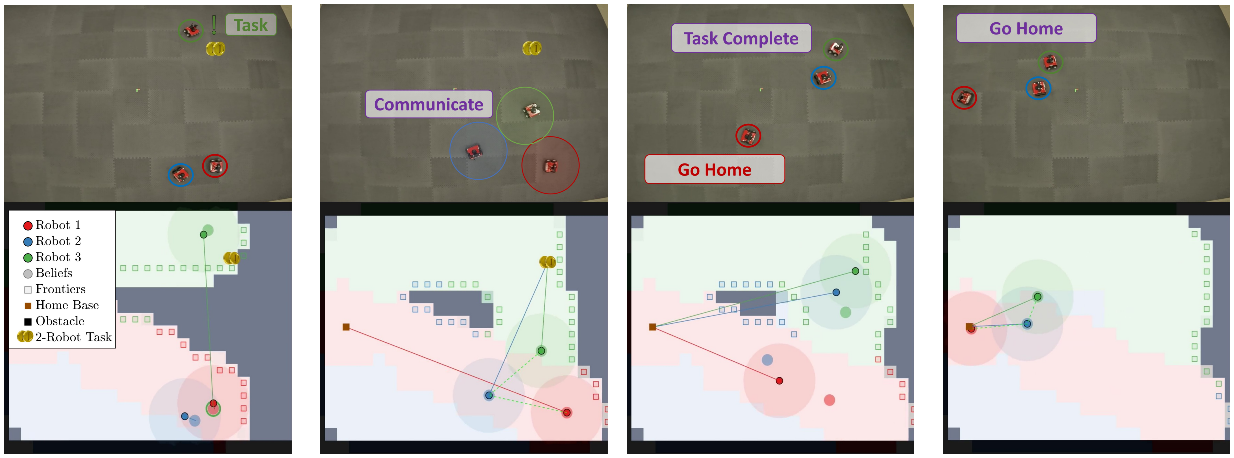
of two robots, the proposed method on average outperforms the flock method by 13s. The ideal method also is 11s faster on average than the proposed method.

The proposed approach was also validated through laboratory experiments with a multi-robot team. The team consists of two to three Husarion ROSbot 2.0 UGVs using a Vicon motion capture system. The experiments effectively demonstrate all parts of the proposed approach, including intentional disconnections, searching, and gossiping behaviors. In all experiments, the UGVs start within communication range and are tasked to cover the environment and complete any discovered tasks. Experiments were performed in a 4m$\times$5.5m space containing convex obstacles considering, as a proof of concept, a sensing and communication range for each robot of 1m.

Displayed in Figure 4.9 are the results from an experiment with a two-robot team. The columns of Fig. 4.9 correspond to different instances within the experiment, and each row from top to bottom shows snapshots of the robots at different times throughout the experiment and the current map of the environment covered by the team.

As shown in the figure, the robot team initially disconnects to more efficiently cover the environment. In Fig. 4.9(b), robot 1 finds a task and plans to gossip the new information to robot 2. Fig. 4.9(c) shows the robots communicating and traveling to complete the task. Lastly, in Fig. 4.9(d), the robots complete the task and return to their home base.

Additionally, we show a Case Study I experiment with a three-robot team. Unknown obstacles were not included in three-robot experiments due to limited space, but the method remained the same for the entire duration of the experiment. The columns of Fig. 4.10 correspond to different



(a) (b) (c) (d)

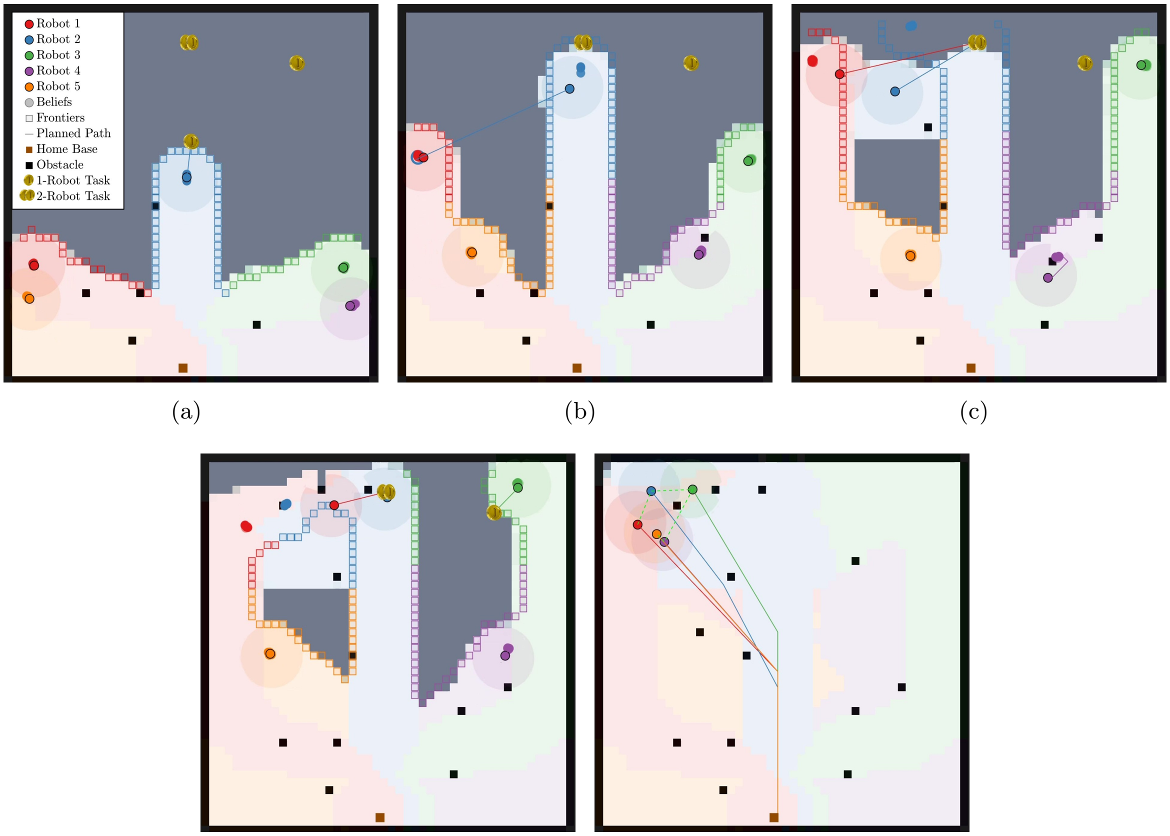
Figure 4.10: This figure illustrates the progression of a Case Study I experiment with a three-robot team and one task. (**A**) shows robot 3 finding a 2-robot task. (**B**) shows robot 3 communicating to robot 1 and 2 and allocating robot 2 and 3 to the task while sending robot 3 home. (**C**) shows a successful task completion and in (**D**) all the robots return to their home base.

instances within the experiment, and each row from top to bottom shows snapshots of the robots at different times throughout the experiment and the current map of the environment covered by the team.

As shown in the figure, the robot team initially disconnects to more efficiently cover the environment. In Fig. 4.10(a), robot 3 finds a task and plans to gossip the new information to robot 1. Fig. 4.10(b) shows the robots communicating and allocating robots 2 and 3 to complete the task while robot 1 returns home. In Fig. 4.10(c) the task is completed, and, in Fig. 4.10(d), all robots return to their home base.

4.6.2 Case Study II: *Multi-Robot Multi-Task*

In Case Study II, all robots do not know how many tasks are in the environment, where the tasks are located, or how many robots are required at each task. This simulated scenario is a recovery of an asset that may be scattered across a large environment. The robots begin by covering the area and, upon discovery of a task, calculate how many robots are necessary for the rescue operation. Then, the robot disseminates the information to the necessary members of the robot team. If a robot is not able not able to be located at its believed location, the robot considers this robot occupied and does not consider it in the next iteration of assignments. The robots must cover the entire environment in order to identify if any tasks lie in the uncovered portions of the environment. Fig. 4.11 presents an example scenario from the comparison scenarios. Fig. 4.11 exhibits a scenario with three tasks at unknown locations. Two tasks require one robot and one task requires two



(a) (b) (c)

(d) (e)

Figure 4.11: This figure illustrates the progression of the Case Study II simulation where the number of tasks are unknown. (**A**) shows the starting state of the robots after initial disconnection. (**B**) shows robot 1 finding the 2-robot task and deciding to communicate with robot 2. (**C**) shows a successful communication and (**D**) shows task completion and the robots route to their home base (**E**).

robots. The individual tasks are completed upon discovery by the closest robot. When robot 2 discovers the two robot task in Fig. 4.11(b), it routes to robot 1's belief state to ask for assistance. Both robots travel to complete the task in Fig. 4.11(c) and robot 3 finds the last single robot task in the environment in Fig. 4.11(d). After all portions of the environment have been covered, all robots route to their home base.

Case Study II was also implemented using two, three, five, and eight robot teams in 15 varying environments. The results are shown in Fig. 4.12.

This Figure displays a strong argument for the proposed method when compared to the flock and ideal methods. On average, the proposed method performed only 16s slower than the ideal method, even considering the communication limitation. Additionally, the proposed method was 35s faster than the flock method. Though the variance for the proposed method was higher than the flock and ideal methods, its worst case mission time is approximately as good as the median

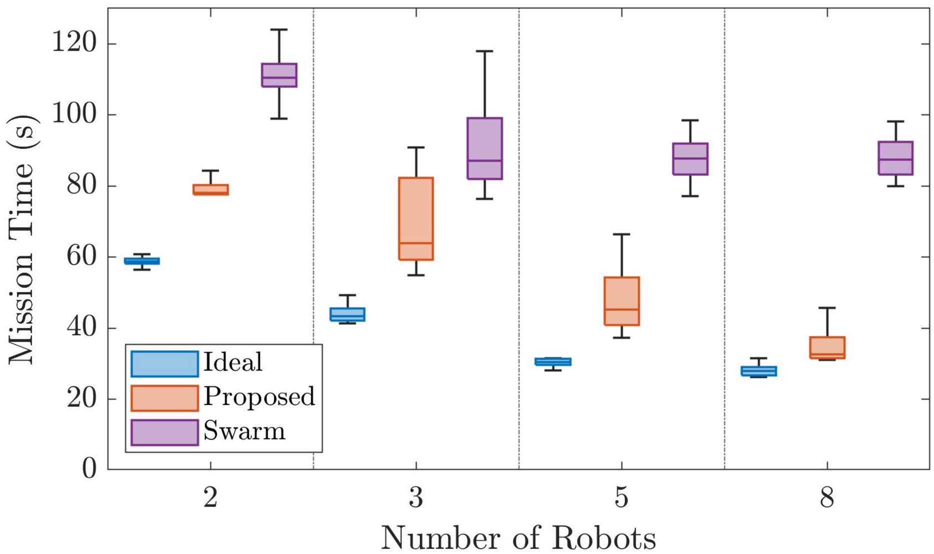


Figure 4.12: Figure comparing the results of the simulated scenarios for Case Study II. The proposed framework is shown to decrease mission time drastically when compared to the always connected flock and perform similarly to the ideal method as team size increased.

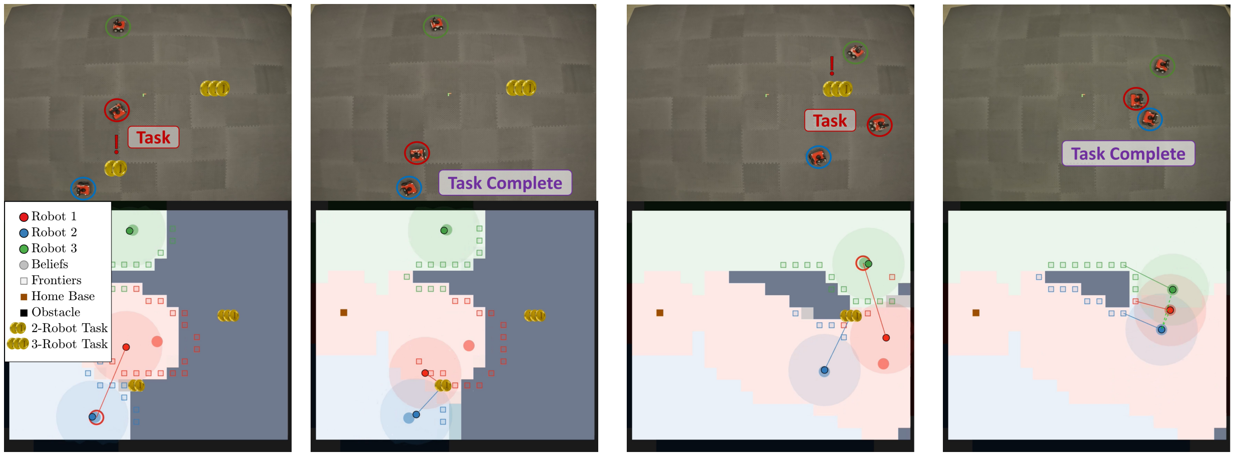
performance of the flock method.

We also show an example experiment with a three robot team in Fig. 4.13. Both the locations of the obstacles and the number of tasks to complete are unknown here. Therefore, the robots are tasked with covering the environment, gossiping to necessary team members, and completing discovered tasks. The environment has two tasks, one that requires two robots and one that requires three robots to complete.

In this example experiment, robots intentionally disconnect to cover the environment more efficiently. After a short time, robot 1 finds a task that requires two robots. Robot 1 plans to tell its believed closest neighbor and gossips to robot 2 in Fig. 4.13(a). In Fig. 4.13(b), the robots who know about the task complete it and travel back to their global belief states. Fig. 4.13(c) shows robot 1 finding a three robot task while connected to robot 2. Robot 2 is allocated immediately to the task and robot 1 is tasked with gossiping the new discovery to robot 3. All robots converge to complete the task and finish covering the environment before returning to home base in Fig. 4.13(d).

4.6.3 Case Study III: *Heterogeneous Task Extension*

In this section, we provide results and comparisons from MATLAB simulations with our approach implemented on two, three, five, and eight robot teams. Each scenario has a randomly generated team makeup consisting of two types of vehicles, UAVs and UGVs, for each run. Simulations were



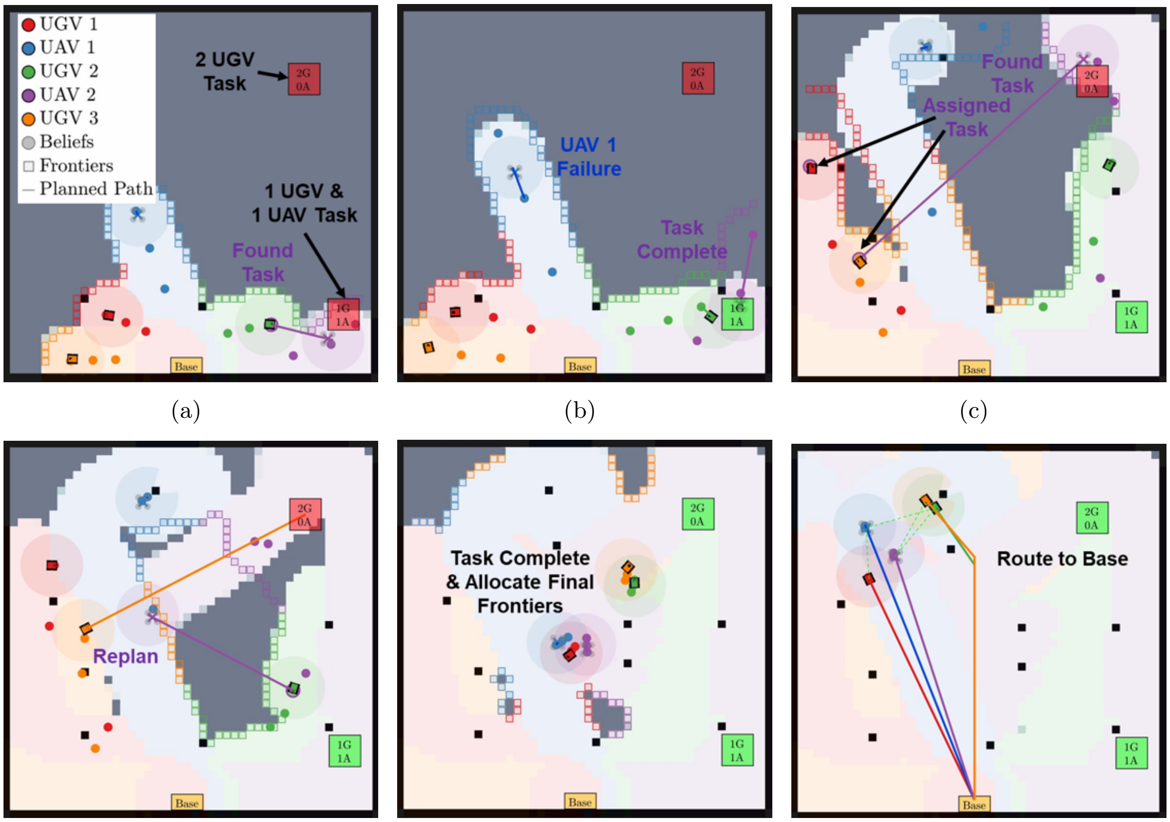
(a) (b) (c) (d)

Figure 4.13: This figure illustrates the progression of a Case Study II experiment where the number of tasks are not known. (**A**) shows the robot 1 finding a 2-robot task after initial disconnection and planning to communicate with robot 1. (**B**) shows robot 1 and robot 2 completing the task and planning to return to their belief states. (**C**) shows robot 1 and robot 2 finding a 3-robot task while connected and planning for robot 1 to communicate with robot 3. (**D**) shows the 3-robot task completed. Subsequently, all agents finish coverage of the environment and return to the home base.

performed in 15 random 50m $\times$ 50m environments with 5-15 initially unknown obstacles. The robots start by assuming that the environment has no obstacles and do not know the location of the tasks.

Each robot propagates three particles. Particles are propagated on the basis of the maximum speed of the vehicle type. A UGV may travel at a max speed of 2m/s, and a UAV may travel at 6m/s. The second and third particles travel at a linear speed decreased from the vehicle's maximum speed of 40% and 80%, respectively. The particles propagate according to these velocities, and the maximum communication range is 10m from the center of the robot. Within our simulations, each scenario was run with zero, one, and two failures that can happen to any robot at any time, causing the affected robot to track its second or third empathy particle.

Fig. 4.14 showcases an example of the simulation scenarios run in this section. As shown, there are two tasks in the environment, but all robots begin not knowing how many tasks or the location. Fig. 4.14(a) shows that after disconnection, one task requiring a ground and aerial vehicle is discovered by UAV 2 who communicates the task to UGV 2 and both complete the task in Fig. 4.6(b). In Fig. 4.6(c), UAV 2 finds a task requiring two ground vehicles. UAV 2 allocates UGV 1 and UGV 3 to the task initially, but after updating the states and replanning upon connection with UGV 1 in Fig. 4.14(d), UAV 2 gossips to UGV 2. Both UGV 1 and UGV 2 route to the task and complete it in Fig. 4.6(e) before connecting with all robots in the system at the common meeting location and partitioning the remaining frontier. In Fig. 4.14(e), no frontier points remain and all robots route to base.



(a) (b) (c)

(d) (e) (f)

Figure 4.14: Example simulation with one known task at an unknown location.

The proposed approach is compared with two other methods. The first method, referred to as the "flock method", applies a constant connectivity constraint, restricting all robots to travel within a 10m communication range of each other. The second "ideal method" assumes that robots can communicate across the entire environment. Both methods use the smooth A* and APF method to control the robots towards uncovered regions and away from obstacles. In all methods, the vehicles' simulated LiDAR range is 5m and the genetic algorithm is used for task allocation.

As shown in Fig. 4.15 and in Table 4.1, the proposed method outperforms the flock method in all scenarios. Additionally, the average coverage time for the proposed method is similar to the

Table 4.1: Average mission times by simulated failure scenarios.

	Number of Faults		
Method	**0**	**1**	**2**
Ideal	27.968s	30.432s	32.884s
Proposed	40.158s	48.63s	53.097s
Flock	113.4s	156.79s	179.31s

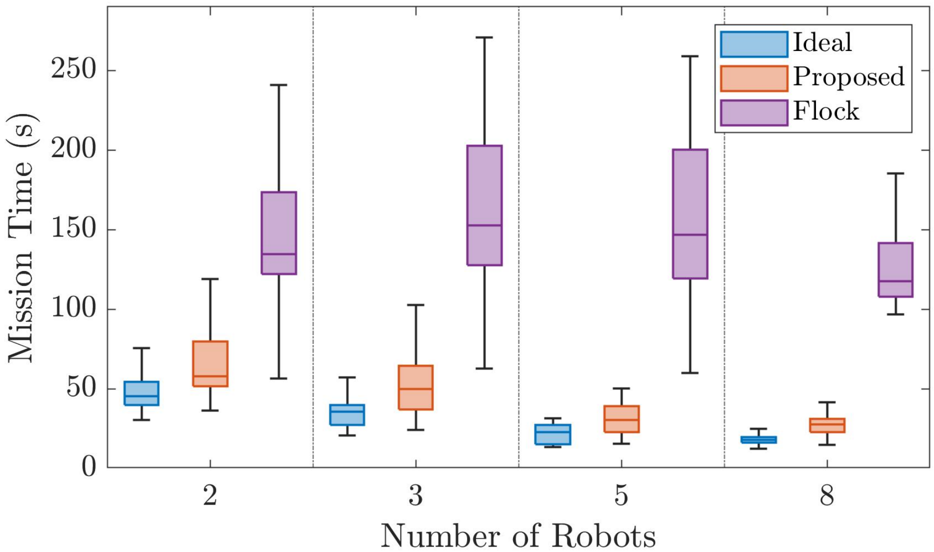


Figure 4.15: Comparison of the simulated scenarios.

ideal method, even with the communication limitation for the simulated failure scenarios.

The heterogeneous extension was also validated through laboratory experiments with a multi-robot team. The team consists of one to two Husarion ROSbot 2.0 UGVs and one Bitcraze Crazyflie 2.1 using a Vicon motion capture system.

The experiments effectively demonstrate all parts of the proposed extension, including intentional disconnections, coverage, gossiping, and task completion behaviors. In all experiments, the vehicles start within the communication range and are tasked to cover the environment and complete any discovered tasks.

Experiments were performed in a 4m×5.5m space containing convex obstacles and using, as a proof of concept, a sensing and communication range for each robot of 0.5m. In Fig. 4.16, we show the results of a sample experiment where there are three tasks in the environment, but the total number of tasks and their locations are unknown. Two tasks require one ground vehicle, and one task requires an aerial and a ground vehicle simultaneously.

As shown in the figure, each robot is assigned a partition of the frontier in Fig. 4.16(a) that encourages the robots to disconnect. Once disconnected, UAV 1 finds a ground and aerial vehicle task and a ground vehicle task. UAV 1 plans to communicate these tasks with UGV 2's common belief or first particle, but UGV 2 has experienced a fault and is now tracking the third particle. Fig. 4.16(c) shows UAV 1 reasoning about the location of UGV 2 by iterating through its beliefs before finding UGV 2 at the third particle and communicating the tasks. In Fig. 4.16(d), UGV 2 and UAV 1 complete the simultaneous task and UGV 2 plans to go to the ground vehicle task while UAV 1 plans to return to its common belief particle. In Fig. 4.16(e), all vehicles' common

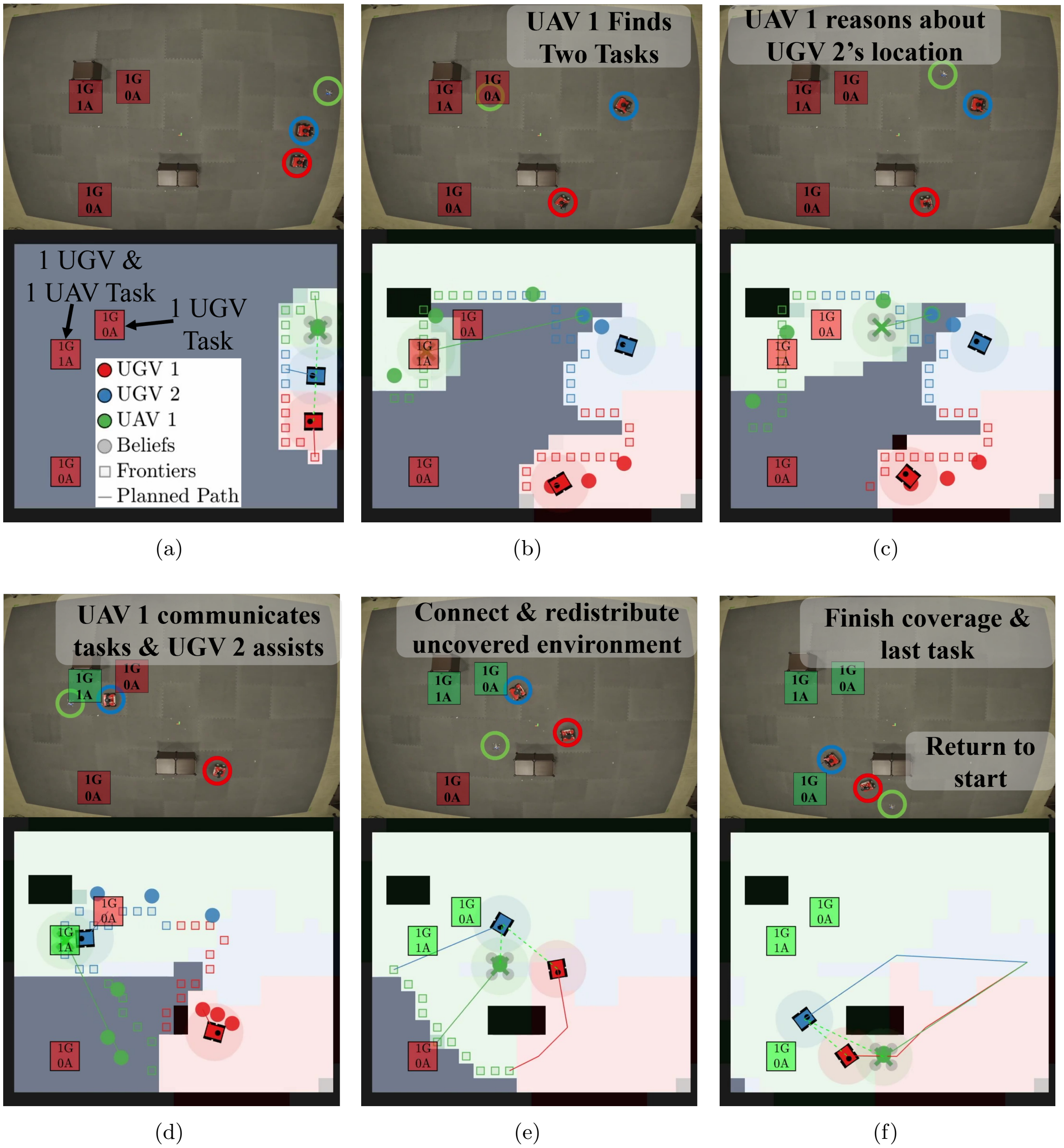


Figure 4.16: Snapshots and results of an experimental case study.

belief particles converge to a common meeting place from (4.16) and plan to cover the remaining environment and completing the last task before returning to their initial position in Fig. 4.16(f).

4.7 Discussion

In this chapter, we have introduced an innovative framework for multi-robot systems that utilize epistemic planning, enabling each robot to integrate depth-of-reasoning into its mission planning framework. The proposed approach allows a multi-robot system to disconnect and collaboratively plan based on a set of belief and empathy states. These beliefs are propagated using a frontier-based method for covering a partially unknown environment and are updated through dynamic epistemic logic and planning. The dynamic epistemic task allocation algorithm employs these belief states to allocate tasks discovered in the environment, fulfilling the epistemic planning task. This facilitates dynamic task allocation even while disconnected. A robot then plans to communicate the allocation by moving to the belief state.

Through our simulations and experiments, we demonstrate the effectiveness and relevance of our method when compared to perfect communication and a conventional flocking method that requires robots to stay within the communication range of the multi-robot system. Our findings, given an unknown number of tasks in the environment, indicate a significant reduction in mission time relative to the flocking method and similar results to scenarios with constant communication. In the single-task case study, we also observed an improvement in overall mission time and a substantial reduction in mission time variance compared to the swarm method. Therefore, the proposed epistemic planning framework allows for performance close to the ideal always-connected systems while enabling each robot to explore the environment more efficiently. We also observed similar outcomes for heterogeneous teams, where agents can collaborate and leverage different capabilities to explore and complete tasks effectively.

Building on this chapter, we extend this work by assuming that tasks may be known apriori. This allows us to utilize a centralized planner to coordinate interactions and employ epistemic planning to revise the plan if the beliefs of the robots in the system need to be adjusted at runtime. The next chapter lays out our framework for this idea.

Part III

Online Epistemic Planning and Inference for Multi-Robot Systems

Chapter 5

Online Epistemic Replanning for Multi-Robot Systems

Chapter 5 augments previous research on epistemic planning by addressing the multiple traveling salesman problem (mTSP) under conditions of uncertainty and limited connectivity. In the context of an mTSP, intermittent communication failures among robots during operations can lead to undetected issues within the MRS, which may endanger complex or urgent missions. To tackle this issue, our proposed framework includes a centralized planner that assigns mission tasks by encouraging periodic rendezvous between robots to reduce the impact of unforeseen events during mission execution, and a decentralized replanning method using epistemic planning to formalize belief propagation and a Monte Carlo tree search for policy optimization based on distributed rational belief updates. The proposed framework outperforms a baseline heuristic and is validated through simulations and experiments with aerial vehicles. This chapter is based on the following publication:

- L. Bramblett, B. Miloradovic, P. Sherman, A.V. Papadopoulos, and N. Bezzo, "Robust Online Epistemic Replanning of Multi-Robot Missions," *2024 IEEE International Conference on Intelligent Robots and Systems (IROS).*

5.1 Introduction

In this chapter, we focus on the following question: *How can we ensure cooperative and efficient behavior for task allocation when a centralized predefined plan must change at runtime?* This question is an expansion of Chapter 4 and the work published in [10, 9], allowing for the elimination of certain limiting suppositions and making the work more suitable for practical scenarios where tasks are known but unforeseen changes in the environment or MRS may occur. Our proposed solution has two main components: 1) a centralized mission planner that accounts for intermittent rendezvous, promoting the discovery of failures and inefficiencies in the MRS if something does not go according to plan, and 2) an efficient runtime plan adaptation that leverages our recent epistemic

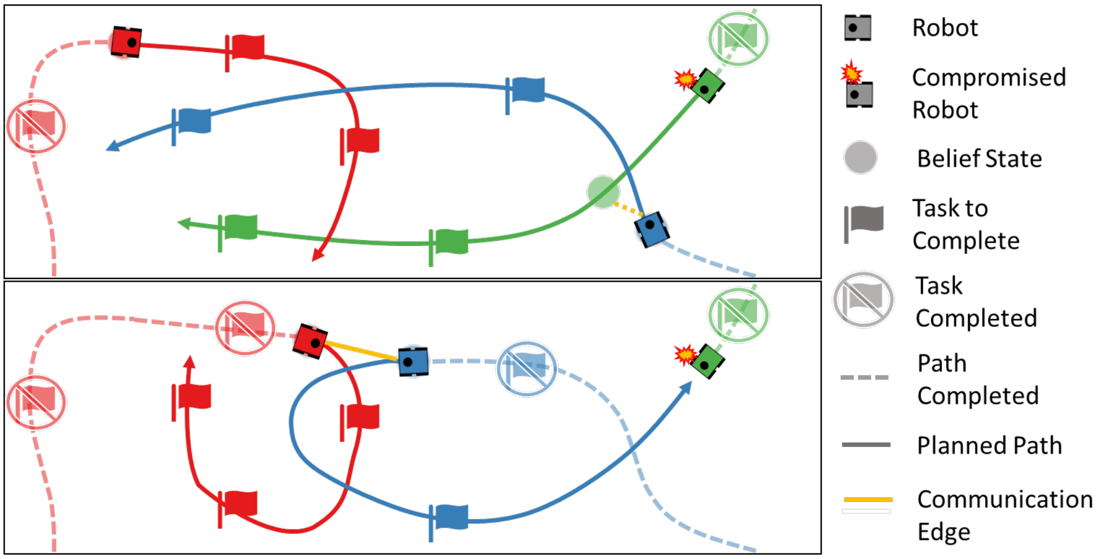


Figure 5.1: Pictorial representation of the problem presented in this chapter. The green robot fails, and the blue robot observes that its belief is false. The blue robot routes to share this information with the red robot, reallocating tasks in the environment before searching for the green robot.

planning research to reason about the probable knowledge and intentions of others based on the current epistemic state and dynamically reassign tasks. Our proposed framework enables MRS to cooperate, given limited communication and an uncertain operating environment.

Consider the example in Fig. 5.1, where three robots complete tasks based on an initial centralized plan. During disconnection, each robot maintains a set of possible *belief* states for other robots and a set of *empathy* states that other robots might believe about it. In the top frame, the blue robot realizes that its belief of the green robot is false. It then communicates this to the red robot, and consequently, the red and blue robots reallocate their tasks (bottom frame). The blue robot is assigned to locate the green robot to ensure all tasks are completed. In this manner, robots can successively reason based on their local observations.

The contributions of this work are two-fold: i) a genetic algorithm for multi-robot mission planning in a centralized manner, accounting for intermittent rendezvous at user-defined priorities, and ii) an epistemic planning framework for local replanning, utilizing a Monte Carlo tree search to maximize policy reward based on knowledge and beliefs about the system and environment. To the best of our knowledge, this is a novel approach that combines epistemic logic with runtime task allocation adaptations given intermittent communication. We show that our method outperforms a baseline heuristic in which robots complete their assigned tasks before backtracking to find faulty robots.

5.2 Epistemic Replanning Preliminaries

We modify our previous epistemic planning formulae in Sec. 3.2 to account for the centralized knowledge of the mission. In this chapter, we assume that the tasks are known, but that uncertainties or failures may occur during operation. The distributed knowledge and reasoning for robots in the system are modeled using epistemic logic [8]. An epistemic state for AP is represented by the tuple $s = (W, (R_i)_{i\in\mathcal{A}}, L, W_d)$ where W is a finite set of possible worlds, $R_i \subset W \times W$ is an accessibility relation for robot i simplified to R for reference to all robots, $L : W \to AP$ assigns a labeling to each world defined by its true propositions, and $W_d \subseteq W$ is the set of designated worlds from which all worlds in W are reachable. The initial epistemic state is denoted as $s_0 = (W, R, V, \{w_0\})$. If $W_d = \{w_0\}$, s_0 is the global epistemic state. The world, w, signifies a set of true propositions that, in our application, is the disposition of each robot. The worlds that exist for the system are defined by the combinations of all possible dispositions of each robot in the MRS (e.g., task assignment, velocity). The truth of $\mathcal{L}$-formulas in epistemic states is defined with standard semantics similar to [8]:

$$
\begin{aligned}
(W, R, L, W_d) &\models \phi \text{ iff } \forall w \in W_d, (W, R, L, w) \models \phi \\
(W, R, L, w) &\models \phi \text{ iff } \phi \in L(w) \text{ where } \phi \in AP \\
(W, R, L, w) &\models K_i\phi \text{ iff } \forall v \in W, \text{ if } (w, v) \in R_i \\
&\qquad \text{then } (W, R, L, v) \models \phi \\
(W, R, L, w) &\models C\phi \text{ iff } \forall v \in W, \text{ if } (w, v) \in \cup_{i\in\mathcal{A}} \\
&\qquad \text{then } (W, R, L, v) \models \phi
\end{aligned}
$$

The accessibility relation R_i represents the uncertainty of robot i at run-time for a global epistemic state $s = (W, R, L, w_d)$. In this state, the robot i cannot distinguish between the actual world w_d and any other world v where $(w_d, v) \in R_i$. Consequently, robot i's knowledge is based on what is true in all of these worlds v. Sequences of relations are used to represent higher-order knowledge. For example, the statement "robot i knows that robot j knows ϕ" is true in s if and only if $s \models K_iK_j\phi$. This condition is satisfied when ϕ is true in all worlds accessible from w_d through the composite relation of R_i and R_j. The perspective of robot i is defined as $s_i = (W, R, L, \{v \mid (w, v) \in R_i; w \in W_d\}$. If s is the global state, then s_i is the perspective of robot i on s. In this work, we represent a subset of these perspectives as particles moving through the environment in the set

$$\mathcal{P}_i = \{p_{ij,b}\ \forall j \in \mathcal{A}, \forall b \in \mathcal{B}\}. \tag{5.1}$$

where beliefs $b \in \mathcal{B}$ are a finite set of particles for each robot i that represent how a robot j would behave given a different, but accessible, world $w \in W_d$.

Dynamic epistemic logic is expanded from epistemic logic through action models [8]. These models affect a robot's perception of an event and influence its set of reachable worlds, R_i. A robot may plan to reduce the run-time uncertainty by taking actions. We simplify the notation of the action model by referring to actions in plain language. The action library, A, is the set of actions that a robot can enact during mission execution. We express the epistemic product model as $s \otimes i : a = (W', L'_i, V', W'_d)$ where $s \otimes i : a$ represents the new epistemic state after the action $a \in A$ has been enacted by robot i. A planning task is represented by the tuple $\Pi = (s, A, \gamma)$. An execution policy π is a sequence of actions in A for robots in the MRS that will satisfy the common mission objective γ given an epistemic state s.

In this work, we assume that all robots know the location of all tasks $\mathcal{V}$ present in the environment and the initial location of all robots in the system. Given a limited communication range r_c, this approach aims to minimize the total time to complete all tasks in the environment since robots can experience failures or disturbances during operation. We formally define our problems as:

Problem 5.1 ***Centralized mTSP Planner with Intermittent Communication:*** *Design a strategy for an MRS to complete all tasks while weighing efficient rendezvous points. The goal is to minimize the mission's duration, considering that faults and disturbances may occur during execution, necessitating a communication and replanning mechanism.*

Problem 5.2 ***Robust Online Replanning:*** *Formulate a policy for robust online replanning of the team's operations when faults or disturbances decrease the original plan's efficiency. The policy should minimize the time to complete all tasks, considering any necessary communications with disconnected robots and the deprecated state of the system.*

The mathematical formulation of Problem 5.1 matches the one of mTSP [62] where we seek to minimize the longest tour of any robot represented by $\mathcal{Q}$. However, in our work, robots can have different starting and ending depots. We also allow robots to have no tasks assigned to them.

The optimization problem is expressed in its epigraph representation, where the objective function is included in the constraints as

$$\begin{aligned} \min \quad & \mathcal{Q} \\ \text{s.t.} \quad & \sum_{i \in \mathcal{V}^{\Sigma}} \sum_{j \in \mathcal{V}^{\Delta}} \omega_{ijs} \cdot x_{ijs} \leq \mathcal{Q}, \qquad \forall s \in \mathcal{A}, \end{aligned} \tag{5.2}$$

where ω_{ijs} is the cost of traveling from task i to task j for robot $s \in \mathcal{A}$. The binary decision variable x_{ijs} defines if the robot s travels from task i to task j. The sets $\mathcal{V}^{\Sigma}$ and $\mathcal{V}^{\Delta}$ represent the inclusion of all the tasks and all the starting depots, and all the tasks and the ending depot, respectively.

The goal of the optimization process is to minimize the variable $\mathcal{Q}$, also known as "minMax" optimization, where we minimize all robots' maximum tour or makespan.

5.3 Approach

Our proposed framework is designed for an open mTSP in which robots are not required to return to their starting location; instead, each robot has a starting and ending depot. When solving the mTSP, we promote intermittent communication by rewarding robot interactions during their respective tours, allowing robots to share information or realize that the original plan has changed. To realize changes, each robot propagates belief and empathy states to allow robots to observe the local environment, reason about system operations while disconnected, and adjust local plans when necessary. For ease of discussion, let us consider two robots, i and j. From the perspective of the robot i, a *belief state*, $p_{ij,b} \in \mathcal{P}_i$, represents a possible state of a robot j and an *empathy state*, $p_{ii,b} \in \mathcal{P}_i$, is robot i's belief about itself from the perspective of other robots. With this knowledge, robot i predicts and tracks empathy states to decrease the number of locations in which robot j may search for robot i and allows the system to complete all tasks more efficiently, given new operational constraints. The diagram in Fig. 5.2 summarizes this architecture, where the centralized planner first routes robots to tasks in the environment, assessing the solution's fitness by minimizing the maximum tour length of a robot and rewarding intermittent communication based on a user's preferred settings. If robots disconnect, the set of belief and empathy particles, $\mathcal{P}_i$ propagates according to the sequence of actions, π_0 set by the centralized planner. If the robot

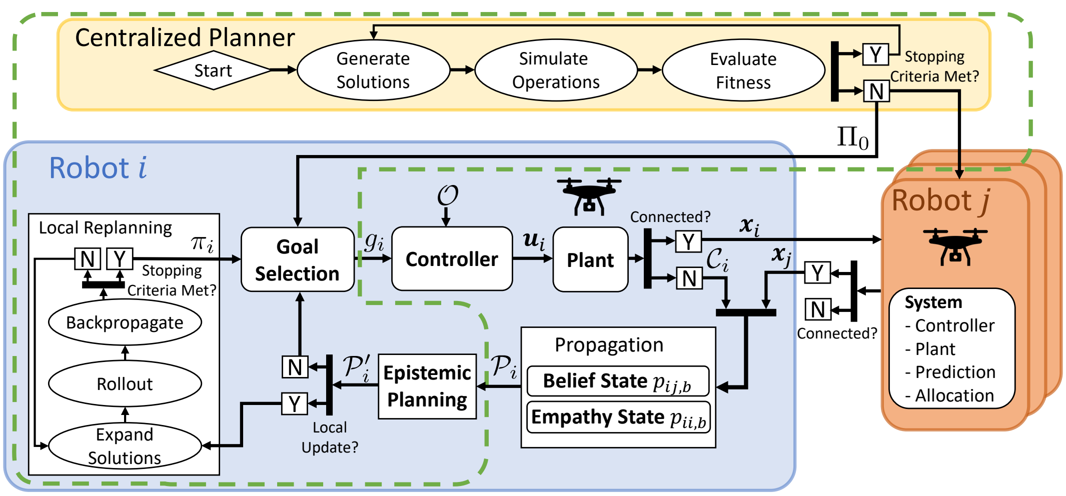


Figure 5.2: Diagram of the proposed approach. The contributions of this chapter are within the green box.

locally observes system changes, epistemic planning allows each robot to determine the best series of actions to estimate the positions of lost robots, share any necessary information with other robots, and navigate to the remaining cities.

5.3.1 The Centralized Planning Algorithm

The centralized planner used in this chapter is based on a GA adapted to solve combinatorial optimization problems, specifically mTSP. Chromosomes are encoded as a set of arrays, where each array encodes a robot's plan. A graphical representation of a single chromosome is given in Fig. 5.3. The size of each array is equal to the sum of the number of robots and tasks, that is, $n + m$. The elements of the array are integer task IDs. Following the task chain, the robot's route can be extracted from the encoding. For example, in Fig. 5.3, if we look at Robot 1, we can see that the first task in its plan is 5, and the next task ID is then stored in column 5, which is Task 7. This continues until a destination depot with the ID of $n + m$, 10 in this example, is reached.

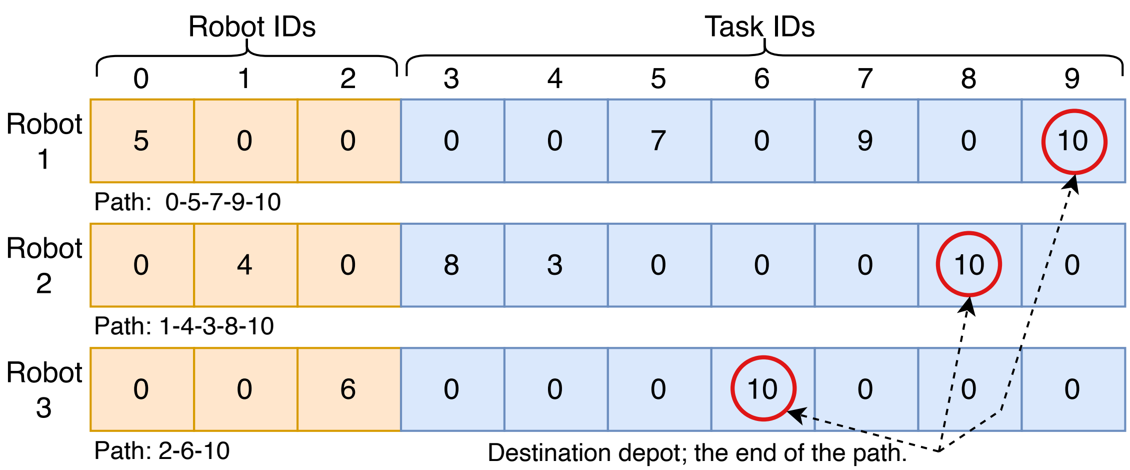


Figure 5.3: Graphical representation of chromosome encoding.

The initial population is generated randomly to start with a high diversity and seeded in the feasible region of the search space.

The crossover operator is a modified version of Edge Recombination Crossover (ERX) [86]. The first step is to select two parents for crossover from the mating pool. The mating pool is generated, accounting for the crossover probability and each individual's fitness. Next, an adjacency matrix that contains the makeup of neighboring tasks based on the two chosen parents is constructed. We then randomly select a starting task and the selection chain continues by randomly selecting a task from a neighboring list of previously allocated tasks. We randomly select a new task if all neighboring tasks are already allocated. We apply a jump mutation that randomly changes the placement of a single task in the route, and the swap mutation selects two tasks and swaps their locations. Jump and swap mutations are invoked twice: the first for intra-robot mutations and the second for inter-robot mutations, such that there are both local and global mutation operations.

Greedy Search (GS) and 2-opt [42] are two local refinement methods implemented to reorder cities within a robot's plan resulting from the GA allocation of cities to salesmen. Local refinement methods exploit the candidate solution by reordering the list of tasks governed by the nearest-neighbor or 2-opt. In the next sections, we explain our modifications to increase the robustness of the MRS.

5.3.2 Interaction Reward Mechanism

The general rule for creating a good plan for TSP or mTSP is to have routes that do not cross. The most successful heuristic for solving these problems directly exploits this rule, e.g., 2-opt or Lin–Kernighan heuristics [50], but, in this chapter, we take a different approach by allowing the planner to create *interactions* between robots. Within this framework, we define an *interaction* as an event where robots are within the range r_c to exchange information. Rewarding robots who travel within r_c can create crossings in the resulting routes, contrary to [50]. However, we argue that this can benefit the overall execution time of the mission when the system does not operate as planned. This will enable robots to detect system failures faster during execution without laborious backtracking after reaching the depot.

To maximize the number of interactions among robots, we introduce a mechanism to reward the exchange of information between robots. However, maximizing the number of interactions alone is not sufficient, as each interaction's value must be taken into account. For example, exchanging information at the beginning or close to the end of a mission may not be beneficial, as little new information can be gained from these interactions. Furthermore, redundant interactions over small-time intervals should not be highly rewarded since no new information is likely to be shared. To capture this, we introduce, for every robot i, a time interval $[\tau_i^s, \tau_i^e]$ when the robot can be rewarded for interacting with other robots. The potential reward is designed to grow linearly from $t = \tau_i^s$ for $\Phi_i = (\tau_i^e - \tau_i^s)/2$ time units and to stay constant for the remaining part of the interval. Then, the potential reward function $U_i(x)$ for robot i is defined as:

$$U_i(x) = \begin{cases} x, & \text{if } 0 < x \leq \Phi_i \\ \Phi_i, & \text{if } \Phi_i < x \leq 2\Phi_i, \\ 0, & \text{otherwise.} \end{cases} \tag{5.3}$$

However, the actual reward $\mathcal{R}_i(t)$ is assigned only if the interaction happens, according to

$$\mathcal{R}_i(t) = \begin{cases} \rho U_i(t - t_i^{lr}), & \text{if robot } i \text{ interacts at time } t \in [\tau_i^s, \tau_i^e], \\ 0, & \text{otherwise,} \end{cases} \tag{5.4}$$

where t is the current time, and t_i^{lr} is the time when the last reward was assigned to robot i. Furthermore, ρ is half of the average distance between tasks, and it is introduced as a weight related to the structure of the problem. This means that ρ scales with the problem instance. The total reward is then calculated as:

$$\mathcal{R}_{tot} = \sum_{t=0}^{\mathcal{Q}} \sum_{i \in \mathcal{A}} \mathcal{R}_i(t). \tag{5.5}$$

However, two robots may exchange information frequently while a third robot is in no contact with them. To overcome this issue, we also introduce a penalty mechanism for robots not interacting with other robots. This mechanism requires a tunable threshold, σ, to be defined, e.g., all robots have to interact with another robot at least once before completing 50% of a given mission, i.e., $\sigma = 0.5$. The penalty for failing to do so is calculated as follows:

$$P_i = \begin{cases} (t_i^{int} - \sigma \cdot t_i^{max}) \cdot \rho & \text{if } t_i^{int} > \sigma \cdot t_i^{max}, \\ 0, & \text{otherwise,} \end{cases} \tag{5.6}$$

where t_i^{max} is the time required for robot i to complete its plan, and t_i^{int} is the time when robot i had its first interaction with another robot. To get the total penalty, P_{tot}, we sum up the penalties over all robots. The optimization problem (5.2) can now be updated with rewards and penalties as follows:

$$\min \;\; \mathcal{Q} - \mathcal{R}_{tot} + P_{tot}. \tag{5.7}$$

We also extend the aforementioned approach if a user requires more control over the mission makespan compared to a traditional mTSP solution. In this case, we solve a bilevel optimization problem where we first minimize $\mathcal{Q}$ subject to (5.2) and then optimize the following:

$$\begin{aligned} \max \;\; & \mathcal{R}_{tot} - P_{tot} - \Delta\mathcal{Q} \\ \text{s.t} \;\; & (5.2);\; \Delta\mathcal{Q} \leq \delta\mathcal{Q}^* \end{aligned} \tag{5.8}$$

where $\Delta\mathcal{Q}$ represents the difference between the solution to the upper-level optimization problem (5.2) represented by $\mathcal{Q}^*$ and the inner optimization task in (5.8). The user-defined variable $\delta \in [0, 1]$

represents the extent to which the typical mTSP makespan minimization can be worsened to increase interactions using (5.5) and (5.6). In this way, we have better control over the quality of the produced solution, with the mission duration upper-bounded by the user-defined limit.

5.3.3 Belief & Empathy Propagation

So far, we have explained how we developed a centralized strategy that allows intermittent interactions. Now, we transition to online adaptations, indicated by the blue section in Figure 5.2, to plan based on the information gained from these interactions. In our framework, each robot propagates belief states for all robots in the MRS. This allows a robot i to plan according to its beliefs about other robots and to empathize with what other robots expect robot i to do while disconnected. Each robot predicts the future states of a set of beliefs for all robots in the system and will follow the closest empathy state even if a malfunction occurs, allowing a robot only to propagate a finite number of beliefs represented by the set $\mathcal{P}_i$ in (5.1). A robot i defines its empathy particles as $\mathcal{P}_i^e = \{p_{ii,b} \ \forall b \in \mathcal{B}\}$ and its belief particles about other robots as $\mathcal{P}_i^r = \{p_{ij,b} \ \forall j \in \mathcal{A}, \forall b \in \mathcal{B}\}$. If disconnected, a robot i propagates beliefs according to the last globally communicated epistemic state between robot i and robot j, moving particles based on how robots would behave given a subset of true propositions from the set AP introduced in Sec. 3.2. Initially, we note that all robots know the initial position and disposition of all robots, defined by the centralized plan in Sec. 5.3.1. Particles are propagated along the tour provided by the centralized planning algorithm. Given that all robots follow an empathy particle during exploration, we next present our strategy to update the epistemic state if changes occur at runtime.

5.3.4 Epistemic Replanning

Robots follow the centralized plan initially, but if operations do not go as planned and a robot experiences a failure or other robots communicate changes to the system, a rational belief update must occur. We formulate a belief update for this application as: (i) A robot updates its own belief given it cannot operate as expected; (ii) a robot updates its belief about another robot, given it is not traveling according to a previously expected belief; (iii) a robot communicates a belief update about a disconnected robot to a subset of connected robots within the communication range. In all these scenarios, a robot can update its execution policy of tasks in the environment, given its new belief about the MRS to complete all tasks in the environment, which is equivalent to satisfying the common goal γ from Sec. 3.2. However, belief updates can have cascading effects across the MRS if new information is not communicated efficiently and on time. Therefore, we introduce a hierarchical framework to update allocations at runtime and when failures or disturbances occur and robots can no longer follow the original plan.

Epistemic Updates

Establishing a mechanism for logical updates is important to determine when or if a robot should find or communicate with other members of its team and how to reach a consensus on disconnected team members within a locally connected team. A dynamic epistemic logic (DEL) framework allows a robot i to succinctly share beliefs and update a robot's perspective s_i on the epistemic state s. There are two cases where updates can occur: i) when connected to all robots and ii) when expecting to connect with another robot. From the established semantics in Sec. 3.2, we know $A = \{percieve(\phi), announce(\phi), complete(\phi)\}$ The action *complete* represents a robot completing a task, *perceive* symbolizes a robot observing a generic proposition ϕ about the MRS, and *announce* constitutes communication with a locally connected team. The set $\Psi = \{track\}$ is functionally interpreted for $B_i\, track(p_{ij,b})$ as the robot i knows that the robot j is tracking the belief particle b.

First, we address a belief update when robots are within communication range. We assume that because robots are cooperative, all belief updates are accepted and are only outdated if an event occurs, such as system failures or disturbances. These updates are announced such that the epistemic state from robot i's perspective is:

$$s_i \otimes announce(\Omega) = s_i' \models K_i V(\Omega^i) \bigwedge_{j \in \mathcal{A}} K_i K_j V(\Omega^j)\ \forall i \in \mathcal{C}. \tag{5.9}$$

where $announce(\Omega)$ is an action symbolizing the announcement of all robots' dispositions, Ω. The notation models robot i's knowledge of the dispositions of all robots, and the function $V(\Omega^i)$ maps the dispositions of the robot i to atomic propositions in the set AP. The set $\mathcal{C} \subseteq \mathcal{A}$ represents the set of robots within robot i's communication range. The belief particles are updated from the announcement of all states to the MRS such that

$$p_{ij,b} \leftarrow \Omega^j,\ \forall (i,j) \in \mathcal{A}^2,\ \forall b \in \mathcal{B}. \tag{5.10}$$

Since beliefs are shared according to (5.9), the particles in this set are propagated according to the dispositions of each robot. For example, in a three-robot team, if robot 1 communicates with robots 2 and 3 that it will execute robot 3's tasks, all robots would propagate a belief particle that moves robot 1 according to its assigned tasks.

The *perceive* action causes a robot to change his belief in the epistemic world. If robot i perceives that robot j is not at its believed location, it updates its epistemic state with the epistemic action *percieve*:

$$s_i \otimes i : perceive(\neg track(p_{ij,b})) \models V(\Omega^j, B_i \neg track(p_{ij,b})) \tag{5.11}$$

where the function V takes two arguments, mapping robot i's updated belief about robot j to an atomic proposition in AP and robot i's new epistemic state is evaluated as the epistemic product after *perceive* has been enacted. The particle propagation does not change in this case since robots may seek out other robots without knowledge of this belief update. In the event of a malfunction or fault of a robot, the robot updates its belief in the same way with $j = i$ and tracks respective empathy particle $p_{ii,b+1}$.

In this way, the knowledge of disconnected robots is not affected, nor does the robot i update its belief that a disconnected robot would know the updated information. With our epistemic states and actions defined, we now describe how these concepts can be used for planning. As stated in Sec. 3.2, a planning task for the MRS is defined by the tuple $\Pi = (s, A, \gamma)$ where γ is a goal formula. In plain language, the goal formula is to complete all tasks. We define the epistemic update associated with a robot who completes an assigned task $\nu \in \mathcal{V}$ as:

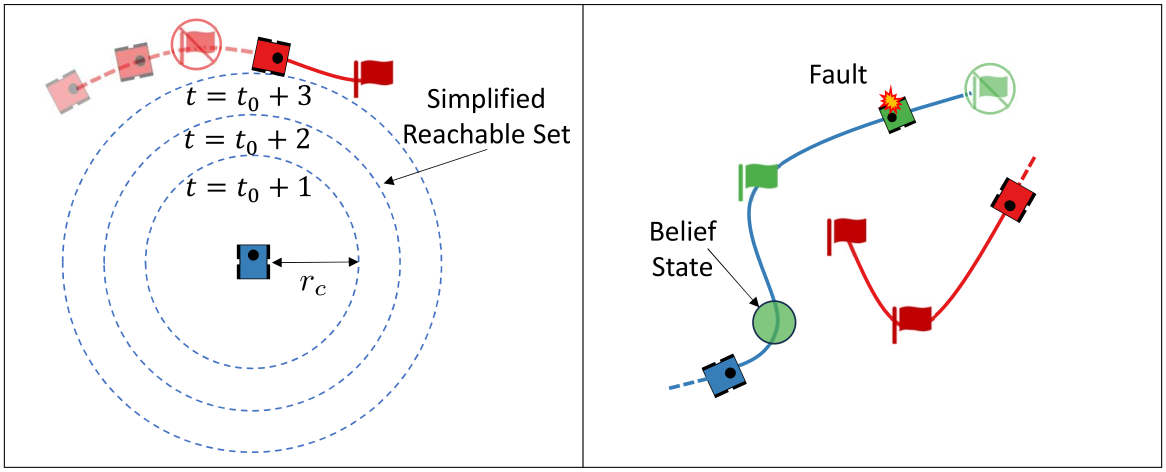
$$s_i \otimes i : \mathit{complete}(\nu) \models V(\Omega^i, B_j \mathit{complete}(\nu)) \; \forall j \in \mathcal{A}. \tag{5.12}$$

The variable Ω^i is also updated to represent the new disposition of robot i. Given that other robots are also tracking the believed location of robot i, belief updates occur without communication, although these beliefs may be incorrect if a failure has occurred. Thus, an augmented policy that allows the MRS to achieve the common goal must be enacted.

In the event of a malfunction, we introduce two new types of tasks that allow operational robots to gather the necessary information about the system's condition and complete any unfinished tasks. These types of tasks are called *gossiping* and *finding*. Given robot i's belief and the planned interactions with other robots according to the mTSP solution (5.7), a robot should communicate before any planned interaction. Ensuring the completion of communication tasks (gossiping) and promptly identifying malfunctioning robots are vital steps to facilitate accurate information exchange and prevent the spread of misinformation within the system. The estimation of the interaction point can be determined using the time-based trajectory of the belief state and the reachable set of the robot involved, as depicted in Fig. 5.4(a). The position of robot j's belief state at a specific time t is denoted as $p_{ij,b}(t)$. To find the point where robot i's communication range intersects with the communication range of robot j's predicted location, we find the minimum timestep t_r that satisfies the equation:

$$\|\boldsymbol{x}_i(t_r) - \boldsymbol{p}_{ij,b}(t_r)\| - R(t_r) > r_c \tag{5.13}$$

where $\|\boldsymbol{x}_i(t_r) - \boldsymbol{p}_{ij,b}(t_r)\|$ represents the distance between the location components of the robot's position, $\boldsymbol{x}_i$ and robot i's belief about robot j, $\boldsymbol{p}_{ij,b}$. The reachable set, R, for robot i expands at every timestep based on the robot's velocity. If all belief states have been checked for a deprecated robot j, a robot i backtracks along the previously established path until robot j is located. An example is shown in Fig. 5.4(b), where a blue robot routes backward along the green robot's path to communicate and reallocate any remaining tasks.



(a) Planning for a dynamic task (b) Finding at an unknown location

Figure 5.4: Examples of tasks generated as a result of belief updates

Balanced Workload Partitioning

Given the limited nature of communication in this application, robots first assign new tasks to connected robots before optimizing their path [58] so that robots do not need to maintain a connection while optimizing routes to new tasks. Robots instead partition tasks based on a balanced workload and accounting for any belief updates. For example, suppose two robots, i and j, are connected, and robot k is not within the communication range. In that case, robot i might believe that robot k is functioning according to the initial state, s_0, but robot j did not *perceive* robot k at its respective belief state $p_{ik,b}$. So robot j *announces* its belief to robot i. Robots i and j then bid on the new task, which is to find robot k. We let $\mathcal{V}_c$ be the set of tasks the connected robots must complete, and the cost function for allocating a task to a robot is user-defined (e.g., distance, time). Algorithm 5 presents the bidding mechanism used in this chapter, noting that this is only instigated if a belief update about the MRS functionality has occurred (i.e., a fault or disturbance). Once the task allocations have been determined, the next step is to find the optimal tour.

Monte Carlo Tree Search

In this section, we combine Monte Carlo Tree Search (MCTS) with a DEL to implicitly coordinate plans when the system does not operate as originally intended. Referring to Sec. 5.3.3, to limit the policy search space, each robot's state consists of believing that a robot j is following one of the particles represented in robot i's set of particles $\mathcal{P}_i$. MCTS is applied to complex games such as chess or Go to find the next best move, even in real-time [46]. To model the solution space

Algorithm 5 Balanced Workload Partitioning

$tour_s \equiv \emptyset \ \ \forall s \in \mathcal{C}$
for each $\nu \in \mathcal{V}_c$ **do**
  for each $s \in \mathcal{C}$ **do**
    $bid_s = cost(tour_s \cup \nu)$
  end for
  $winner = \arg\min_{s \in \mathcal{C}}(bid_s)$
  $tour_{winner} \leftarrow tour_{winner} \oplus \nu$
end for

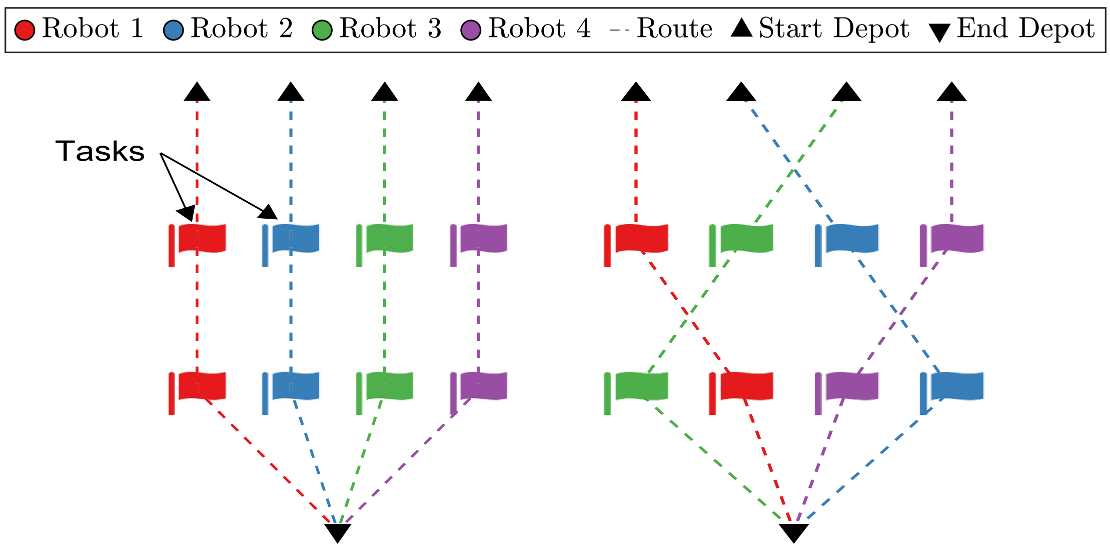
effectively, we use the current epistemic state from a robot i's perspective s_i. The MCTS algorithm simulates changes to the epistemic state when an action is taken and is represented as $s'_i \sim s_i \otimes a$. Robots add *gossiping* or *finding* tasks based on local observations, as discussed in Sec. 5.3.4 when 1) a robot experiences a fault or disturbance, or 2) a robot i observes that its epistemic belief about the state of robot j is incorrect.

The search tree is generated by repeating the four steps – *selection*, *expansion*, *simulation*, and *backpropagation* – until a certain termination condition is met, which in this approach is a certain number of simulations. In the selection stage, a leaf node that has not yet been fully expanded is selected. We employ the upper confidence bound applied to trees (UCT) technique, which is typical in MCTS, to decide which vertex to simulate from the root node. Specifically, UTC was chosen because it has been shown to strike a good balance between exploration and exploitation [46]. Expansion occurs by randomly applying a random action or, in this case, adding a random task to a robot's route. The simulation then performs a random remaining route until termination (i.e., all of the robots' allocated tasks are performed) and then backpropagates the reward, applying the estimated value to the expanded node in the expansion step. The MCTS seeks to maximize the negative time it takes for a robot to complete all its assigned tasks, estimating the time to *find* and *gossip* with robots using the methods in Fig. 5.4. We summarize the MCTS simulation applied in this approach in Algorithm 6. The tour with the lowest estimated cost is the chosen execution policy, π for all the robots in the system that will satisfy γ.

To aid the reader in understanding, the proposed approach is implemented on the toy example shown in Fig. 5.5. We show the trivial solution to the mTSP in Fig. 5.5(a) and the modified mTSP solution considering the interaction reward from (5.5) in Fig. 5.5(b). In Fig. 5.6, we show a subset of frames from the approach in which the blue robot realizes that the purple robot is deprecated in Fig. 5.6(a), gossips the information to the red and green robots who reallocate remaining tasks while the blue robot is charged with finding the purple robot in Fig. 5.6(b). In Fig. 5.6(c), the blue robot has communicated with the purple robot and routes to the remaining task in the environment before returning to the home base.

Algorithm 6 MCTS - Simulate

$tour$ = child.tour($child\_rollout$)
$cost = 0$
for each $c \in tour$ **do**
  if $type(c)$ = "Static Robot" **then**
    $path$ =reverse($particle(c).tasks$)
  else if $type(c)$ = "Dynamic Robot" **then**
    $path$ = find_intersect_point($particle(c).tasks$)
  else
    $path$ = task_location(c)
  end if
  $cost$ += time_to_traverse($path$)
end for
$reward = -cost$



(a) Ideal: 61m makespan

(b) Ours: 69m makespan

Figure 5.5: Ideal mTSP allocation for 4 robots is shown in (a) and (b) is the solution with our proposed method.

5.4 Simulations

This section showcases the outcomes obtained through MATLAB simulations of our method executed by teams of two to five robots. The square environment used for the simulations has dimensions of $30 \times 30, 30 \times 30, 90 \times 90$, and 150×150 [m], and for each scenario, a total of 10, 10, 30, and 50 tasks were generated. The locations of the tasks were randomly generated for each scenario.

In our proposed approach, each robot propagates three particles. The initial maximum speed of each vehicle is 5 [m/s], and the second and third particles travel at a linear speed that is 80% and 60% of the vehicle's maximum speed, respectively. The maximum communication range is 5 [m] from the center of the robot. In our simulations, we implemented one fault for teams of two

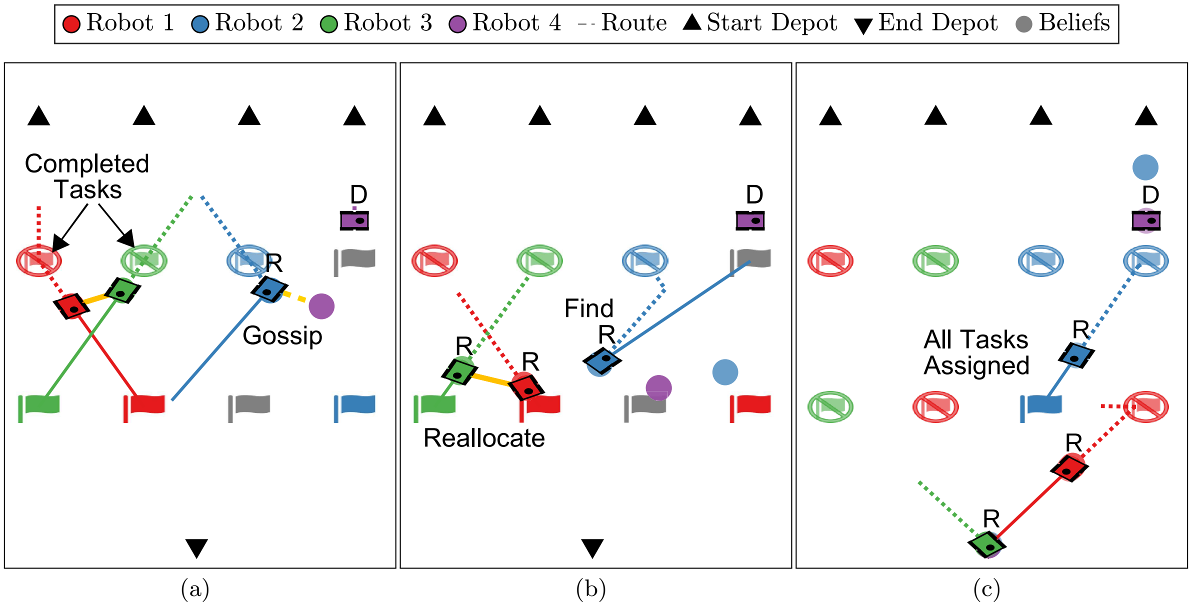
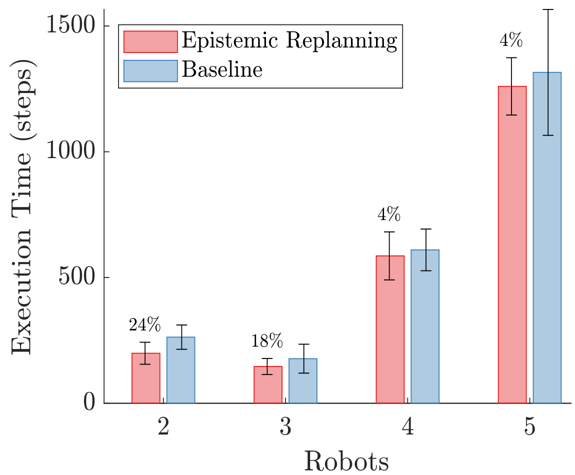


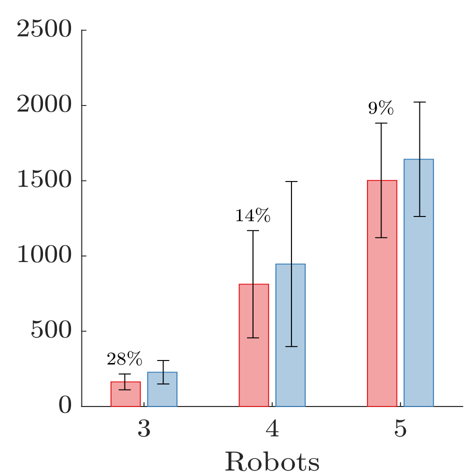
Figure 5.6: Our approach on a toy example, where the purple robot fails, and the blue robot realizes that the purple robot is not where expected.

to five robots and two for teams of three to five robots, randomly occurring to any robot, causing the affected robot to track its second or third empathy particle or fail (i.e., zero velocity). Our approach was compared to a baseline heuristic in which the routes are determined by minimizing their makespan from (5.2) and backtracking to find team members who do not arrive at the depot when expected. We let δ equal 30% to increase the number of interactions between robots from (5.8) such that the makespan of our solution can be up to 30% longer than the baseline heuristic solution. We compare with this baseline to determine whether increased interactions and epistemic replanning truly improved the outcome. As shown in Fig. 5.7, our approach outperforms the baseline heuristic by a significant margin in all scenarios, and we note that the margin increases as the number of failures increases between Fig. 5.7(a) and Fig. 5.7(b).

Note – We emphasize that the margin of improvement is smaller as the teams become larger because information sharing becomes more inefficient as interactions between all robots become sparse. This introduces an interesting expansion outside the scope of this work for introducing optimal sub-teaming to create more efficient information sharing for static or dynamic teams during operations. In addition, not all vehicles are equally likely to fail. As vehicles age, they may become less reliable, requiring more dependable vehicles to take over or pick up additional tasks if a vehicle's operating capacity is deprecated during operations [77].



(a) Scenarios with 1 failure



(b) Scenarios with 2 failures

Figure 5.7: Comparison of a baseline heuristic and the proposed approach.

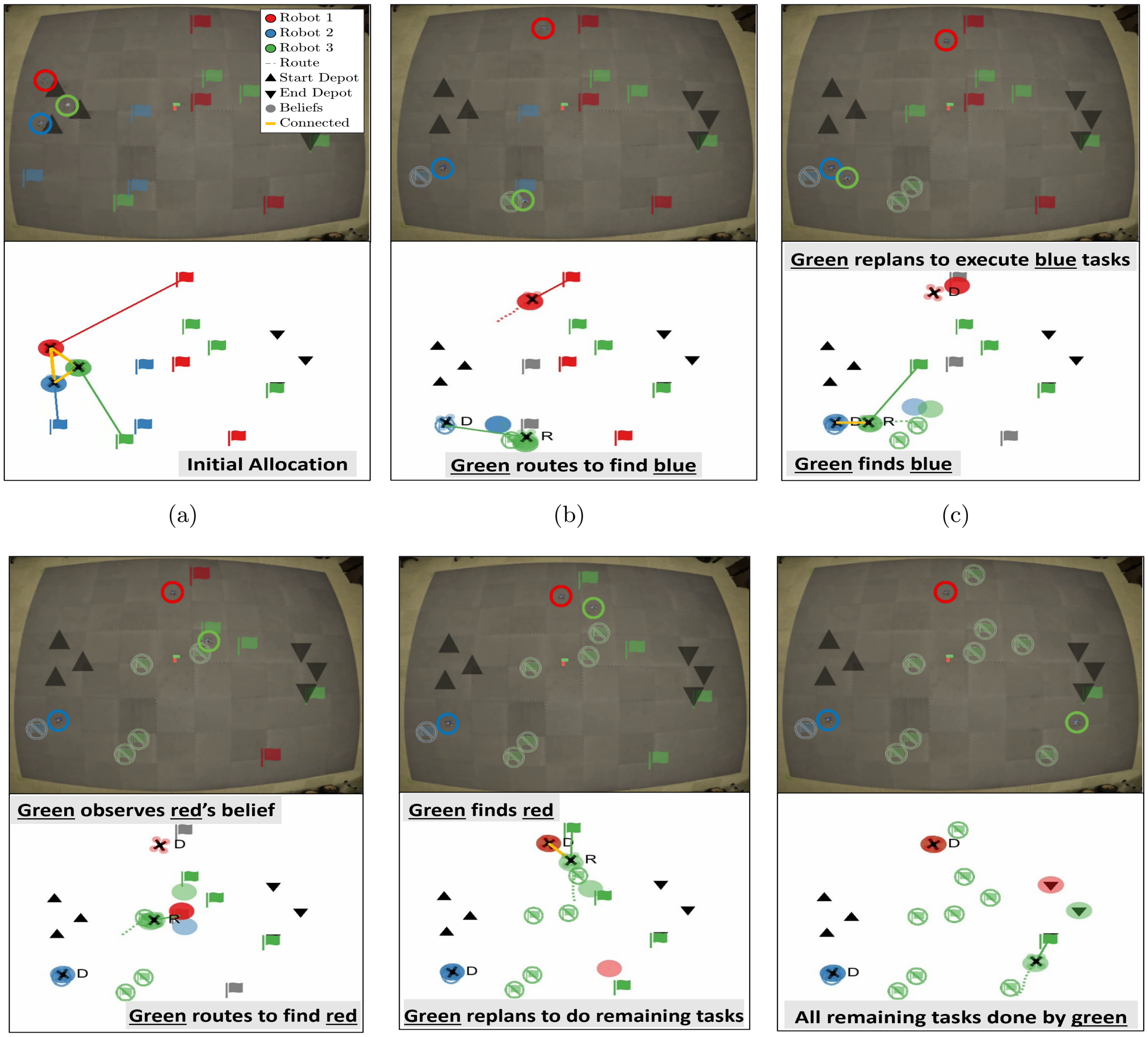
5.5 Experiments

Our approach was validated through several laboratory experiments with a multi-robot team. The team consists of several Bitcraze Crazyflies that used a Vicon motion capture system for localization. Vehicles start within the communication range to complete all tasks in the environment. The experiments were carried out in a 4×5.5 [m] space with a sensing and communication range of 0.5 [m] for each robot. The results of a sample experiment with ten tasks and three Crazyflies are shown in Fig. 5.8.

As shown in the figure, each robot is assigned a subset of tasks in Fig. 5.8(a). After disconnection, the blue robot fails and the green robot observes that the blue robot is not where expected in Fig. 5.8(b); the green robot backtracks along the blue robot's path and finds the blue robot in Fig. 5.8(c). In Fig. 5.8(d), the green robot also observes that the red robot is not where expected. The green robot backtracks along the path of the red robot and finds the red robot in Fig. 5.8(e). The green robot then replans all remaining tasks before ending at the depot in Fig. 5.8(f).

5.6 Discussion

This chapter presented a novel framework for multi-robot systems to use a modified centralized planning method to assign tasks, accounting for intermittent interactions. These interactions enable the system to use epistemic planning, which adapts to faults and disturbances by reassigning tasks based on a robot's reasoning about the system while disconnected. This method allows an MRS to disconnect and cooperatively plan based on a set of belief and empathy states if the system does not function as intended. The generalized task allocation algorithm uses these belief states



(a) (b) (c)

(d) (e) (f)

Figure 5.8: Snapshots and results of an experimental case study.

to assign tasks while considering the potential need to communicate with disconnected robots, facilitating dynamic task allocation without constant communication. We show the improvement of our framework compared to a baseline heuristic over several scenarios and apply our framework to real-world experiments.

In the previous chapters, we developed a complete solution for multi-robot exploration and task completion. We have shown how a multi-robot system can be more efficient by disconnecting and rendezvousing to share information when needed. In the next chapter, we explore the idea that if robots are able to observe each other and take actions to signal their intention, there may not need to be any communication infrastructure to accomplish mission goals.

Chapter 6

Active Inference with Epistemic Planning

In this chapter, we highlight our current work where a team of autonomous robots must complete tasks in the environment without communication. The robots are equipped with the ability to use higher-order reasoning and infer the goals of other agents while signaling their own intent to the other robots in the system. Some methods have addressed this problem by utilizing a theory of mind (ToM) framework, but typically only allow agents to use first-order reasoning about observations. In contrast, to deal with this problem, our proposed framework includes an efficient runtime plan adaptation using active inference to signal intentions and reason about a robot's own belief and the beliefs of others in the system, and a hierarchical epistemic planning framework to iteratively reason about the current MRS mission state. The proposed framework outperforms a baseline heuristic and is validated using simulations and virtual experiments with unmanned ground vehicles. This chapter is based on the following current work:

- L. Bramblett, J. Reasoner, and N. Bezzo, "Active Epistemic Inference for Task Allocation in Multi-Robot Systems," *in preparation for submission to the IEEE Transactions on Systems, Man, and Cybernetics: Systems.*

6.1 Introduction

Multi-robot systems (MRS) have the potential to transform current robotics applications, performing tasks more effectively and efficiently than a single robot. Central to MRS research is the idea that robots work together to accomplish common goals. The need for cooperative teaming is evident in numerous applications, such as search and rescue missions, firefighting, and underwater exploration. Effective collaboration requires robots to communicate and synchronize their actions, but difficulties often occur when communication is restricted, disrupted, or compromised. Especially for a heterogeneous MRS where different robots have different operating or sensing capabilities. Humans have an inherent ability to "see things from another's perspective" by understanding and sharing the beliefs of others without communicating. Imagine a parent attempting to convey a message to a child solely through their movements. The parent might understand the child's short attention span and limited observational skills, thus making exaggerated movements to communicate their intentions. In standard multi-robot missions, there are usually no strategies in place for communication breakdowns, or the strategies that do exist rely solely on each robot's first-order understanding of the environment and system.

Theory of mind (ToM) and epistemic planning [7], on the other hand, can enable these robots to reason about the likely knowledge and intentions of others based on their last known state and act accordingly. Our previous work incorporates epistemic planning to allow a MRS to continue to cooperate, given limited communication and an uncertain operating environment [10, 9]. In conjunction, active inference can be used to compute belief-action pairs. Active inference operates on the principle that all entities strive to minimize variational free energy. This results in straightforward update rules for actions, perceptions, policy choices, learning processes, and the representation of uncertainty, which can extend to multi-agent processes [53]. In this work, we focus on the following question: *How can we ensure cooperative and efficient behavior for multi-robot tasks when robots cannot explicitly communicate?* Our proposed solution has two main components: 1) heirarchical epistemic planning that leverages our recent research [11] which allows the robots to reason about the system goals, and 2) an efficient runtime plan adaptation that leverages active inference to signal to others their own knowledge and intentions based on the current epistemic state and probable goals to accomplish the mission.

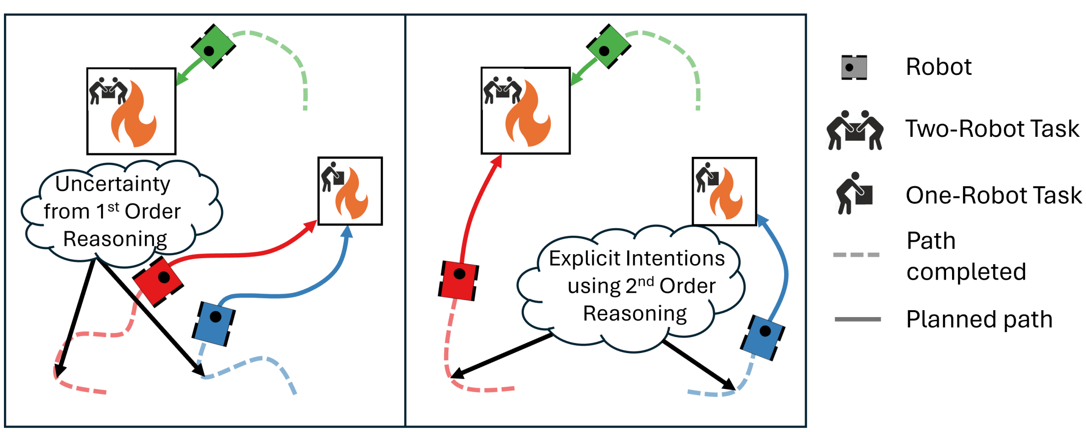


Figure 6.1: Pictorial representation of the problem presented in this chapter. The robots are unable to explicitly communicate their beliefs and must convey their intentions through sensorimotor communication. In the left frame, the red and blue robot are unable to converge to a correct belief state using only first-order reasoning. In the right frame, the red and blue robot clearly display their intentions by using higher-order reasoning about their observations.

Consider the example in Fig. 6.1, where three robots cannot communicate and the mission is defined by a two-robot task and one task requires only one robot. During operation, each robot maintains a belief about the system defined by the likelihood that any robot is moving toward one of the two tasks in the environment. In the left frame, each robot uses only its first-order observations, reasoning about each robot based only on its own observations. In the right frame, each robot uses higher-order reasoning to not only infer other robots' goals based on its own observations but also by empathizing with how other robots' beliefs would change based on its own actions and their subsequent observations. We note that by using only first-order reasoning, the red and blue robots cannot determine the goals of the other, and they converge to an incorrect belief. Using higher-order reasoning, the red and blue robots both account for the observation of the other in deciding their next actions and clearly indicate their intent and beliefs. In this way, the MRS is able to converge to a correct belief state using only local observations. In this work, we utilize up to the third level of reasoning as described in [81]: zero-order is a belief about oneself, first-order is a belief about others, second-order is a belief about what others believe about oneself, and third-order is a belief about what others believe about each other.

The contributions of this work are three-fold: i) a higher-order active inference framework for multi-robot task allocation without communication, ii) an epistemic planning framework for belief updates and iterative task allocation, and iii) a receding horizon controller and reinforcement learning pipeline that allows for real-time adaptations and policy decisions. To the best of our knowledge, this is the first work combining epistemic logic and active inference with runtime task allocation adaptations and no communication. We show that our higher-order reasoning method outperforms a baseline heuristic and first-order active inference to allocate tasks in an environment without communication.

6.2 Problem Formulation

Consider an MRS of n robots in the set $\mathcal{R}$. We let $\boldsymbol{x}_i$ denote the state variable of the robot i that evolves according to general dynamics at time t such that:

$$\dot{\boldsymbol{x}}_i(t) = \boldsymbol{f}_i(\boldsymbol{x}_i(t), \boldsymbol{u}_i(t), \boldsymbol{\nu}_i)$$

where $\boldsymbol{u}_i(t) \in \mathbb{R}^{d_u}$ and the variable $\boldsymbol{\nu}_i \in \mathbb{R}^{d_\nu}$ denote the control input and zero-mean Gaussian process uncertainty. The function $\boldsymbol{f}_i$ represents the stochastic dynamics of robot i given a control input and process uncertainty. We also assume that all robots are equipped with sensors (e.g., camera, lidar) that allow robots to ascertain certain measurements from other robots in the system. A robot's continuous observation $\boldsymbol{o}_i(t)$ at time t depends on its own position $\boldsymbol{x}_i(t)$ and sensor configuration ω_i such that:

$$\boldsymbol{o}_i(t) = \boldsymbol{h}_i(\boldsymbol{x}_i(t), \omega_i) + \text{noise}. \tag{6.1}$$

where the function $\boldsymbol{h}_i$ is a function that maps a robot i's position to its observation $\boldsymbol{o}_i(t)$.

In addition, we let the set $G_i \subseteq G$ represent the subset of all tasks in G that robot i believes is assigned to the MRS. All possible combinations of these assignments are represented by the power set $\mathcal{P}(G)$ and valid configurations for the multi-robot mission is denoted as $\mathcal{G} \subseteq \mathcal{P}(G)$.

In this work, we assume that all robots know the location of all tasks G present and the sensor configuration ω_i of each robot i in the system. Given that robots are not able to communicate, we represent this problem as a bilevel resource optimization problem [37]:

Problem 6.1 ***Upper-Level – Epistemic System Tasking:*** *Design a epistemic strategy for an MRS to allocate a subset of tasks from $\mathcal{V}$ to the system at any given time t, accounting for uncertainty in local observations and considering that robots are unable to communicate.*

Problem 6.2 ***Lower-Level – Intent Signaling for Subtask Assignment:*** *Given the subset of tasks to complete from the upper-level optimization, formulate a policy for effective intent signaling for each robot and efficient task completion.*

The mathematical formulation of Problem 6.1 and 6.2 can be expressed as:

$$\begin{aligned}
\min_{x,y} \quad & \sum_{\tau=1}^{|G|} c_\tau x_\tau + \sum_{i=1}^{|\mathcal{R}|} \sum_{\tau=1}^{G} c'_{i\tau} y_{i\tau} && (6.2)\\
\text{subject to} \quad & \sum_{\tau=1}^{|G|} x_\tau \leq K && (6.3)\\
& x_\tau \in \{0,1\} \; \forall \tau \in \{1, \ldots, |G|\}, && (6.4)\\
& y_{i\tau} \in \arg\min_{y'_{i\tau}} \sum_{i=1}^{|\mathcal{R}|} \sum_{\tau=1}^{|G|} b_{i\tau} y'_{i\tau} && (6.5)
\end{aligned}$$

The variable c_i represents the cost associated with selecting task τ represented by the binary variable x_τ. $c'_{i\tau}$ represents the cost associated with assigning robot i to task τ. $b_{i\tau}$ represents the cost in the lower-level problem for assigning robot i to task τ. K is a constant representing the maximum number of tasks that can be selected from the set G. The upper-level objective function minimizes the total cost of selecting tasks and assigning robots, while the lower-level problem ensures the optimal assignment of robots to the selected subset of tasks. The constraints ensure that each task is assigned to exactly one robot if selected from the set G and that each robot is assigned to at most one task.

6.3 Approach

Our proposed framework is designed for a task allocation problem (TAP) in which robots are unable to communicate explicit information, but must instead signal their intent and infer other robot's intent in the system. When solving the TAP, robots must update their beliefs about the state of the system and also empathize with what others might believe. To realize these changes, each robot observes the observable states of each robot and reasons about the system in a hierarchical manner. Initially, each robot evaluates the subtasks that need to be allocated by the system at time t, using epistemic reasoning to converge a common belief about the allocation of tasks within the multi-robot system. Subsequently, each robot examines the evidence related to the movements of all robots in the system and ultimately signals its own intent, employing active inference to reduce the system's free energy. The diagram in Fig. 6.2 illustrates this decentralized framework, where robots first gather observations about the MRS. These observations are then filtered based on previous measurements before generating and assessing allocations for the MRS to execute at time t. Upon generating these solutions, resulting perspective of robot i is denoted as s_i and represents the possible assignments of robots to tasks.

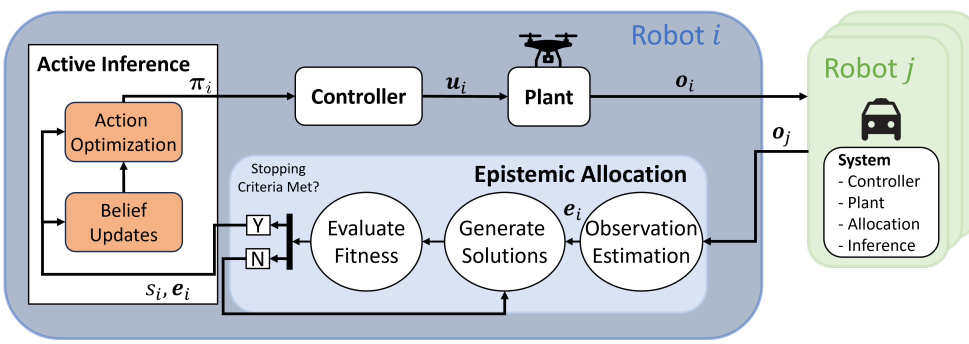


Figure 6.2: Diagram of the proposed approach

In the next sections, we will initially concentrate on the epistemic framing of the problem and how a robot can use higher-order reasoning to update its beliefs. We then show how active inference can be used to model how to measure the results of deeper reasoning to enhance the accuracy of non-communicative robots. We will also explore scalable runtime tools for employing this belief-based inference method before discussing the online epistemic distribution of subtasks to enable more effective reasoning during operation.

6.3.1 Epistemic Framing

In this section, we frame the problem of multi-robot goal selection and coordination in environments where direct communication between robots is not feasible with dynamic epistemic logic (DEL) and epistemic planning. Our approach leverages DEL to model the knowledge, beliefs, and intentions of robots, integrating this with planning algorithms for goal selection and sequence optimization. To limit the possible combinations of execution policies for each robot to infer about the MRS, we use epistemic logic and allow the robots to reason about the system state. For this application, the epistemic language, $\mathcal{L}(\Psi, AP, \mathcal{R})$ is obtained as follows in Backus-Naur form [45]:

$$\phi ::= H(\eta) \mid \phi \wedge \phi \mid \neg\phi \mid K_i\phi \mid B_i\phi$$

where $i \in \mathcal{R}$. $H \in \Psi$ with Ψ being a set of functions that describe the system state. η generally denotes function arguments. $\neg\phi$ and $\phi \wedge \phi$ are propositions that can be negated and form logical conjunctions, where $\phi \in AP$ and AP is a finite set of atomic propositions. We denote the set of possible worlds by W, where each world $w \in W$ represents a distinct state of the system where each robot is assigned to a subset of tasks. $K_i\phi$ and $B_i\phi$ are interpreted as "robot i knows ϕ" and "robot i believes ϕ", respectively. Practically, we consider ϕ to be the generic assignment of a robot to a task.

The accessibility relation R_i represents the uncertainty of robot i at run-time for a global epistemic state $s = (W, (R_i)_{i\in\mathcal{R}}, L, w)$ where $L : W \to AP$ assigns a labeling to each world defined by its true propositions. An accessibility relation R_i for robot i defines which worlds are indistinguishable to i; that is, $R_i(w, v)$ holds if robot i cannot distinguish between worlds w and v. A robot may not be able to distinguish worlds if the *evidence* associated with both worlds is equivalent or similar according to the uncertainty associated with our observations from (6.1).

Sequences of relations are used to represent higher-order knowledge. For example, the statement "robot i knows that robot j knows ϕ" is true in s if and only if $s \models K_iK_j\phi$. This condition is satisfied when ϕ is true in all worlds accessible from w through the composite relation of R_i and R_j. The perspective of robot i on the system state is notated as s_i.

Dynamic epistemic logic is expanded from epistemic logic through action models [8]. These models affect a robot's perception of an event and influence its set of reachable worlds, R_i. A robot may plan to reduce the run-time uncertainty by taking actions. An action a transforms a world w to a world w' such that if $w \models [a]\phi$, then ϕ holds in the resultant world w'. Robots generate plans π_i that are sequences of actions $\pi_i = \langle a_{i1}, a_{i2}, \ldots, a_{ik}\rangle$ leading from an initial state s_i^0 to a goal state $s_i^* \in \Gamma_i$, where Γ_i represents the set of epistemic goal states for robot i.

$$\pi_i = \langle a_{i1}, a_{i2}, \ldots, a_{ip}\rangle \tag{6.6}$$

such that

$$s_i^0 \xrightarrow{a_{i1}} s_{i1} \xrightarrow{a_{i2}} \cdots \xrightarrow{a_{ip}} s_i^* \in \Gamma_i. \tag{6.7}$$

Coordination among robots is achieved through continuous observation and nested belief updates:

$$\boldsymbol{B}_i \leftarrow \boldsymbol{o}_i \tag{6.8}$$

which represents what robot i believes about robot j's goal assignments, noting that $\boldsymbol{B}_i$ represents robot i's nested beliefs about the system such that $\boldsymbol{B}_i\phi \models B_i \dots B_r\phi$ where the subscript denotes the nested belief of the r^{th} robot from the perspective of robot i. The planning process incorporates the robots' knowledge and beliefs:

$$\pi_i = \text{Plan}(s_i^0, \Gamma_i, \boldsymbol{B}_i) \tag{6.9}$$

This allows robots to infer each other's goals:

$$\forall i, j \in \mathcal{R}, \quad B_i[\boldsymbol{o}_j](\Gamma_j) \tag{6.10}$$

If robot i observes that robot j is moving towards goal g_m, it updates its beliefs to $B_i(\Gamma_j = g_m)$ where if $B_i(j \text{ is moving towards } g_m)$ then $\boldsymbol{B}_i(\Gamma_j = g_m)$. Robots update their beliefs based on observations and actions using DEL:

$$\boldsymbol{B}_i \leftarrow a \quad \text{or} \quad \boldsymbol{B}_i \leftarrow \boldsymbol{o}_i \tag{6.11}$$

This framework ensures soundness, where the inference rules correctly reflect the environment's state, and completeness, where sufficient observations lead to accurate goal inference. Combining DEL with epistemic planning offers an effective strategy for managing multi-robot systems without the need for direct communication. However, refining these beliefs and developing a logical policy requires a probabilistic method that can manage complex reasoning. In the subsequent section, we merge the epistemic framework into the active inference model, utilizing nested beliefs and data gathering to signal a robot's intentions and infer the goals of other robots.

6.3.2 Enhanced Reasoning for Active Inference

Active inference robots perform perception and action planning by minimizing variational free energy. To minimize free energy, these robots utilize a generative model that depicts the joint probability of the stochastic variables responsible for their perceptions [29]. Fig. 6.3 shows our generative model for this framework where a robot receives observations $\boldsymbol{o}_i \in \mathcal{O}$, $\forall i \in \mathcal{R}$. Observations are then processed through generalized Bayesian filtering [30], leading to an update in the beliefs of the system. Each robot can then utilize these revised beliefs to forecast the system's behavior. This process results in a robot i creating a set of policies for all robots, but only able to control its own policy. However, these actions affect the environment and, in turn, the state of the system as perceived by other robots. This cycle continues, enabling the robots to infer the intentions of other robots and to use their own control policies to influence the beliefs of others. Active inference is distinct from perception or learning because it involves an active process driven by the goal of producing observations that are minimally surprising.

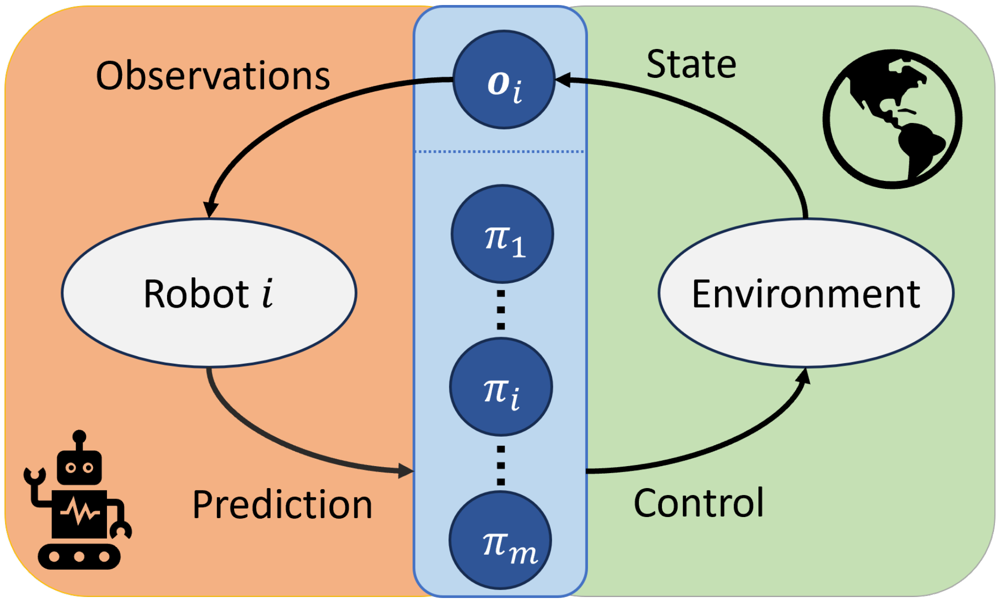


Figure 6.3: Overview of generative model and process used in our multi-robot application. We assume that the state is hidden and the robot is only able to observe using their own on-board sensing capability (e.g., depth sensors, cameras).

The generative model of any i^{th} robot in our approach is mathematically defined similar to the formalism first introduced by [53]; however, we augment this model with continuous states, observations, and actions represented in [67]. We let the dynamics model for the generative model be represented by (6.2), influenced by control inputs $\boldsymbol{u}_i(t)$ and process noise. Next, we formulate the observation likelihood for a robot i's observations $\boldsymbol{o}_i(t)$ at time t as:

$$P(\boldsymbol{o}_i(t) \mid \boldsymbol{x}_i(t), \omega_i) \sim \mathcal{N}(\boldsymbol{o}_i(t); \boldsymbol{h}_i(\boldsymbol{x}_i(t), \omega_i), \Sigma_i) \tag{6.12}$$

where $\boldsymbol{h}_i$ maps the robot i's position $\boldsymbol{x}_i(t)$ to its observation $\boldsymbol{o}_i(t)$ and Σ_i is the observation noise.

The two main components of active inference are belief updates and active selection. In our application, we note that each robot maintains a belief over the possible goal configurations for the multi-robot system. Depending on the application, these goal configurations should represent possible states that will accomplish a pre-defined mission. Each robot maintains this belief about the possible goal configurations that the system is accomplishing at time t. We represent this posterior belief as:

$$Q(\mathcal{G} \mid \boldsymbol{o}_i(t), \omega_i) \tag{6.13}$$

where $\mathcal{G}$ is a subset of possible goal configurations that would accomplish the task allocation problem in (6.5). We use Bayes' rule to update the prior belief $P(\mathcal{G})$ based on the likelihood $P(\boldsymbol{o}_i(t) \mid \mathcal{G}, \omega_i)$ derived from the observations and sensor configurations of robot i. This is modeled as follows:

$$Q(\mathcal{G} \mid \boldsymbol{o}_i(t), \omega_i) \propto P(\boldsymbol{o}_i(t) \mid \mathcal{G}, \omega_i) P(\mathcal{G}). \tag{6.14}$$

These posterior updates are then used in the active selection component; however, we can only approximate the posterior given that we do not have direct access to the true system state. Therefore, we approximate the posterior $q(\mathcal{G})$ as follows:

$$q(\mathcal{G}_i) \propto \boldsymbol{L}_i(\mathcal{G} \mid \boldsymbol{o}_i(t), \Omega) P(\mathcal{G}) \tag{6.15}$$

where robot i's approximate posterior $q(\mathcal{G})$ is proportional to the product of the likelihood function $\boldsymbol{L}_i(\boldsymbol{o}_i(t), \Omega, \mathcal{G})$ and prior $P(\mathcal{G})$. The likelihood function is discussed further in the next sections, but first we define how we use the belief update for the free energy calculation.

By employing active inference, a robot executes a perception-policy loop through the application of the aforementioned matrices to hidden states and observations. In our scenario, perception involves estimating which of the valid goal configurations the system is achieving. At the start of any mission, the MRS might have access to a prior over goal configurations providing each robot with an initial state estimate, which is then refined by subsequent observations.

For anticipated future states, the robot deduces the current hidden goal configuration $\mathcal{G}$ taking into account the expected transitions defined by the control $\boldsymbol{u}_t$ and general dynamics in (6.2). Active inference utilizes an approximate posterior for hidden states and control policies. As demonstrated by the authors in [76], the distribution is most accurately approximated by minimizing the variational free energy (VFE), which is defined at time t as:

$$\boldsymbol{F}(\boldsymbol{x}_i(t), \boldsymbol{u}_i(t), \mathcal{G}, \boldsymbol{o}_i(t), \omega_i) = H[q(\mathcal{G})] + D_{KL}[q(\mathcal{G}) \parallel P(\boldsymbol{x}_i(t) \mid \boldsymbol{o}_i(t), \omega_i)] \tag{6.16}$$

where H is the model uncertainty computed using Shannon entropy and D_{KL} denotes the Kullback-Leibler (KL) divergence. This can be further generalized to expected free energy for a policy π_i:

$$\mathbb{E}_{\pi_i}[\boldsymbol{F}] = \mathbb{E}_{\pi_i}[H[q(\mathcal{G})]] + \mathbb{E}_{\pi_i}[D_{KL}[q(\mathcal{G}) \parallel P(\boldsymbol{x}_i(t) \mid \boldsymbol{o}_i(t), \omega_i)]] \tag{6.17}$$

The expected free energy (EFE) is a metric that integrates the entropy of the variational distribution $Q(\mathcal{G})$ with the expected log-likelihood of the generative model. By minimizing $\boldsymbol{F}$, robots adjust their beliefs to better approximate the true posterior distribution, balancing model complexity with alignment to observed data. The EFE comprises two components that assess the quality of the policy. The first component is the expected Kullbeck-Leibler divergence, which promotes low-risk policies by minimizing the discrepancy between the approximate posterior and the desired outcome. The second component is the expected entropy of the posterior over hidden states, representing the epistemic aspect of the quality score and encouraging policies that reduce uncertainty in future outcomes.

In this section, we let the beliefs over the hidden states ($Q(\mathcal{G})$) for a MRS represent a discrete array of probabilities for any desired goal configuration, $\tilde{g} \in \mathcal{G}$. For example, consider the scenario in Fig. 6.1 where there is one goal that requires two robots and one goal that requires one robot. The valid configurations in $\mathcal{G} = [(0,1,1),(1,0,1),(1,1,0)]$ and the beliefs would represent the probability that any of those configurations were the true state of the system or how the mission would be accomplished since the robots cannot explicitly communicate.

6.3.3 Salience

As mentioned in the previous section, the likelihood function is an important function to update a robot's approximate posterior belief about the hidden states. A factor for how to interpret likelihood based on observations is salience. Salience describes how prominent or emotionally striking something is. In neuroscience, salience is an attentional mechanism that helps organisms learn and survive by allowing them to focus on the most relevant sensory data. In our application, salience is the evidence that a robot is aligned with a goal $g_j \in G$. The set G is different in that G is the set of all goals in an environment, but $\mathcal{G}$ is the valid goal configurations for the multi-robot mission such that $\mathcal{G} \subseteq \mathcal{P}(G)$ from $\mathcal{G}$.

We note that previous salience functions used in active inference and robotics literature such as in [52, 53], typically only use up to first-order reasoning to define their evidence and subsequent posterior belief. We begin our formulation generally with the salience value defined as:

$$\boldsymbol{e}_i^{(k)} \leftarrow \boldsymbol{v}_i^{(k)}(\boldsymbol{o}_i, \Omega, G) = \exp\left(-\frac{1}{\eta}\boldsymbol{h}_i^{(k)}(\boldsymbol{o}_i, \Omega, G)\right) \tag{6.18}$$

where the array $\boldsymbol{e}_i^{(k)} \in \mathbb{R}^{|G|}$ is the mapping of observations to evidence for goals in G from the perspective of robot i and given the sensor configurations of all robots in the system Ω. The superscript k denotes the level of reasoning at which the robot is evaluating its observations. Since a robot's observations are independent of other robot's observations, we can aggregate the evidence associated with each robot i's perspective of other robots. The function is a general function that maps a positive evidence value to each goal configuration in G such that:

$$\boldsymbol{h}_i^{(k)}(\boldsymbol{o}_i, \Omega, G) = \sum_{a \in \mathcal{R}_k} \left[h_{ij,a}^{(k)}(\boldsymbol{o}_i, \Omega, g_j)\right]_{j=1}^{|G|} \tag{6.19}$$

where $\mathcal{R}_i^k$ denotes the subset of robots that are considered for k^{th}-order reasoning from the perspective of robot i, the value $h_{ij,a}(\omega_i, \Omega, g_j)$ is positive for a goal $g_j \in G$, and robot a that indicates if a robot is aligned with a goal g_j.

One can observe that in a heterogeneous system, a robot may not always be able to consider other robots' perspectives if the perspective is not measurable. As such, evidence $h_{ij,a}^{(k)}(\boldsymbol{o}_i, \Omega, G) = 0$ when a robot i is unable to compute the evidence from the perspective of robot a ($\omega_a \succ \omega_i$). This is represented in the set $\mathcal{R}_i^k \in \mathcal{R}$ and, as a result, no new information is mapped by robot i from robot a's perspective and is not able to inform the variational distribution for non-measurable perspectives for second- and third-order reasoning.

Higher-order reasoning can be aggregated iteratively to form a comprehensive joint probability distribution. To calculate the likelihood of a robot being aligned with any particular goal in G, we let the probability distribution for k^{th}-order reasoning be represented as:

$$P_i(G) = \sigma\left(\sum_k \boldsymbol{e}_i^{(k)}\right) \tag{6.20}$$

where $\boldsymbol{e}_i^{(k)}$ represents the evidence gathered using (6.18) and σ is representative of the softmax function. The belief over the goal configurations specified by $\mathcal{G}$ can be inferred using a joint probability distribution of the result from (6.20). We formulate the joint probability distribution we first initialize as:

$$P_i^J(g) = 1 \ \forall g \in G^n \tag{6.21}$$

where G^n represents all the possible combinations of n robots assigned to $|G|$ tasks. Then we calculate the joint probability distribution for all possible goal configurations as

$$P_i^J(\tilde{g}) = P_i^J(g_1, \ldots, g_n) = \prod_{i=1}^{n} P_i(g_i) \tag{6.22}$$

where $P_i(g_i) \in P_i(G)$ and $\tilde{g} \in \mathcal{G}$. Additionally, the goals $g_1, \ldots, g_n$ represents the allocation of a goal in the set G to each robot. The result in (6.22) gives the probability for all configurations in $\mathcal{G}$ and we extract and normalize the subset of valid configurations to form a distribution:

$$L_i\left(\mathcal{G} \mid \boldsymbol{o}_i(t), \Omega\right) = \frac{P_i^J(\tilde{g})}{\sum_{\tilde{g}' \in \mathcal{G}} P_i^J(\tilde{g}')} \ \forall \tilde{g} \in \mathcal{G} \tag{6.23}$$

where the likelihood $L_i\left(\mathcal{G} \mid \boldsymbol{o}_i(t), \Omega\right)$ can be used to update the prior from (6.15).

The joint likelihood associated with higher-order reasoning can increase the robustness of the overall system because of the integration of information across multiple layers. Suppose each independent likelihood has variance (σ_i^2). The combined variance in the joint likelihood can be lower due to the aggregation of information. In addition, joint likelihood can better manage the bias-variance tradeoff. While independent likelihoods may lead to higher variance due to lack of dependency modeling, a joint likelihood balances the bias introduced by modeling dependencies and the variance reduction due to joint estimation. Lastly, using a joint likelihood fits well within the Expectation Maximization (EM) algorithm, which iteratively maximizes the expected log-likelihood. The joint likelihood provides a more accurate expectation step, leading to more stable estimates.

Lemma 6.1 *In a multi-robot system where higher-order reasoning (second- and third-order) is based on first-order measurements, incorporating dependencies through joint likelihoods reduces the overall variance of the parameter estimates compared to first-order reasoning, thereby enhancing robustness.*

Proof: Higher-order reasoning in multi-robot systems can significantly enhance the solution quality by aligning the robots' beliefs and reducing uncertainty. Initially, each robot i evaluates the evidence $\boldsymbol{e}^(1)_i$ over goals $g \in G$, yielding a first-order probability

$$P_i^{(1)}(g) = \frac{\exp(\boldsymbol{e}_i^{(1)}(g))}{\sum_{g' \in G} \exp(\boldsymbol{e}_i^{(1)}(g'))}. \tag{6.24}$$

where However, this does not account for the beliefs or actions of other robots. By incorporating higher-order reasoning, robots consider the expected probabilities from others. For example, in second-order reasoning, robot i updates its belief to

$$P_i^{(2)}(g) = \frac{\exp\left(\boldsymbol{e}_i^{(1)}(g) + \sum_{j \neq i} \mathbb{E}[P_j^{(1)}(g)]\right)}{\sum_{g' \in G} \exp\left(\boldsymbol{e}_i^{(1)}(g') + \sum_{j \neq i} \mathbb{E}[P_j^{(1)}(g')]\right)}. \tag{6.25}$$

This generalizes for k-th order reasoning. The belief alignment using joint probabilities over goal configurations $\tilde{g} \in \mathcal{G}$ is as follows. We define the joint probability $P_i^J(\tilde{g})$ for robot i over the goal configurations:

$$P_i^J(\tilde{g}) = \prod_{g \in \tilde{g}} P_i(g). \tag{6.26}$$

Higher-order reasoning minimizes the Kullback-Leibler divergence between the robots' beliefs,

$$\sum_{i=1}^{n} D_{\mathrm{KL}}\left(P_i^J(\tilde{g}) \parallel \mathbb{E}[P_i^J(\tilde{g})]\right), \tag{6.27}$$

thereby aligning their beliefs more closely. This alignment reduces the entropy of each robot's belief distribution,

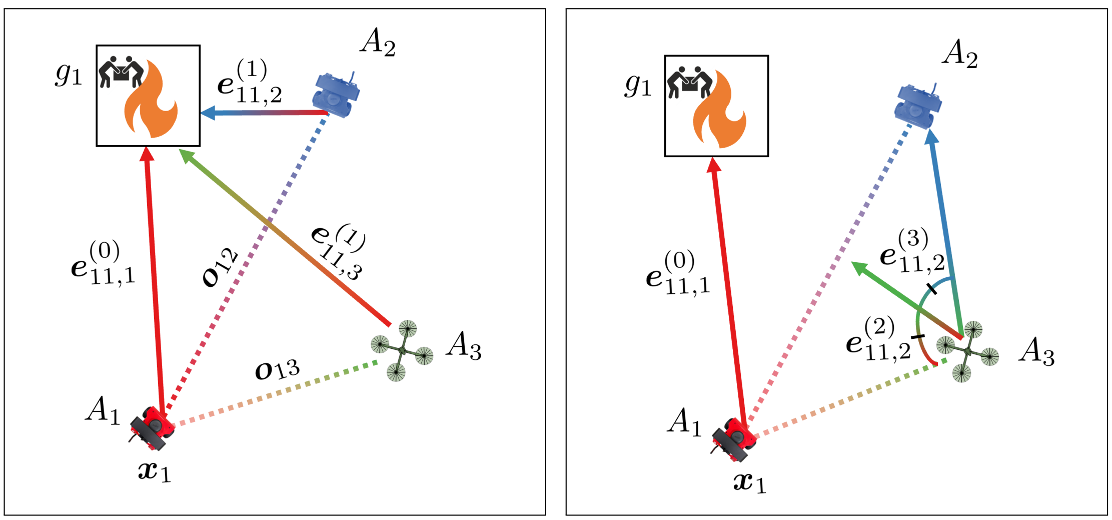
$$H(P_i^J(\tilde{g})) = -\sum_{\tilde{g} \in \mathcal{G}} P_i^J(\tilde{g}) \log P_i^J(\tilde{g}), \tag{6.28}$$

because higher-order reasoning incorporates more information, leading to

$$H(P_i^J(\tilde{g})) \le H(P_i^{(1)}(\tilde{g})). \tag{6.29}$$

Improved belief alignment enhances coordination among robots, as they can more accurately predict each other's actions, maximizing the expected utility. Thus, higher-order reasoning helps robots to converge towards optimal solutions, reducing variance and improving overall system performance. ■

Higher-order reasoning models, even when based on first-order measurements, reduce the overall variance of parameter estimates by capturing dependencies and interactions between different layers of reasoning. This leads to enhanced robustness, as the model can provide more stable and reliable estimates in the presence of noise and uncertainties.



(a) Zero and First Order

(b) Second and Third Order

Figure 6.4: Pictorial depiction of observation mapping to evidence and depth of reasoning.

We motivate these formulations with a simple example shown in Fig. 6.4. Consider a multi-robot system consisting of two ground vehicles equipped with depth sensors and one aerial vehicle equipped with a monocular camera. The robots are attempting to allocate tasks without communication and ground robot A_1 is assessing evidence between a goal g_1 and other two robots A_2, A_3. In Fig. 6.4(a), we show the observations and subsequent evidence calculation for A_1's sensor configuration which allows A_1 to calculate the distance to the goal for both itself (zero-order reasoning) and other robots (first-order reasoning). In Fig. 6.4(b), since A_1's

We note that through minimizing the expected free energy and using the likelihood function to update the posterior we will maximize the probability of converging to a common goal configuration without any communication. However, the Bayesian method inherently suffers from the curse of dimensionality, since the number of possible joint configurations grows as the number of robots and/or the number of goals grows. Thus, in the next section, we introduce epistemic planning for reducing the goal configurations possible in each iteration.

6.3.4 Epistemic Allocation for Dimensionality Reduction

Epistemic planning involves robots making decisions based on their knowledge and beliefs, with the aim of reducing uncertainty and achieving goals in a shared environment. In this context, the function for choosing subset goals plays a crucial role. Let $B_i(g_j)$ represent robot i's belief about goal g_j. The belief update mechanism uses a softmax function over the evidence e_{ij} between robot i and goal j, formalized as:

$$B_i(g_j) = \sigma\left(\sum_k \boldsymbol{e}_{ij}^{(k)}\right) \tag{6.30}$$

This update represents the probability that robot i believes goal g_j is achievable, given the evidence. With higher-order reasoning, robots can collectively maximize the diversity of evidence. This involves each robot considering not just their perspective but the perspectives of all robots to select goals that maximize the collective knowledge. Formally, the set of chosen goals G_c can be described as:

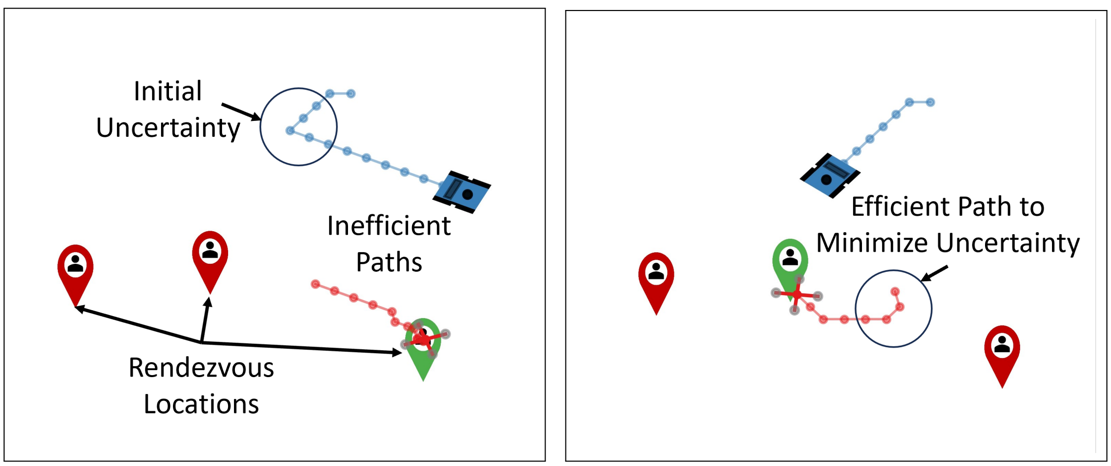
$$G_c = \{g_j \mid \arg\max_j B_i(g_j), \forall i\} \tag{6.31}$$

This selection ensures that the chosen goals maximize the diversity and coverage of evidence across all robots, thus enhancing the collective knowledge and reducing overall uncertainty.

6.4 Simulations

This section showcases the outcomes obtained through python simulations of our method executed by teams of two to five robots. We focus on two different goal configurations. In the first, we show a comparison between a first-order reasoning baseline inspired by [53, 72] and higher-order reasoning when converging to a single goal given multiple goals to choose from. Random configurations of goal locations, robot sensor configurations, and starting positions of robots are generated for 50 trials per each combination of robots and goals. The number of robots varies between two and five, whereas the number of goals ranges between three and five. The size of the environment for each test is set at $30m \times 30m$ and the maximum number of iterations or time steps per simulation is set to 100 iterations. The maximum velocity for each robot is 1m/s and the multi-robot system has converged if all robots reach a single goal within 100 iterations and are within 1.5m of the position of the goal. Observation error is normally distributed as $\mathcal{N}(0, 0.5)$ for distance measurements and

$\mathcal{N}(0, 0.1)$ for angular measurements. The multi-robot system is randomly spawned with one of two different types of sensor configurations. One sensor configuration is a range sensor (ω_1) which can observe distance measurements to other robots, while the other configuration (ω_2) can measure relative angles to the observing robot's position. We consider that $\omega_1 \succ \omega_2$ since the robots are capable of abstracting angle measurements from distance measurements. We show a sample result in Fig. 6.5 comparing first-order reasoning and higher-order reasoning in a sample environment where two robots with two different sensor configurations are trying to converge to a single goal. The red UAV can observe angles while the blue UGV can measure distances.



(a) First-order reasoning (b) Higher-order reasoning

Figure 6.5: Sample comparison of where in first-order reasoning the red UAV does not consider the blue UGV's perception of its movements

As shown in Fig. 6.5(a), the red UAV does not consider the blue UGV's perception of its movements to gain more certainty about its observations before the robots end up converging to a goal farther away. In contrast, higher-order reasoning allowed the red UAV to make small movements to gain more certainty about its observations before converging to the closer goal. The results shown in this example explain how first-order reasoning is more prone to fail when the objective is to rendezvous at a goal. The results of all trials are depicted in Fig. 6.6 which show that higher-order reasoning results in a higher success rate and that an increase in complexity does not result in a significant decrease in success.

In the second set of trials, we show a comparison between the same first-order reasoning baseline and higher-order reasoning when converging to separate goals. A random configuration of goal locations, robot sensor configurations, and initial robot positions are randomly generated for 50 trials per number of robots which ranges from two and five robots. In these comparisons, the number of tasks is equal to the number of robots. The environment size for each trial is set at $30m \times 30m$ and the maximum number of iterations is set to 100 iterations. The maximum velocity

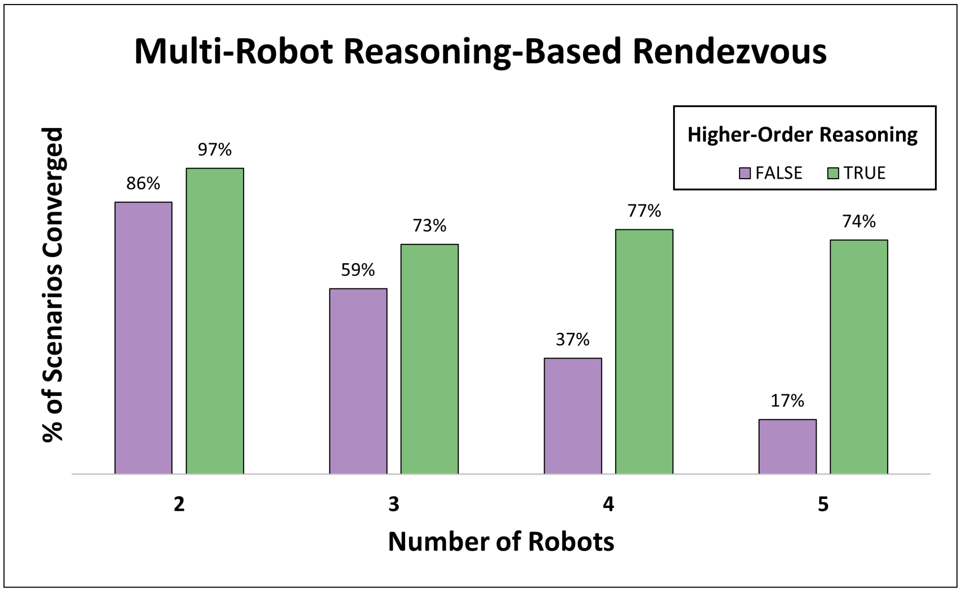
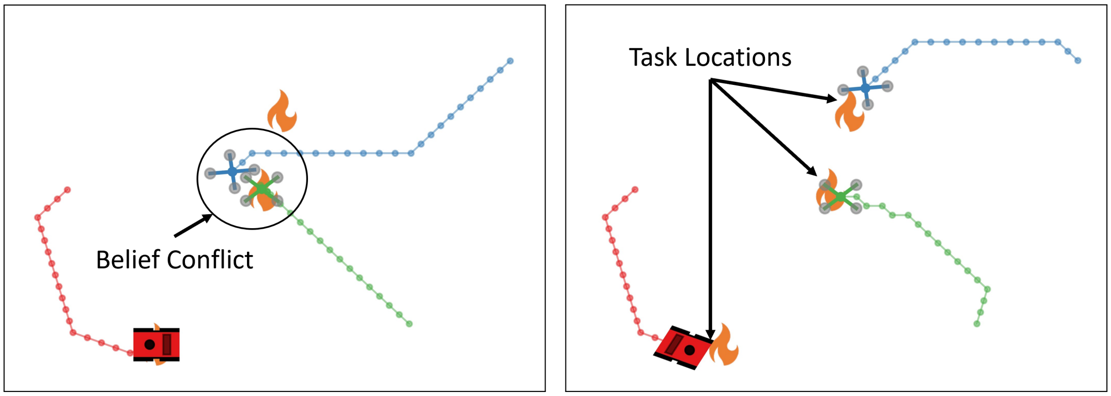


Figure 6.6: Comparison of using first-order versus higher-order reasoning for a rendezvous mission.

for each robot is 1m/s and the multi-robot system has converged if all robots reach a separate goal within 100 iterations and are within 1.5m of the position of the goal. Observation error is normally distributed as $\mathcal{N}(0, 0.5)$ for distance measurements and $\mathcal{N}(0, 0.1)$ for angular measurements. The multi-robot system is randomly spawned with one of two different types of sensor configurations. One sensor configuration is a range sensor (ω_1) which can observe distance measurements to other robots, while the other configuration (ω_2) can relative angles to the observing robot's position. As in the first set of trials, we consider that $\omega_1 \succ \omega_2$ since the robot's can abstract angle measurements from distance measurements. We show a sample result in Fig. 6.7 comparing first-order reasoning and higher-order reasoning in a sample environment where two robots with two different sensor configurations are trying to converge to a single goal. The blue and green UAV can observe angles while the red UGV can measure distances.

As illustrated in Fig. 6.7(a), the blue and green UAVs are incapable of resolving their belief discrepancies. In contrast, Fig. 6.7(b) demonstrates that through higher-order reasoning, the UAVs can convey and understand intentions, allowing them to resolve task assignment conflicts without communication. The results of all trials are depicted in Fig. 6.8 which show that higher-order reasoning results in a higher success rate and that an increase in complexity does not result in a significant decrease in success similar to the results from Fig. 6.6.



(a) First-order reasoning (b) Higher-order reasoning

Figure 6.7: Illustration of our simulations showing that first-order reasoning fails to allow the blue and green UAVs to convey their intentions or reason from the other robot's perspective, resulting in both robots converging on the same task. Higher-order reasoning allows the UAVs to interpret and signal clear intentions, successfully completing different tasks.

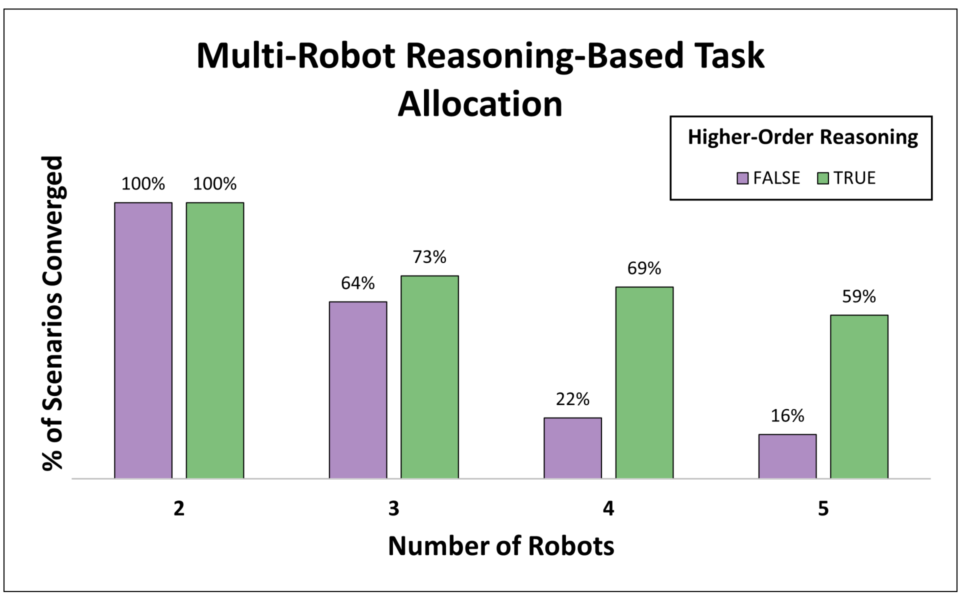


Figure 6.8: Comparison of using first-order versus higher-order reasoning for a task allocation mission.

6.5 Discussion

In this work, we demonstrated the effectiveness of utilizing higher-order reasoning for multi-robot systems (MRS) operating under communication constraints. By integrating theory of mind (ToM) and epistemic planning, our proposed framework allows robots to infer the knowledge and intentions of others based on their observations and last known states. This approach enables robots to cooperate and achieve common goals even when explicit communication is not possible.

Our findings show that higher-order reasoning, extending up to the third level, significantly enhances the ability of MRS to converge to correct belief states and complete tasks efficiently. The hierarchical epistemic planning combined with active inference for runtime plan adaptation provides a robust solution to mitigate the challenges of limited communication in heterogeneous robot teams.

Part IV

Epilogue

Chapter 7

Conclusions and Future Work

In this chapter, we conclude the dissertation with an overview of what we have accomplished and learned, followed by a discussion of real-world applications for this work and also any possible directions we could take for future work to build on what we have achieved thus far.

7.1 Conclusions

In this dissertation, we have introduced several innovative frameworks for multi-robot systems that utilize higher-order reasoning to enhance the reliability and robustness of mission planning and execution methodologies for complex multi-robot applications in environments where communication is unreliable or restricted. The proposed approaches mainly emphasize the integration of epistemic planning into the multi-robot reasoning process to develop logical frameworks for traditional problems such as the multiple traveling salesman problem or multi-robot exploration. We have demonstrated that similar techniques can be applied to scenarios where communication is restricted, allowing robots to reason up to the third-order level. All presented techniques were validated through extensive simulations and hardware experiments to confirm their generality and applicability to real robots.

First, we introduced our research on coordinated exploration, rendezvous, and task distribution. We employed a Sobel edge detection frontier-based technique, assigning each robot a distinct exploration task. During their exploration, robots could encounter new tasks and were equipped to reason about these tasks using a weighted multi-objective optimization algorithm. Robots that did not find new tasks would rendezvous to update their exploration assignments and could also be assigned new tasks communicated by other agents or due to their absence during the rendezvous. We demonstrated that our method outperformed those without a reasoning step or where agents explored the environment in formation. Nonetheless, we observed that extensive backtracking was necessary for rendezvous in unknown environments, a requirement not needed in mostly known environments.

To alleviate this limitation, we then addressed the challenge of multi-robot exploration and rendezvous in partially-known environments, further complicating the scenario by introducing potential failures or disturbances. In these situations, robots might not perform as previously agreed upon at an unknown time while disconnected from the rest of the team. To address this, we propose epistemic planning for a multi-robot system, which reduces uncertainty propagation to a finite set of particles. This allows robots to update their beliefs upon observing the particles implicitly without the need for communication. We demonstrated that our method outperformed those requiring robots to stay within communication range, and performance times were closely correlated with scenarios where robots could communicate continuously.

Building on this epistemic planning framework, we extended the method for heterogeneous multi-robot teams to execute complex tasks in environments that are partially known and where communication is limited. We proposed a dynamic task assignment and gossiping protocol, enabling agents to communicate with other robots as necessary to accomplish tasks. This method employed a decentralized genetic algorithm that allowed robots to locally update their epistemic state and dynamically assign tasks.

A significant contribution of our work is the development of an epistemic planning framework for scenarios where tasks are predetermined, but robots might fail during execution while being disconnected from the system. In this methodology, we employ a centralized genetic algorithm to design efficient 'crossings' in the mTSP solution, enabling robots to update their epistemic state locally. Subsequently, we utilize an online Monte Carlo tree search to orchestrate gossiping and task allocation processes, guaranteeing that all tasks are completed. Our results demonstrate that this approach outperforms a baseline heuristic that waits for all robots to return to their final depot before retracing the paths of failed robots to finish tasks.

Finally, we demonstrated how existing frameworks can be expanded to situations where robots cannot communicate with each other. Some approaches have tackled this issue by employing a theory of mind (ToM) framework, but generally, they only enable agents to perform first-order reasoning based on observations. Conversely, we introduced a solution mechanism that features efficient online plan adaptation through active inference to convey intentions and reason about a robot's own beliefs and the beliefs of others in the system, along with a hierarchical epistemic planning framework to iteratively consider the current MRS mission state. This method surpasses first-order reasoning techniques previously used for multi-agent, multi-goal scenarios.

7.2 Discussion and Possible Future Work

There are numerous intriguing research questions to explore. One promising direction is adopting a macro-action reinforcement learning approach that utilizes epistemic beliefs about the system. In our framework, each robot can exist in a finite set of states, such as gossiping, accomplishing a task, or exploring. Using macro-actions to describe these states would reduce the decision space, and the solution to a reinforcement learning approach could provide a distributed policy for complex interactions over a continuous space. This reduction in decision space would decrease computational complexity for each vehicle when replanning is required. Inspired by the work of [64], we hypothesize that using high-level representations of the belief space can facilitate scalable solutions and account for future rewards over any sequential actions. This approach could allow robots to make more informed decisions based on their understanding of the entire system's state, leading to more efficient and effective operations. By integrating macro-actions with epistemic planning, we aim to enhance the decision-making process, making it feasible for large-scale, multi-robot systems to operate in dynamic environments with reduced computational overhead.

Additionally, another research direction involves decreasing the computational complexity of our approach for large-scale systems by introducing sub-teaming. This strategy would ideally lead to a locally optimal execution policy, allowing each robot to propagate beliefs only for their sub-team or for a simplified representation of other sub-teams. Previous research by authors in [79] demonstrated that by constraining communication, robot swarms could better adapt to changes. Merging this concept with our current framework would enable robots to accomplish tasks without needing to communicate with disconnected members of other teams, thus saving valuable time. In this sub-teaming approach, robots would only need to predict and plan based on a subset of the total system. However, optimizing the allocation of members to sub-teams is a complex problem that should consider the tasks to be accomplished, the capabilities of the robots, and the initial operating plan for each robot. This optimization could involve developing algorithms that dynamically form and re-form sub-teams based on the evolving mission requirements and environmental conditions.

Future work could also explore how to seamlessly integrate these approaches into our current epistemic planning framework. This integration would involve designing algorithms that allow for efficient macro-action planning and sub-teaming strategies, ensuring that the system remains robust and adaptive to changes. Additionally, investigating the potential for reinforcement learning to improve the adaptability and resilience of the multi-robot system in the face of uncertainties and dynamic environments would be a valuable direction. These future research directions aim to enhance our current work on multi-robot systems operating in complex, communication-constrained environments. Establishing an extensive framework that leverages macro-action reinforcement learning and sub-teaming strategies will allow multi-robot systems to execute complex tasks with reduced computational load and heightened operational efficiency.

Another possible avenue for future work involves addressing adversarial scenarios in which robots might encounter hostile agents or deceptive information. In such cases, the use of the theory of mind and epistemic planning could be crucial. Using theory of mind, robots can infer the intentions and deceptive strategies of adversarial agents, allowing them to anticipate and counteract hostile actions. Epistemic planning can help maintain robust cooperation among robots even when facing adversaries by continuously updating their beliefs and strategies based on observed behaviors. This approach can ensure that the multi-robot system remains resilient and adaptive, effectively mitigating the risks posed by adversarial situations and enhancing overall mission success.

Overall, multi-robot systems significantly increase the efficiency, robustness, and effectiveness of tasks traditionally performed by single robots. The advanced higher-order reasoning techniques developed and presented in this dissertation empower robots to exhibit an unprecedented level of intelligence, surpassing previous capabilities.

Bibliography

[1] Francesco Amigoni, Jacopo Banfi, and Nicola Basilico. "Multirobot exploration of communication-restricted environments: A survey". In: *IEEE Intelligent Systems* 32.6 (2017), pp. 48–57.

[2] John R Anderson, Daniel Bothell, Michael D Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin. "An integrated theory of the mind." In: *Psychological review* 111.4 (2004), p. 1036.

[3] Torsten Andre and Christian Bettstetter. "Collaboration in multi-robot exploration: to meet or not to meet?" In: *Journal of intelligent & robotic systems* 82.2 (2016), pp. 325–337.

[4] Arash Asgharivaskasi and Nikolay Atanasov. "Active Bayesian Multi-class Mapping from Range and Semantic Segmentation Observations". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 1–7.

[5] Simon Baron-Cohen, Alan M Leslie, and Uta Frith. "Does the autistic child have a theory of mind?" In: *Cognition* 21.1 (1985), pp. 37–46.

[6] Mandell Bellmore and Saman Hong. "Transformation of multisalesman problem to the standard traveling salesman problem". In: *Journal of the ACM (JACM)* 21.3 (1974), pp. 500–504.

[7] Thomas Bolander and Mikkel Birkegaard Andersen. "Epistemic planning for single-and multi-agent systems". In: *Journal of Applied Non-Classical Logics* 21.1 (2011), pp. 9–34.

[8] Thomas Bolander, Lasse Dissing, and Nicolai Herrmann. "DEL-based Epistemic Planning for Human-Robot Collaboration: Theory and Implementation". In: *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*. Vol. 18. 1. 2021, pp. 120–129.

[9] Lauren Bramblett and Nicola Bezzo. "Epistemic planning for multi-robot systems in communication-restricted environments". In: *Frontiers in Robotics and AI* 10 (2023), p. 1149439.

[10] Lauren Bramblett, Shijie Gao, and Nicola Bezzo. "Epistemic Prediction and Planning with Implicit Coordination for Multi-Robot Teams in Communication Restricted Environments". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 5744–5750.

[11] Lauren Bramblett, Branko Miloradovic, Patrick Sherman, Alessandro V Papadopoulos, and Nicola Bezzo. "Robust Online Epistemic Replanning of Multi-Robot Missions". In: *arXiv preprint arXiv:2403.00641* (2024).

[12] Lauren Bramblett, Rahul Peddi, and Nicola Bezzo. "Epistemic Planning for Heterogeneous Robotic Systems". In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2022, pp. 12706–12712.

[13] Guilhem Buisan and Rachid Alami. "A human-aware task planner explicitly reasoning about human and robot decision, action and reaction". In: *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*. 2021, pp. 544–548.

[14] 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*. IEEE. 2011, pp. 3254–3259.

[15] Stephanie M Carlson, Melissa A Koenig, and Madeline B Harms. "Theory of mind". In: *Wiley Interdisciplinary Reviews: Cognitive Science* 4.4 (2013), pp. 391–402.

[16] Ozan Çatal, Tim Verbelen, Toon Van de Maele, Bart Dhoedt, and Adam Safron. "Robot navigation as hierarchical active inference". In: *Neural Networks* 142 (2021), pp. 192–204.

[17] Kyle Cesare, Ryan Skeele, Soo-Hyun Yoo, Yawei Zhang, and Geoffrey Hollinger. "Multi-UAV exploration with limited communication and battery". In: *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2015, pp. 2230–2235.

[18] Konstantinos Charalampous, Angelos Amanatiadis, and Antonios Gasteratos. "Efficient robot path planning in the presence of dynamically expanding obstacles". In: *International Conference on Cellular Automata*. Springer. 2012, pp. 330–339.

[19] Genshe Chen and Jose B Cruz. "Genetic algorithm for task allocation in UAV cooperative control". In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*. 2003, p. 5582.

[20] Jie Chen, Xianguo Qing, Fang Ye, Kai Xiao, Kai You, and Qian Sun. "Consensus-based bundle algorithm with local replanning for heterogeneous multi-UAV system in the time-sensitive and dynamic environment". In: *The Journal of Supercomputing* 78.2 (2022), pp. 1712–1740.

[21] Ivano A Ciardelli and Floris Roelofsen. "Inquisitive dynamic epistemic logic". In: *Synthese* 192.6 (2015), pp. 1643–1687.

[22] Rafael Gonçalves Colares and Luiz Chaimowicz. "The next frontier: combining information gain and distance cost for decentralized multi-robot exploration". In: *Proceedings of the 31st Annual ACM Symposium on Applied Computing*. 2016, pp. 268–274.

[23] Micael S Couceiro, Rui P Rocha, and Nuno MF Ferreira. “A novel multi-robot exploration approach based on particle swarm optimization algorithms”. In: *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*. IEEE. 2011, pp. 327–332.

[24] George Dantzig, Ray Fulkerson, and Selmer Johnson. “Solution of a large-scale traveling-salesman problem”. In: *Journal of the operations research society of America* 2.4 (1954), pp. 393–410.

[25] Marc Peter Deisenroth, Gerhard Neumann, Jan Peters, et al. “A survey on policy search for robotics”. In: *Foundations and trends in Robotics* 2.1-2 (2013), pp. 388–403.

[26] Thorsten Engesser, Thomas Bolander, Robert Mattmüller, and Bernhard Nebel. “Cooperative epistemic multi-agent planning for implicit coordination”. In: *arXiv preprint arXiv:1703.02196* (2017).

[27] Jonathan Fink, Alejandro Ribeiro, and Vijay Kumar. “Robust control for mobility and wireless communication in cyber–physical systems with application to robot teams”. In: *Proceedings of the IEEE* 100.1 (2011), pp. 164–178.

[28] Mirgita Frasheri, Branko Miloradović, Lukas Esterle, and Alessandro V. Papadopoulos. “GLocal: A Hybrid Approach to the Multi-Agent Mission Re-Planning Problem”. In: *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*. 2023, pp. 1696–1703. DOI: 10.1109/SSCI52147.2023.10371893.

[29] Karl Friston, Thomas FitzGerald, Francesco Rigoli, Philipp Schwartenbeck, Giovanni Pezzulo, et al. “Active inference and learning”. In: *Neuroscience & Biobehavioral Reviews* 68 (2016), pp. 862–879.

[30] Karl Friston, Klaas Stephan, Baojuan Li, and Jean Daunizeau. “Generalised filtering.” In: *Mathematical Problems in Engineering* 2010 (2010).

[31] Wenshuo Gao, Xiaoguang Zhang, Lei Yang, and Huizhong Liu. “An improved Sobel edge detection”. In: *2010 3rd International conference on computer science and information technology*. Vol. 5. IEEE. 2010, pp. 67–71.

[32] Yuman Gao, Yingjian Wang, Xingguang Zhong, Tiankai Yang, Mingyang Wang, Zhixiong Xu, Yongchao Wang, Yi Lin, Chao Xu, and Fei Gao. “Meeting-merging-mission: A multi-robot coordinate framework for large-scale communication-limited exploration”. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2022, pp. 13700–13707.

[33] Bruno Duarte Gouveia, David Portugal, Daniel C Silva, and Lino Marques. "Computation sharing in distributed robotic systems: A case study on SLAM". In: *IEEE Transactions on Automation Science and Engineering* 12.2 (2014), pp. 410–422.

[34] Meng Guo and Michael M Zavlanos. "Multirobot data gathering under buffer constraints and intermittent communication". In: *IEEE transactions on robotics* 34.4 (2018), pp. 1082–1097.

[35] Samira Hayat, Evşen Yanmaz, Timothy X Brown, and Christian Bettstetter. "Multi-objective UAV path planning for search and rescue". In: *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2017, pp. 5569–5574.

[36] Mark K Ho, Rebecca Saxe, and Fiery Cushman. "Planning with theory of mind". In: *Trends in Cognitive Sciences* 26.11 (2022), pp. 959–971.

[37] Pei-Qiu Huang, Qingfu Zhang, and Yong Wang. "Bilevel optimization via collaborations among lower-level optimization tasks". In: *IEEE Transactions on Evolutionary Computation* (2023).

[38] Simon Hunt, Qinggang Meng, Chris Hinde, and Tingwen Huang. "A consensus-based grouping algorithm for multi-agent cooperative task allocation with complex requirements". In: *Cognitive computation* 6 (2014), pp. 338–350.

[39] Sarah Al-Hussaini, Jason M Gregory, and Satyandra K Gupta. "Generation of context-dependent policies for robot rescue decision-making in multi-robot teams". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 4317–4324.

[40] Sarah Al-Hussaini, Jason M Gregory, and Satyandra K Gupta. "Generation of context-dependent policies for robot rescue decision-making in multi-robot teams". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 4317–4324.

[41] Miguel Juliá, Arturo Gil, and Oscar Reinoso. "A comparison of path planning strategies for autonomous exploration and mapping of unknown environments". In: *Autonomous Robots* 33.4 (2012), pp. 427–444.

[42] Michael Jünger, Gerhard Reinelt, and Giovanni Rinaldi. "The traveling salesman problem". In: *Handbooks in operations research and management science* 7 (1995), pp. 225–330.

[43] Alaa Khamis, Ahmed Hussein, and Ahmed Elmogy. "Multi-robot task allocation: A review of the state-of-the-art". In: *Cooperative robots and sensor networks 2015* (2015), pp. 31–51.

[44] Oussama Khatib. "Real-time obstacle avoidance for manipulators and mobile robots". In: *Autonomous robot vehicles*. Springer, 1986, pp. 396–404.

[45] Donald E Knuth. “Backus normal form vs. backus naur form”. In: *Communications of the ACM* 7.12 (1964), pp. 735–736.

[46] Levente Kocsis and Csaba Szepesvári. “Bandit based monte-carlo planning”. In: *European conference on machine learning*. Springer. 2006, pp. 282–293.

[47] Christopher Krupenye, Fumihiro Kano, Satoshi Hirata, Josep Call, and Michael Tomasello. “Great apes anticipate that other individuals will act according to false beliefs”. In: *Science* 354.6308 (2016), pp. 110–114.

[48] ZHANG Kuang-wei, ZHANG Shao-jie, CHEN Yan, PAN Hong, and ZHANG Chen-guang. “Simulation research on pipeline map system based on multi-robot queue cooperation”. In: *2020 International Conference on Communications, Information System and Computer Engineering (CISCE)*. IEEE. 2020, pp. 387–390.

[49] Séverin Lemaignan and Pierre Dillenbourg. “Mutual modelling in robotics: Inspirations for the next steps”. In: *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*. 2015, pp. 303–310.

[50] Shen Lin and Brian W Kernighan. “An effective heuristic algorithm for the traveling-salesman problem”. In: *Operations research* 21.2 (1973), pp. 498–516.

[51] Tony X Lin, Esen Yel, and Nicola Bezzo. “Energy-aware persistent control of heterogeneous robotic systems”. In: *2018 Annual American Control Conference (ACC)*. IEEE. 2018, pp. 2782–2787.

[52] Pierre Lison, Carsten Ehrler, and Geert-Jan M Kruijff. “Belief modelling for situation awareness in human-robot interaction”. In: *19th International Symposium in Robot and Human Interactive Communication*. IEEE. 2010, pp. 138–143.

[53] Domenico Maisto, Francesco Donnarumma, and Giovanni Pezzulo. “Interactive inference: a multi-agent model of cooperative joint actions”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2023).

[54] Sandeep Manjanna, Alberto Quattrini Li, Ryan N Smith, Ioannis Rekleitis, and Gregory Dudek. “Heterogeneous multi-robot system for exploration and strategic water sampling”. In: *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2018, pp. 4873–4880.

[55] Bastien Maubert, Sophie Pinchinat, and François Schwarzentruber. “Reachability games in dynamic epistemic logic”. In: *arXiv preprint arXiv:1905.12422* (2019).

[56] Bastien Maubert, Sophie Pinchinat, François Schwarzentruber, and Silvia Stranieri. "Concurrent games in dynamic epistemic logic". In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*. 2021, pp. 1877–1883.

[57] David Q Mayne and Hannah Michalska. "Receding horizon control of nonlinear systems". In: *Proceedings of the 27th IEEE Conference on Decision and Control*. IEEE. 1988, pp. 464–465.

[58] Petra Mazdin and Bernhard Rinner. "Distributed and communication-aware coalition formation and task assignment in multi-robot systems". In: *IEEE Access* 9 (2021), pp. 35088–35100.

[59] Branko Miloradović, Baran Çürüklü, Mikael Ekström, and Alessandro Vittorio Papadopoulos. "GMP: A genetic mission planner for heterogeneous multirobot system applications". In: *IEEE Transactions on Cybernetics* 52.10 (2021), pp. 10627–10638.

[60] Elias Mueggler, Matthias Faessler, Flavio Fontana, and Davide Scaramuzza. "Aerial-guided navigation of a ground robot among movable obstacles". In: *2014 IEEE International Symposium on Safety, Security, and Rescue Robotics (2014)*. IEEE. 2014, pp. 1–8.

[61] Christian Muise, Vaishak Belle, Paolo Felli, Sheila McIlraith, Tim Miller, Adrian R Pearce, and Liz Sonenberg. "Efficient multi-agent epistemic planning: Teaching planners about nested belief". In: *Artificial Intelligence* 302 (2022), p. 103605.

[62] Raluca Necula, Madalina Raschip, and Mihaela Breaban. "Balancing the subtours for multiple TSP approached with ACS: Clustering-based approaches vs. MinMax formulation". In: *EVOLVE-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation VI*. Springer. 2018, pp. 210–223.

[63] Ernesto Nunes and Maria Gini. "Multi-robot auctions for allocation of tasks with temporal constraints". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 29. 1. 2015.

[64] Shayegan Omidshafiei, Ali–Akbar Agha–Mohammadi, Christopher Amato, Shih–Yuan Liu, Jonathan P How, and John Vian. "Decentralized control of multi-robot partially observable Markov decision processes using belief space macro-actions". In: *The International Journal of Robotics Research* 36.2 (2017), pp. 231–258.

[65] Temel Öncan. "A survey of the generalized assignment problem and its applications". In: *INFOR: Information Systems and Operational Research* 45.3 (2007), pp. 123–141.

[66] Michael Otte, Michael J Kuhlman, and Donald Sofge. "Auctions for multi-robot task allocation in communication limited environments". In: *Autonomous Robots* 44.3 (2020), pp. 547–584.

[67] Thomas Parr, Karl Friston, and Giovanni Pezzulo. "Generative models for sequential dynamics in active inference". In: *Cognitive Neurodynamics* (2023), pp. 1–14.

[68] Corrado Pezzato, Carlos Hernández Corbato, Stefan Bonhof, and Martijn Wisse. "Active inference and behavior trees for reactive action planning and execution in robotics". In: *IEEE Transactions on Robotics* 39.2 (2023), pp. 1050–1069.

[69] Giovanni Pezzulo and Paul Cisek. "Navigating the affordance landscape: feedback control as a process model of behavior and cognition". In: *Trends in cognitive sciences* 20.6 (2016), pp. 414–424.

[70] Giovanni Pezzulo, Francesco Rigoli, and Karl J Friston. "Hierarchical active inference: a theory of motivated control". In: *Trends in cognitive sciences* 22.4 (2018), pp. 294–306.

[71] David Premack and Guy Woodruff. "Does the chimpanzee have a theory of mind?" In: *Behavioral and brain sciences* 1.4 (1978), pp. 515–526.

[72] Matteo Priorelli and Ivilin Peev Stoianov. "Flexible intentions: An active inference theory". In: *Frontiers in Computational Neuroscience* 17 (2023), p. 1128694.

[73] Ralf Regele and Paul Levi. "Cooperative multi-robot path planning by heuristic priority adjustment". In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2006, pp. 5954–5959.

[74] Rasmus Rendsvig and John Symons. "Epistemic logic". In: (2019).

[75] Matthew A Schack, John G Rogers, and Neil T Dantam. "The Sound of Silence: Exploiting Information from the Lack of Communication". In: *IEEE Robotics and Automation Letters* (2024).

[76] Ryan Smith, Karl J Friston, and Christopher J Whyte. "A step-by-step tutorial on active inference and its application to empirical data". In: *Journal of mathematical psychology* 107 (2022), p. 102632.

[77] Stephen B Stancliff, John Dolan, and Ashitey Trebi-Ollennu. "Planning to fail—Reliability needs to be considered a priori in multirobot task allocation". In: *2009 IEEE International Conference on Systems, Man and Cybernetics*. IEEE. 2009, pp. 2362–2367.

[78] Kartik Talamadupula, Gordon Briggs, Tathagata Chakraborti, Matthias Scheutz, and Subbarao Kambhampati. "Coordination in human-robot teams using mental modeling and plan recognition". In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2014, pp. 2957–2962.

[79] Mohamed S Talamali, Arindam Saha, James AR Marshall, and Andreagiovanni Reina. "When less is more: Robot swarms adapt better to changes with constrained communication". In: *Science Robotics* 6.56 (2021), eabf1416.

[80] Zheng Tian, Shihao Zou, Ian Davies, Tim Warr, Lisheng Wu, Haitham Bou Ammar, and Jun Wang. "Learning to communicate implicitly by actions". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 05. 2020, pp. 7261–7268.

[81] Annalisa Valle, Davide Massaro, Ilaria Castelli, and Antonella Marchetti. "Theory of mind development in adolescence and early adulthood: The growing complexity of recursive thinking ability". In: *Europe's journal of psychology* 11.1 (2015), p. 112.

[82] Johan Van Benthem. "Games in Dynamic-Epistemic Logic". In: *Bulletin of Economic Research* 53.4 (2001), pp. 219–248.

[83] Hans Van Ditmarsch, Wiebe van Der Hoek, and Barteld Kooi. *Dynamic epistemic logic*. Vol. 337. Springer Science & Business Media, 2007.

[84] Mathijs M de Weerdt, Yingqian Zhang, and Tomas Klos. "Multiagent task allocation in social networks". In: *Autonomous Agents and Multi-Agent Systems* 25 (2012), pp. 46–86.

[85] Briana Lowe Wellman, Shameka Dawson, Julian de Hoog, and Monica Anderson. "Using rendezvous to overcome communication limitations in multirobot exploration". In: *2011 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE. 2011, pp. 2401–2406.

[86] L Darrell Whitley, Timothy Starkweather, and D'Ann Fuquay. "Scheduling problems and traveling salesmen: The genetic edge recombination operator". In: *ICGA*. Vol. 89. 1989, pp. 133–40.

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

[88] Evşen Yanmaz. "Dynamic Relay Selection and Positioning for Cooperative UAV Networks". In: *IEEE Networking Letters* 3.3 (2021), pp. 114–118.

[89] Logan Yliniemi, Adrian K Agogino, and Kagan Tumer. "Multirobot coordination for space exploration". In: *AI Magazine* 35.4 (2014), pp. 61–74.

[90] Yuening Zhang and Brian Williams. "Adaptation and Communication in Human-Robot Teaming to Handle Discrepancies in Agents' Beliefs about Plans". In: *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 33. 1. 2023, pp. 462–471.