AN ABSTRACT OF THE THESIS OF

Yawei Zhang for the degree of Master of Science in Robotics presented on September 23, 2015.

Title: Multi-Robot Coordination: Applications in Orchard Bin Management and Informative Path Planning

Abstract approved: ____

Geoff Hollinger

Efficient coordination is desired for multi-robot systems in many scenarios. In this research, we first provide a multi-robot system to help human workers during tree fruit harvest. We present an auction-based method to coordinate a team of self-propelled bin carriers to retrieve fruit bins. Second, we propose a more general information gathering problem in a dynamic environment. In this problem, locations of points of interest change over time. Further, the amount of meaningful information or reward that can be obtained from each point is limited. We propose to use a distributed sampling algorithm for task allocation, and a receding horizon strategy for path planning in this problem. To evaluate its performance, the proposed algorithm is compared to a baseline algorithm that implements sequential auction for task allocation with greedy path planning. Experimental results suggest that the proposed algorithm is more suitable for solving the aforementioned information gathering problem. Finally we present an effective approach to coordinating a team of UAVs (unmanned aerial vehicles) to simultaneous explore, map, and search in unknown environments. The UAVs can perform a weighted trade off between the three sub-tasks. Moreover, human operators can limit the time allowed for each UAV to remain without a valid communication link to the control base station. We compare results to a market-based baseline algorithm. Results suggest that our relay system improves the efficiency of exploring, mapping, and searching tasks.

Multi-Robot Coordination: Applications in Orchard Bin Management and Informative Path Planning

by

Yawei Zhang

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented September 23, 2015
Commencement June 2016

Master of Science thesis of Yawei Zhang presented on September 23, 2015.

APPROVED:

Major Professor, representing Robotics

Head of the School of Mechanical, Industrial and Manufacturing Engineering

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Yawei Zhang, Author

ACKNOWLEDGEMENTS

The author would like to acknowledge everyone involved in this research: my advisor Dr. Geoff Hollinger, who provided the opportunity to participate in the Bin Management and UAV project, Dr. Matt Taylor, Dr. Qin Zhang, and my friend Teresa Tjahja who helped with this research.

This research has been funded in part by USDA/NIFA grant number 2014-67021-22174, and Department of the Air Force award number FA8651-14-c-0135.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

Algorithm | Page

Chapter 1: Introduction

Recently, multi-robot systems have attracted increasing attention by providing more robustness and efficiency than single-robot systems. Robots can achieve goals more effectively when they are organized as a team, thus achieving better overall performance. In many scenarios, multiple robots provide the capability to fulfill some difficult tasks that a single robot system cannot achieve. Based on these advantages, many multi-robot systems have been designed to help people in different scenarios, such as environmental monitoring [1], warehouse management[2], and search-and-rescue [3].

On the other hand, multi-robot systems usually require more resources than single-robot systems. This necessitates efficient coordination between robots to accomplish tasks well. In many scenarios multi-robot system designers also need to consider interactions between robots and humans, in addition to environment uncertainty. This is difficult since the complexity usually increases exponentially with the expanding size of the robot team (i.e., coordination problems are NP-hard). Most multi-robot problems are challenging to solve due to complex environments, large action spaces, and uncertainty of the future environment. To provide efficient coordination, we apply different methods to reduce the difficulty of finding a solution. Fist, we reduce the environment complexity by decomposing the global goal to sub-tasks. Then we apply underlying representations to assign each task to one unique robot, which reduces the robots' action state space. Finally, robots estimate a finite horizon of future actions to make decisions that fulfill the sub-tasks and reduce uncertainty about the future environment. In this work, we utilize these methods to provide efficient coordination in three different domains. Specifically, the main contributions of our work are:

- We first provide an effective bin management system that coordinates a team of robotic self-propelled fruit bin carriers to retrieve and deliver fruit bins in tree fruit harvest.
- We then discuss a more general information collection problem and present an algorithm that balances workload while optimizing information gathered.
- Finally, we derive a multi-robot coordination system that allows a team of mini UAVs (unmanned aerial vehicles) to explore, map, and search in unknown environments. Moreover, we allow the operator (human controller) to limit the time each UAV can remain without a valid communication link to a base station.

Chapter 2 provides a literature review of general multi-robot coordination in different scenarios. Chapter 3 presents a field multi-robot application in tree fruit harvest: intelligent bin-managing. Then Chapter 4 discusses a informative gathering problem as a generalized version of the in-orchard bin-managing. Chapter 5 proposes a related coordination solution for UAV exploration, mapping, and searching in unknown environments. We conclude in Chapter 6 and provide our final discussion.

Chapter 2: Literature Review

Multi-robot coordination has been studied since 1980s [4], when some researchers started work on distributed robotics with multiple mobile robots [5][6]. The study of this field has grown dramatically afterwards. Many researchers have been focusing on building efficient coordination for multi-robot systems in different scenarios. Some famous practical projects include the RoboCup competition [7] and Amazon's warehouse management system (Kiva Systems) [8].

Coordination can be achieved by decision making, which also is the focus of this research. Researchers have developed many decision making methods for multi-robot systems. In general, those methods can be separated based on two main design principles: (1) decentralized approaches and (2) centralized approaches. Both kinds of approaches have advantages and disadvantages. Decentralized approaches allow each robot of the team to examine its own sensed information and makes decisions based on it. Instead of making decisions on their own, centralized planners make global assignments depending on all the robots' current situation. Centralized planners usually perform better than decentralized approaches in terms of solution quality, since they have a better understanding of the overall environment to make decisions. Thus, the optimal solution can be potentially produced. However, centralized planners require more time to generate solutions, leading to poor scalability with the robot team size. Some centralized real-world applications need to be run offline due to expensive computation [9][10] . Centralized approaches can be used for path planning [11][12], exploration [13][14], and transportation [15][16]. In contrast, decentralized approaches give more freedom to the system. Unlike centralized approaches, each robot only needs to consider its own information, which allows robots to make decisions more quickly. Therefore, decentralized approaches can be used on many practical applications [17][18]. In this work, we utilize a centralized algorithm (Chapter 4) to coordinate multiple robots to gather information in known environments. In unknown environments, a decentralized approach is applied to avoid expensive computation.

Different communication models provide different coordination challenges. Specifically, communication can be categorized as explicit communication or implicit communication. Explicit communication allows robots to exchange information directly with other robots, while implicit communication model refers to indirect information swap. As an example of explicit coordination, Mostofi designs a cooperative network to drive robots to map obstacles [19]. Wang [20] proposes an ad hoc wireless communication frame that allows interaction in a large robot group. Gil et al. [21] provide an adaptive communication scheme to maintain the communication of a team of robots. However, when the size of the robot team reaches hundreds or more, explicit communication can be very expensive, and thus intractable. In this case, robots can interact through the environment. A lot of work in this domain is inspired by biological studies of insects, such as swarm robotics [22][23]. In this thesis, explicit communication is applied, since the goals can be achieved by a finite number of robots with knowledge of their teammates.

In this work, we propose efficient multi-robot systems for both indoor and outdoor domains. We employ an auction-based approach for in-orchard bin management (Chapter 3), a centralized approach for distributed information gathering (Chapter 4), and a decentralized method to enable multi-robot exploration in unknown environments with limited communication (Chapter 5).

Chapter 3: Intelligent In-Orchard Bin-Managing System for Tree Fruit Production

3.1 Introduction

Effective and efficient production is becoming a greater challenge to the tree fruit industry in the United States due to its heavy use of labor during harvest season. Harvest is a labor-intensive process, and labor shortages threaten the viability of the tree fruit industry in the long term. To overcome this problem and maintain the industry's competitiveness in the global marketplace, technological innovation is required.

A traditional tree fruit harvest consists of four steps: in-orchard bin-positioning, fruit picking, bin-filling, and bin-transporting. In the first step, human workers use a forklift-like machine to place empty bins at different locations within the orchard. Next, the workers pick fruits from the tree and unload them to the nearest bin. Once a bin is full, it is transported to a collecting point (repository) by a forklift. Among those four steps, bin-positioning and fruit picking are the most labor-intensive tasks with high associated costs. Our (unpublished) preliminary study shows that human workers typically spend less than half of their time picking fruit. The rest of their time is used to move, reset, climb ladders, and walk to and from bins. Therefore, there is a critical need to reduce labor cost by utilizing

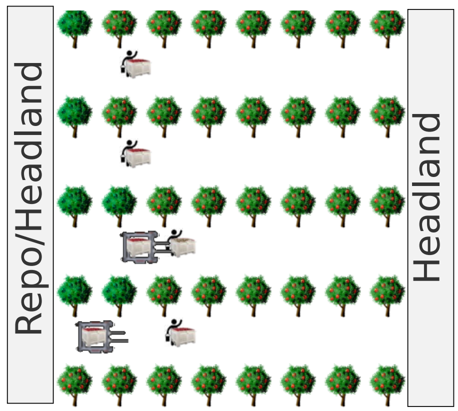


Figure 3.1: Layout of the orchard environment in the simulation.

autonomous robots to assist workers during these processes.

Current robotics technology is capable of meeting this need by providing a robotic self-propelled fruit bin carrier system for transporting bins [24]. Our work aims to develop a multi-robot system to assist human workers in placing and moving bins in the orchard to allow for efficient harvest

3.2 Simulation Setup

To test a number of multi-robot coordination approaches, we create a simulation of the typical orchard conditions in the real world. The remainder of this section provides the detailed parameters and variables of the simulator.

Fig. 3.1 shows the layout of the simulation environment as a 10×5 grid world. The leftmost and rightmost column are headland. The repository (i.e., a collecting point) is located in the leftmost column. Robots can move horizontally within the harvest area and are only allowed to move vertically in the leftmost and rightmost columns.

According to our conversations with orchard managers, they are working to make the distribution of fruit uniform. In the simulation, we assume that the fruit are uniformly distributed, with enough fruits in one grid to fill two bins. Usually, 5 pickers work together in a group to fill one bin. In our simulation, the number of groups is set to 4 and 8 with 5 pickers in each group. Thus, there are 20 and 40 pickers in total. In the real world, 1 to 5 pickers usually work together to fill one bin. Since each picker can fill one bin in one hour, a bin can be filled in roughly 12 minutes if 5 pickers work together. A picker can usually fill 8 to 10 bins per day. To simplify the problem in our simulation, one group consisting of 5 pickers can fill one bin in two time steps. Since one of the goals of this project is to have a scalable approach, we use 2 to 20 robots in our simulation.

The robot's speed depends on whether it is carrying a full bin. A bin-carrier with a full bin moves more slowly than one without a workload. In our simulation, a robot moves 2 grids per each time step while not carrying a full bin and 1 grid otherwise. At this early stage of research, all robots are assumed to have unlimited communication range within the environment.

3.3 Problem Statement

Groups of simulated workers are initialized randomly within the harvest area in the orchard. Bins are placed at locations where there are workers. All robots start from the top-left grid. Once the simulation starts, workers begin filling the bin at their location. When the bin is full, the workers request a new bin, which will be delivered by a robot. If there is no more fruit in a location, the workers move to a new location, sending a new bin request for that location. Workers cannot harvest fruit if there is no bin at their location, and they must instead wait for a bin to be delivered.

A robot in our system must make two decisions: (1) which (potentially) full bin should be picked up (and returned to the repository), and (2) where should it carry a new bin (a robot can only take a new bin if it is at the repository in the leftmost column). Once it makes a decision, it proceeds to complete its task and only makes another decision when it has finished its current task.

3.4 Algorithm Design

Our approach makes robots coordinate with each other using auctions based on the market framework [25]. Before discussing the detailed algorithm, we first explain a baseline algorithm.

3.4.1 Baseline Algorithm

The baseline algorithm is designed to use a greedy approach. An idle robot selects the closest full bin from its current location to pick up. If there are no full bins in the orchard, the robot estimates the closest bin that will be filled soonest based on the number of workers at the bin's location. Once it selects a bin, the other robots cannot choose to pick up the bin, even if their distance to the bin is closer than the first robot. Selecting a new location is also done greedily: an idle robot selects the earliest (unfulfilled) request made. A robot only considers taking a new bin to a requested location if it does not see any bin that can be picked up because all bins in the harvest area are, or will be, taken by the other robots.

3.4.2 Autonomous Robots with Auction for Coordination

Auction-based methods have been well studied and applied in many different multi-robot domains for resource and task allocation. Auction-based approaches are well suited to our scenario because they are scalable with limited computation requirements. Among different kinds of auctions (e.g. English, Dutch, etc.), we employ English auctions [26] as our task allocation method due to its simplicity and effectiveness. In an English auction, every robot 'bids' for a task. The price of the task gets increased until no robot bids a higher price. The one with the highest 'bid' wins the task. In our case, the robots bid their cost to fulfill the task. Instead of selecting the highest price, the one with the least cost wins the task.

One potential improvement to the greedy baseline approach is in the bin selection process. Instead of a single robot greedily selecting the closest (potentially) full bin, we want the robots to perform some coordination to achieve a more efficient solution. In our current implementation this coordination is done using one-turn English auction.

An idle robot scans the orchard environment to identify possible bins to pick up, which will be subsequently called *idle bins*. A bin is considered idle if there is no robot that is on the way to pick it up and it is not being carried by another robot. The robot then makes a list of plans to pick up one of the idle bins and assign a cost to each plan. The cost C to pick up the bin in plan p is formally defined as:

$$C_p = T_t + T_w$$

where T_t is the time required to reach the target bin and T_w is the estimated time of the robot to wait for the target bin to be filled. Each idle robot then sorts its plans based on the associated cost, with the plan with the lowest cost being the most preferable.

Once all idle robots construct their plans, they broadcast their plans to each other. If their most preferred plans conflict, the robot with the lowest plan cost wins the auction. The other robots then remove the auctioned plan from their list and start broadcasting their next preferred plans. This process is repeated until all idle robots have a plan to execute. However, if a robot cannot win any plan, then it will make an idle plan to not pick up any bin.

After the plan selection process, a robot also considers carrying a new bin to a requested location. If the robot has a target bin to pick up, it estimates whether there will still be fruit at the target bin's location. If so, the robot takes a new bin to that location. Otherwise, it has to make another decision to choose a requested location to deliver a new bin to. In the case where a robot does not have a target bin, it chooses to carry a new bin to the closest requested location.

A straightforward way to select a requested location is to calculate the time required to travel between the target bin and the requested location. A seemingly efficient method is to choose the closest requested location to the target bin.

Alg. 1 provides our proposed algorithm.

Algorithm 1 Auction-Based Coordination

Require: Robots set R
Ensure: Robot plans

while Any robot r_i has no plan **do**
  for $r_i \in R$ **do**
    MakePlan(r_i)
  for $r_i \in R$ **do**
    $conflictPlans \leftarrow CheckPlanConflict(r_i, R)$
    if conflictPlans not $\varnothing$ **then**
      Auction(conflictPlans)

3.5 Experimental Results

We evaluate our algorithm with different simulation setups to observe its scalability. We observe its performance in a 10×10 grid world with 4 groups of workers. With this setup, fruit in the orchard will fill a maximum of 160 bins. As can be seen in Fig. 3.2, our proposed algorithm produces better results (i.e., fewer number of steps) compared to the baseline.

Fig. 3.3 shows the number of full bins returned to the repository within a 150 step limit. The maximum number of bins is 160 — systems with large numbers of robots finished harvesting in fewer number of steps. As can be seen from the figure, our proposed algorithm consistently outperforms the baseline algorithm.

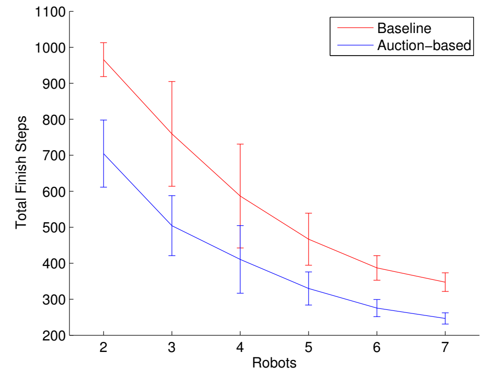


Figure 3.2: The number of steps required by different numbers of robots to retrieve all bins(160) to the repository.

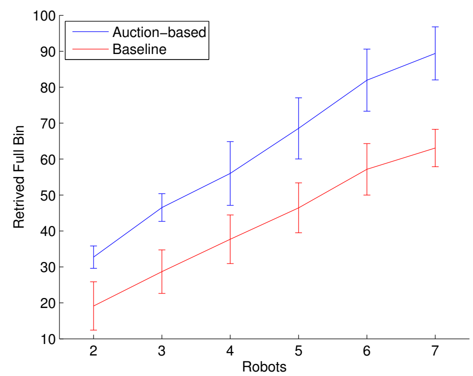


Figure 3.3: The number of full bins returned to the repository by different algorithms in 150 step limits.

Fig. 3.4 shows the scalability of both algorithms. We examine a orchard with 5 lanes and 10 trees per lane (80 bins totally). The results shows that when the size of robot team scales to 20 robots, the auction-based approach still out perform the baseline approach.

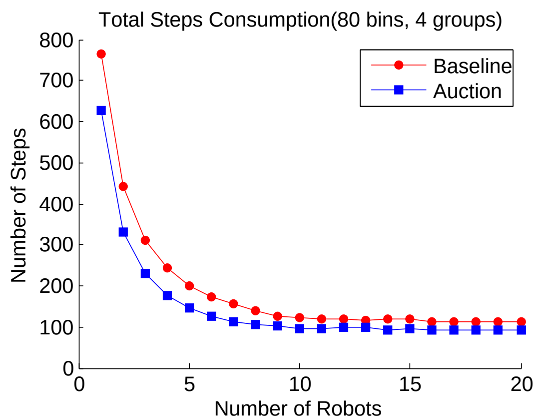


Figure 3.4: The number of steps required by 2-20 robots to retrieve all bins(160) to the repository.

3.6 Discussion

In this work, we proposed a scalable coordination between autonomous robots, which is implemented for fruit bin-carriers to increase the effectiveness of fruit harvesting, specifically to assist human workers to transport fruit bins within the orchard. To achieve a more optimal system result, robots use auction-based approach to coordinate their decisions to pick up bins in the orchard.

In the auction-based coordination, it is also possible to consider not only one step, but two or more steps ahead. For instance, each robot considers picking up multiple bins in each plan instead of only one bin as implemented in our current system. We briefly explored this possibility in this project, but the results were not yet satisfactory. We observed that the only the first bin matters in the plan cost calculation. The bins after the first will have been picked up by some other robots. Thus, including them in the cost calculation only introduces noise. This happens especially when the numbers of robots and idle bins (which is related to the number of groups of workers) are relatively close. To work around this problem, one idea is to predict whether the next bins would still be at their current locations after the robot finishes returning the first bin to the repository. For example, the costs of picking up the next bins are weighted by the probability of the bins remaining at their locations. This scenario will allow more complex coordination between robots, thus improving the overall system performance.

In addition to the improvements that can be done on our current algorithm, there are also potential actions that can be introduced to the robots, such as exchanging plans in the middle of tasks execution (even when the robots are not idle), bins swapping between robots in the orchard, and moving bins around depending on the remaining fruits. Given these more sophisticated actions, system efficiency can be improved more significantly.

Chapter 4: Multi-robot Routing for Dynamic Information Gathering

4.1 Introduction

The bin-managing system discussed in the previous chapter is closely related to more general robotic information gathering problems. In a typical information gathering problem, a number of pre-specified waypoints are located in different places. Visiting a waypoint gives rewards, which may be different across different waypoints. Thus, some waypoints have higher priorities than the others. In a multi-robot system, several robots are required to visit the waypoints to retrieve rewards. The overall system goal is to obtain the maximum reward within a certain period. Often, each robot has a certain budget that limits its travel distance or the amount of reward it can carry at any given time.

A multi-robot system for information gathering then has two key components: task distribution and multi-robot routing. Some of the existing methods work with pre-determined, static information depositories (e.g., stationary sensors). Such information gathering problems can then be formulated as a Traveling Salesman Problem (TSP) [27], specifically a variant of it called the Prize-Collecting Traveling Salesman Problem (PC-TSP) [28]. However, in many scenarios, locations of the waypoints and the amount of reward that can be obtained from each waypoint can change. For example, a sensor at a waypoint can only store a certain amount of reward in its depository. Once the depository is full, the sensor has to stop working, in which case the rewards that it could have gathered after its depository is full can be seen as potentially lost. In another case, the amount of meaningful information that can be obtained from one location can decrease over time. This will cause a sensor to move to a new location to get newer information. Thus, the locations of waypoints are no longer static. Previous approaches for persistent monitoring, which consider waypoints to be static, are not suitable for this type of problem. Novel and effective task distribution and routing methods are thus required.

In this work, we define a novel information gathering problem with dynamic waypoints. We propose to use distributed sampling algorithm for task distribution with path planning based on Receding Horizon strategy to solve the problem. Our proposed method will be compared to a baseline algorithm that performs sequential auction to distribute tasks and uses greedy approach to path planning.

4.2 Background

As a key part of multi-robot coordination, task distribution problem has received considerable focus from existing research. Among them, market-based approaches are popular, since many problems can be formulated as a virtual economy. In a market-based framework, tasks are typically allocated through auctions. For a specific task, each robot computes the costs for the a given task and the reward of finishing the task, and makes a bid based on them. Then, it broadcasts its cost and reward. The robot with the best bid wins the auction and inserts that task to its to-do list. In [29], Gerkey et al. propose an auction-based system to allocate the tasks for physical multi-robot coordination. Kalra et al. [25] describe a market framework named Hoplites to solve tightly-coupled coordination problems. However, these works do not focus on how to evenly distribute the workload for the robots. In [30], Goldberg et al. allows robots to sell some of their current tasks to avoid overloading but the robots are assumed to explore an unknown environment (e.g., simulated Mars). In our scenario, a centralized scheduling algorithm is feasible since the environment is known. Tovey et al. [31] propose a Sequential Single-Item (SSI) method that allocates one unassigned task to one robot so that the total team cost increases minimally. This is similar to our approach, but the order of assigned tasks is domain specific. In our approach, we modify their method and derive our sequential auction approach as our baseline algorithm.

In this work, because a robot can visit a series of goals to retrieve information, an efficient trajectory needs to be developed to minimize the total system cost. This problem can be viewed as an Informative Path Planning (IPP) problem. In [32], Binney and Sukhatme propose a branch and bound approach for a single robot to find the optimal path from start to a goal on an informative map. Yu et al. [33] formulate the problem as a Mixed Integer Quadratic Problem (MIQP) and solve it by Gurobi [34]. Most of these works deal with stationary graphs, where the problem can be modeled as a Traveling Salesman Problem (TSP). A very similar problem is K-Traveling Repairman Problem (KTR) [35], where K traveling repairman attempt to cover a set of goals with minimum global latency. The differences of our problem from KTR are: (1) the visiting requirement is continuous (depositories keep collecting data as long as they are not saturated and there is uncollected data); and (2) the object of our system is to minimize the saturated time of the depositories. In our case, since the rewards in each depositories are increasing with time, different decisions at a step generate different states at the next step. This means that the graph changes dynamically during the robot's journey. Therefore, most TSP and KTR algorithms cannot be directly applied to this problem. To overcome this difficulty and to avoid the computationally-expensive brute-force search, we use a Receding Horizon (RH) strategy. Previous work has shown that a receding horizon path planner is effective at optimizing paths in a smooth environment [36]. Hollinger and Singh [37] describe an approach for multiple agents searching for a target in a known environment using receding horizon. Tisdale et al. [38] describe a receding horizon path planner for multiple unmanned aerial vehicles to search for a stationary object. In our case, RH can be a reasonable strategy to find a feasible solution.

4.3 Problem Statement

Our information gathering problem can be defined as follows. Given an environment $\mathcal{W}$ where n waypoints with sensors and m robots exist, we want to maximize the amount of reward collected while minimizing the idle time of the sensors. In each waypoint, a sensor with arbitrary fill rate f gathers reward from its location and stores it in a depository with capacity $\mathcal{C}$. When the depository is full, the sensor stops working and waits for some robots to retrieve some reward from its depository. Also, the amount of reward that can be obtained from each location in the environment is limited to $\mathcal{R}$. When a sensor has gathered all reward at its location, it randomly moves to another location. Each robot also has its own depository. The job of each robot is then to visit the waypoints (i.e., sensors), retrieve some or all of the reward in the visited depository, and returns to its base station to unload its own depository. The total collected reward is then calculated based on the amount of reward returned to the base stations.

Given the above setup, we want to develop a framework for multi-robot coordination. Specifically, we want to solve two problems: task distribution and path planning. Our goal is to maximize the collected reward while minimizing the potential reward loss, which is the reward that could have been obtained while sensors are idle. The problem can be posed as the following optimization problem:

$$\arg\min_{p \in \Psi} \sum_{i=1}^{n} I_i f_i \tag{4.1}$$

where I_i is the total idle time of sensor s_i, $i = 1, \ldots, n$, f_i is s_i's fill rate, p the set of optimal paths of robots, and Ψ is the set of all possible paths.

Given n sensors and m robots, each robot then has to decide the sensors to visit and the amount of reward to be collected from each depository. We formulate the problem with a graph $G = (V, E)$, where a vertex $v_i \in V$ represents an existing bin in the orchard and an edge $e_{ij} \in E$ is the edge between v_i and v_j. The weight

w_i of vertex v_i is determined by

$$w_i = -t_i + f_i \tag{4.2}$$

where t_i is the estimated time to fill the depository. After visiting all target locations, a robot returns to its base station. Given a team of robots, two components in this problem need to be addressed: task distribution and path planning.

4.4 Proposed Algorithm

4.4.1 Task Distribution using Distributed Sampling Algorithm

In order to evenly distribute the workload to each robot, a partition should be formed on the graph G. However, given m robots, the problem of splitting the graph into m partitions is an NP-hard problem [39]. Kim and Shell [40] adjusted the heuristic in [41] to distribute the workload in robotics environmental monitoring problem with time complexity of $O(m^3)$. Here, we apply their Distributed Sampling Algorithm (DSA) to solve task distribution problem.

Consider a two-robot case. Given two robots a_i and a_j, in order to split the graph into two evenly partitioned sub-graphs G_i and G_j, the problem can be formulated as:

$$\arg \min_{(G_i, G_j)} [Cost(G_i) - Cost(G_j)] \tag{4.3}$$

where $Cost(G_i)$, $G_i = (V_i, E_i)$ is calculated as

$$Cost(G_i) = \sum_{k=1}^{|V_i|} w_k + \sum_{e_{ab} \in E_i} |e_{ab}| \tag{4.4}$$

with w_k is the weight of vertex v_k that is computed using Eq. 4.2, and $|e_{ab}|$ is the length of edge e_{ab} (the distance between v_a and v_b, $v_a, v_b \in V_i$). Our goal is to minimize the difference between sub-graphs G_i and G_j. The pseudocode of DSA is provided in Alg. 2.

Algorithm 2 Distributed Sampling

Require: Subgraphs $G_1, \ldots, G_m$ for m robots
Ensure: Reconstructed subgraphs $G'_1, \ldots, G'_m$

```
for i ← 1, ..., m do
    R_i ← Cost(G_i)
Let G_i be the subgraph with the maximum cost R_i
Let G_j be the subgraph with the minimum cost R_j
S_ij ← {(v_i, v_j) | v_i ∈ G_i, v_j ∈ G_j}
costTemp ← ∞
for i ← 1, ..., |S_ij| do
    R'_i ← Cost(G_i − {v_i})
    R'_j ← Cost(G_j + {v_i})
    if R_i − R_j > R'_i − R'_j and R'_i − R'_j < costTemp then
        a ← i
        costTemp ← R'_i − R'_j
G'_i ← G_i − {v_a}
G'_j ← G_j + {v_a}
```

4.4.2 Path Planning based on a Receding Horizon Strategy

After each robot has been assigned a set of waypoints to visit, it has to determine the order of waypoints to be visited and the amount of reward to be collected from each waypoint. Since one of our goals is to minimize the potential reward loss that occurs when a sensor is idle, we want to prioritize visiting the sensor with the earliest time to fill its depository. Thus, we need to assign a score to each waypoint that reflects its *urgency*. When deciding the urgency of a particular waypoint, we need to consider the sensor's fill rate, the contents of its depository, and the distance from the robot's location. At a glance, this problem is straightforward: once the cost to visit each waypoint has been calculated, a complete and optimal shortest-path algorithm can be used. However, in our case, visiting a waypoint changes the urgency score of the other waypoints, since the contents and location of waypoints can change while the robot is on the way to visit a waypoint, i.e. our graph is dynamic. An optimal solution can be computed by exploring all possible permutations (since the order matters) of the assigned waypoints, which is NP-hard. Thus, to determine the order of visit, we use Receding Horizon (RH) strategy.

Our RH-based algorithm is based on the urgency of each waypoint instead of its cost (e.g., distance). The urgency is calculated as follows

$$u_i = \frac{w_i}{d_i} \tag{4.5}$$

where w_i is defined in Eq. 4.2 and d_i is the distance to reach v_i. If v_i is the first waypoint to visit, then d_i is the distance from the robot's current location to v_i. Otherwise, d_i is the distance from the previously visited waypoint to v_i. The robot then greedily chooses the most urgent waypoint to visit with three-step look ahead (the number of layers in the horizon is 3).

Once the order of waypoints is determined, the next step is to decide the amount of reward taken from each waypoint. We cannot simply take all reward, since the robot can only carry a limited amount of reward. Thus, we want to take more from waypoints with faster-working sensors and less from slower-working ones. The reward to collect from each waypoint should then be proportional to the estimated total reward that the sensor would gather during the time the robot needs to complete its tour (i.e., visit all waypoints and return to base station). Let T be the time required by the robot to complete its tour. The estimated amount of reward in sensor s_i's depository is

$$\delta_i = c_i + T f_i \tag{4.6}$$

where c_i is the amount of reward in s_i's depository at the current time step and f_i is the fill rate of s_i. Then, the reward collected from the waypoint is calculated as

$$\alpha_i = \frac{\delta_i}{\sum_{j=1}^{|V_r|} \delta_j} \mathcal{C}_r \tag{4.7}$$

Here, $G_r = (V_r, E_r)$ is the subgraph assigned to the robot and $\mathcal{C}_r$ is the maximum amount of reward that can be carried by the robot.

4.4.3 Baseline: Sequential Auction with Greedy Path Planning

In order to evaluate our task distributed sampling algorithm, we introduce a baseline algorithm with a market-based approach. Two main sub-problems need to be solved: (1) assigning waypoints to the robots such that the global latency is minimized, and (2) planning a path for each robot and determining the amount of reward collected from each waypoint such that the sensor's idle time is minimized.

A market-based approach is applicable to this scenario. Here, we consider visiting a waypoint as a task. Task assignment is then done with sequential auctions. Each robot bids for the auctioned task, and the robot that wins the bid adds the task to its plan. Path planning can be settled simultaneously with one-horizon greedy approach, where a robot visits first the most *urgent* waypoint in their plans. The auction are held as follows. Given n waypoints, we first sort the waypoints based on urgency. In this approach, the urgency is slightly different from the one used in distributed sampling algorithm. Since the tasks are not assigned yet, the urgency is simply equal to the time to fill the depository in the waypoint. Once the tasks are sorted, the auction starts with the most urgent task, which is the waypoint with the shortest time to fill. Each robot then posts its bid (i.e., cost of performing the task). If the robot has no task in its plan so far, the cost of this robot for the auctioning task is the distance from its current location to the target waypoint. If the robot has some tasks, the cost equals the the total previous cost of its plan plus the distance from its last task (waypoint) in its plan to the auctioned task. The robot with the minimum cost wins the auction and adds the task to the end of its plan. After the most urgent task is assigned, the second most urgent one is auctioned, and so on. After all of the tasks are assigned, the path of each robot will be the order of the waypoints in its plan.

After the tasks are allocated and paths are planned, the last step is to determine how much reward a robot will collect from each waypoint. This is determined before the robots starts its tour. The amount is calculated with Eq. 4.6.

When a robot finishes its plan and returns the gathered reward to the base station, another round of auction will be held again. Robots with unfinished tasks will compare the new plans to previous plans. If the rewards lost for all waypoints in the new plan is less than the rewards lost from previous plans, then the robot switches to the new plan. Otherwise, the previous plan is used. An additional auction will be held for all unassigned tasks among idle robots to ensure that no robot would be idle. This method guarantees that no robot will be idle, while maintaining the performance of the other robots. Algorithm 2 provides more details.

4.5 Evaluation

4.5.1 Experimental Setup

We evaluate our proposed algorithm in a grid-world simulation as depicted in Fig. 4.1. There are ten randomly-placed waypoints and two robots with base points on the top-left and bottom-right corners. The robots can move in eight

Algorithm 3 Sequential Auction

Require: m robots, n waypoints
Ensure: Information loss I

```
I ← 0
for i ← 1, ..., m do
    TaskAssign({v_i, ..., v_n}, {r_1, ..., r_m})
while Information map ≠ empty do
    if there is no idle robot then
        for i ← 1, ..., n do
            Move(r_i)
    if there is an idle robot then
        TaskAssign({v_i, ..., v_n})
        for i ← 1, ..., m do
            if InfoLoss(p_old) ≥ InfoLoss(p_new) then
                TaskAssign(unassigned v, idle robots r)
    I ← I+ IdleTime(v_1, ..., v_n)
```

directions, one step at a time. There are 200 units of reward at every grid, and each depository (either in each waypoint or each robot) can store at most 100 reward units. Each simulation is run for 1,000 time steps.

Three metrics are used to evaluate the performance of our algorithm:

- Reward gain: the total units of reward collected, i.e. returned to base stations. This quantity does not include the rewards in depositories.

- Idle time: the total number of time steps where sensors are idle. Let I_i be the total time sensor s_i is idle during the period of simulation (1,000 time steps). Then, the total idle time is calculated as

$$\tau = \sum_{i=1}^{n} I_i \tag{4.8}$$

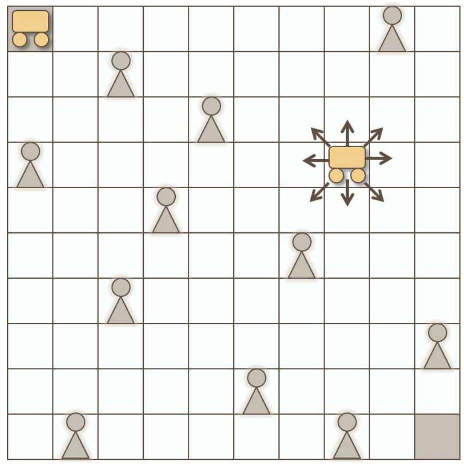


Figure 4.1: Grid world simulation of our information gathering problem.

where n is the number of waypoints.

- Reward loss: the amount of rewards that could have been collected while sensors are idle. In some cases, it is not possible to provide enough space in the depository in every waypoint, so some waypoints will be idle. However, it is more desirable for a waypoint with slower-working sensor to be idle than a waypoint with faster-working sensor. The idle time Γ provides the number of time steps wasted due to the sensors being idle. But, it does not reflect the difference between an idle sensor with faster and slower fill rate. Thus, we calculate the potential reward loss to see this difference. Formally, the reward loss is calculated as

$$\Gamma = \sum_{i=1}^{n} I_i f_i \tag{4.9}$$

where f_i is the fill rate of sensor s_i.

4.5.2 Experimental Results and Discussions

Based on the experimental setup and evaluation metrics described in Section 4.5.1, we compare the results between our proposed algorithm and the baseline. The results are shown in Fig. 4.2. As can be seen from the figure, the proposed algorithm outperforms the baseline in all three metrics. In our proposed algorithm, three-layer RH strategy is used for path planning. However, since the baseline algorithm uses only one layer, we also present the results of our proposed algorithm with one-layer RH for path planning. As can be observed from the figure, the results of using one and three layers only differ slightly, and are still significantly better than the baseline results. Thus, the proposed algorithm still outperforms the baseline even when both use one-layer RH strategy for path planning. This suggests that distributed sampling algorithm is more suitable than sequential auction for task distribution in our problem.

To see the scalability of the algorithms in terms of the number of agents, we conducted experiments with 2, 3, 4, and 5 agents. The simulation setup is identical to the one used in previous experiments. Here, we use three-layer RH strategy for path planning in the proposed algorithm, since using one layer does not produce a significant difference. The results are shown in Fig . 4.3. The proposed algorithm show monotonic improvement in reward gain, idle time, and reward loss as the number of agents increases. This is not the case for the baseline, where the results of 4- and 5-agent systems only differ slightly. Also, the proposed algorithm produces greater improvement in idle time and reward loss compared to the baseline.

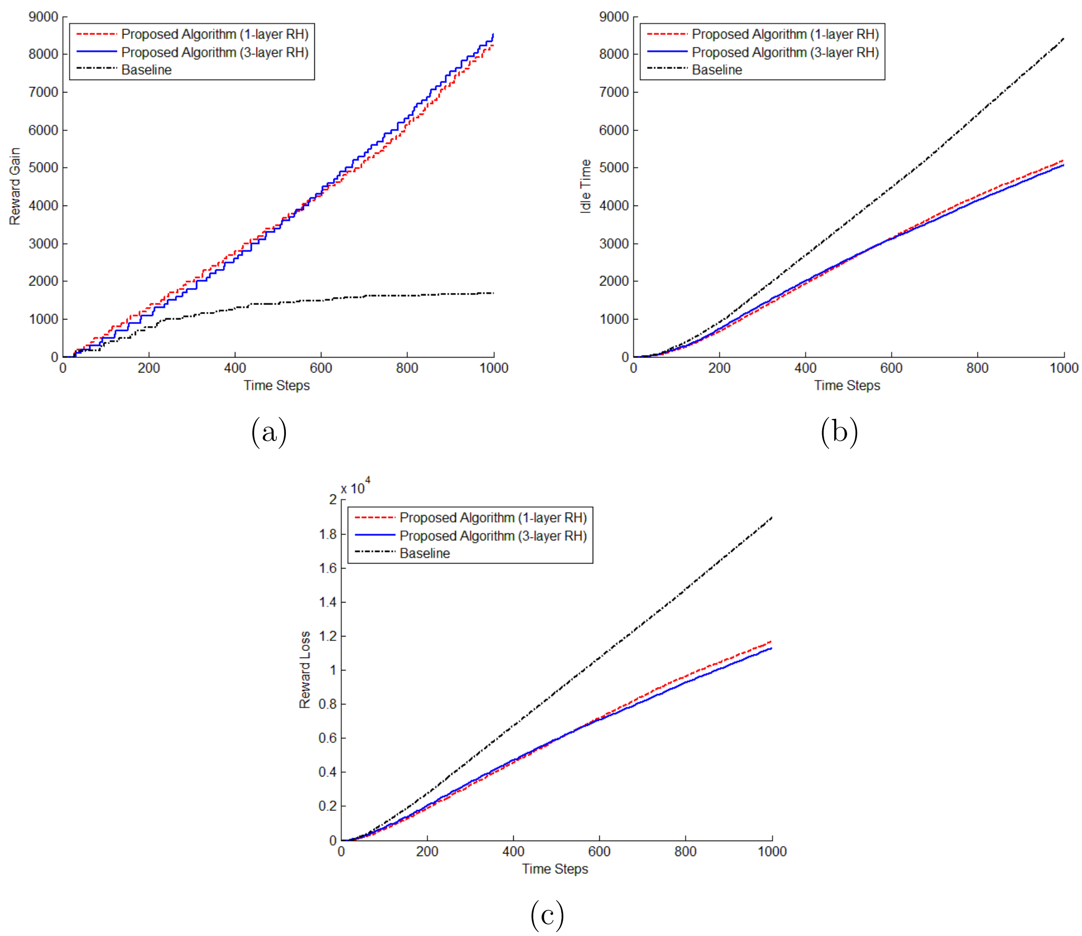


Figure 4.2: Performance comparison of proposed algorithm with one and three layers RH and the baseline over 1,000 time steps. (a) shows the total rewards collected, (b) shows the total idle time , and (c) shows the potential reward loss suffered by the algorithms. In (a), higher is better, while in (b) and (c), lower is better.

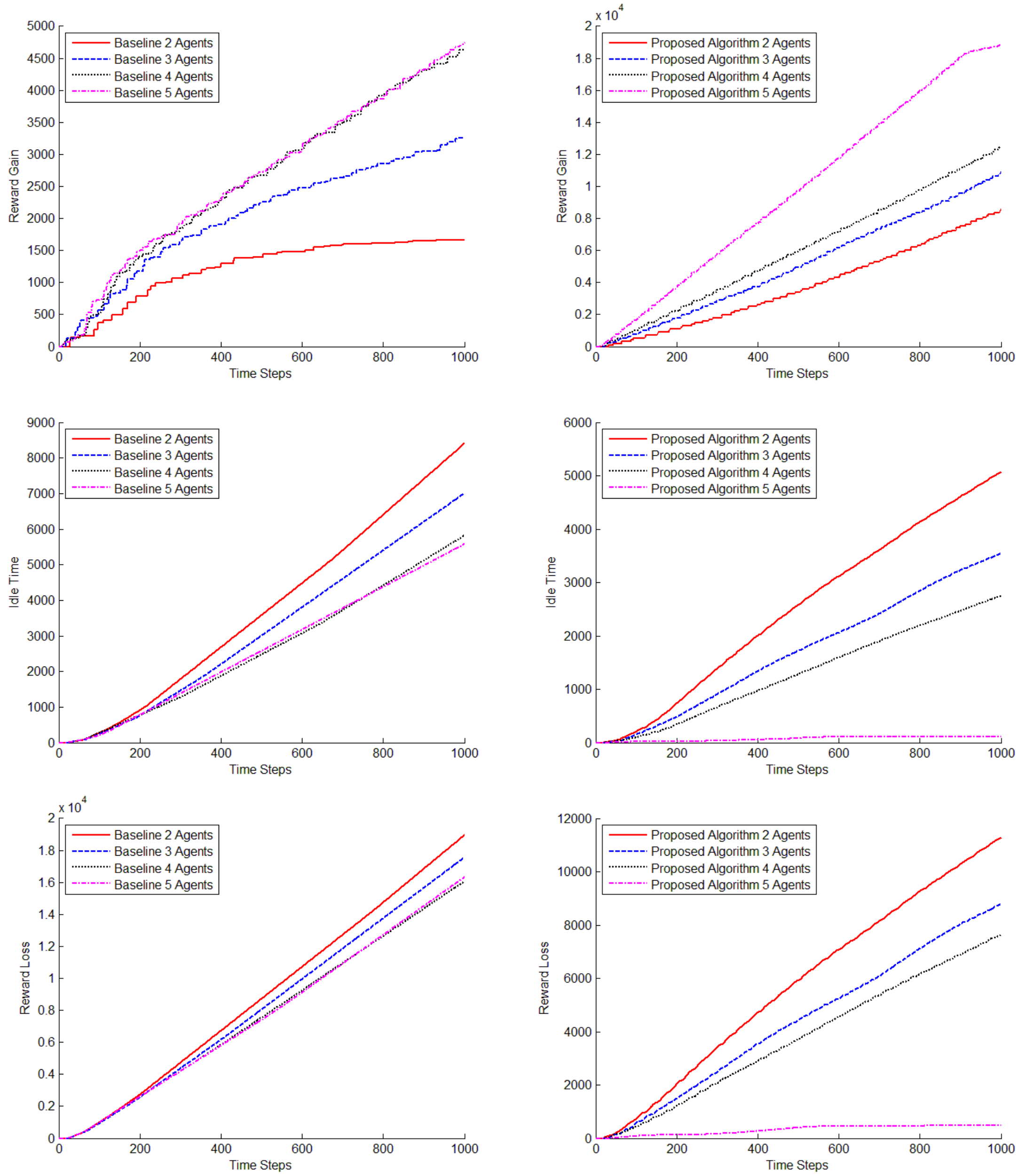


Figure 4.3: Performance comparison of the baseline and proposed algorithm over 1,000 time steps. The figures on the left show the results of the baseline algorithm, while the figures on the right show the results of the proposed algorithm.

4.6 Discussion

Traditional information gathering problems are generally focused on environments where points of interest (e.g., waypoints, sensors, etc.) are static. In such cases, the points of interest are assumed to have fixed locations and reward or information available at every point is unlimited. The sensors and robots gathering the information can also carry unlimited amount of information. In our problem, locations of interesting points can change and the amount of meaningful information that can be collected from each point is limited. The information that can be stored by sensors and robots is also limited. Thus, the problems need be solved in a dynamic environments, which exponentially increases the difficultly. We try to solve this problem with a combination of centralized and decentralized approaches. To distribute the tasks, a centralized partition approach provides near-even distribution of the tasks, while the decentralized path planning reduces the computation of finding a efficient path.

Chapter 5: Multi-UAV Exploration, Mapping, and Searching in Unknown Indoor Environments

5.1 Introduction

As shown in prior chapters, centralized task allocation performs well when the environment is known. However, in uncertain environments, centralized task allocation may be brittle to unforeseen losses of communication. A common domain of interest is indoor environment exploration, in which mobile robots have been utilized to build maps of unknown environments. In particular, unmanned aerial vehicles (UAVs) have gained popularity due to their agility and ability to cover large areas faster than ground robots[42][43][44]. However, UAVs generally require shorter mission duration due to limited battery life. Therefore, efficient planners are needed to coordinate a team of UAVs to achieve the goals.

A typical indoor exploring or mapping scenario is described as follows. Initially, a human controller sets up the assignments and releases the UAVs. UAVs then perform the planner's commands to finish the tasks. Eventually, UAVs return to the base station and retrieve the explored information.

In many domains, human controllers need to adjust the goals based on the collected information. This requires robots to communicate with the base station occasionally to exchange information. However, extra energy and time could be used to fly back to swap information, thus potentially delaying the completion of the mission. Therefore, a effective planner should efficiently relay an information among the whole team.

To solve this problem, we propose a multi-UAV planner that can fulfill exploring, mapping, and searching simultaneously in unknown indoor environments. The main contributions of this chapter are:

- A decentralized planner that coordinates a team of robots to fulfill three different goals (exploration, mapping, and searching) simultaneously. We allow human operators to set priority weights among these three subtasks.

- An effective method that allows the base station to communicate with UAVs, so operators can conveniently adjust the weights of different tasks. Moreover, the base station can limit the time allowed for each UAV to remain without a valid communication link to the base station.

5.2 Background

Muti-robot exploration, mapping, and searching have been studied under different perspectives. In order to explore an unknown environment, Burgard et al. describe a multi-robot system in [45] to minimize the exploration time in an unknown indoor environment. Mirzaei et al. provides an decentralized approach to organize a heterogeneous robot team to explore and cover a uncertain environment by a look-ahead path planner and Voronoi regions partition in [46]. Marjovi et al [47] describe a frontier-based planner that drives multiple UAVs search and track wild fires. Gan et al.[48] apply a group of UAVs to search for a target while running a gradient-based optimization algorithm. Charrow et al.[49] employ a team of robots to create a 3D map of an indoor environment using Cauchy-Schwarz quadratic mutual information. Yang et al. [50] utilize opportunistic learning method to construct an online planner for multi-UAV searching in uncertain environments.

Many works studied how to maintain connectivity between robots during the mission. Relay is a general utilized method to build the connections. Gil et al. [51] provide both exact and approximate algorithms for finding the best relay locations. Pei et al. [52] utilized the Minimum Steiner Tree problem to generate the best relay positions. However, the connection needed to be consistently maintained. We employ a similar strategy, but we allow more freedom for the UAVs to break communication connectivity.

5.3 Problem Setup

5.3.1 Environment Model

Indoor environments are often organized as many rooms (Fig. 5.1). As a topological representation, the concept of room is utilized in many different domains [53][54]. Therefore, in this work we set the environment as a typical indoor office building environment that contains a number of rooms. Each room is represented as a node in the map. Based on the sense range of the UAVs, there are r_n nodes in this building and each node belongs to a specific room type $t \in T$ (e.g., office, lab, corridor, etc.). Each node r_i has a probability p_i that it contains a static target. Initial probability is set by operator's priors. To simplify the approach, we formulate the indoor environment as a undirected graph $G = (V, E)$. Each room r_i corresponds to a node $n_i \in V$, and an edge e_{ij} connects n_i and n_j if UAVs can reach n_j from n_i directly.

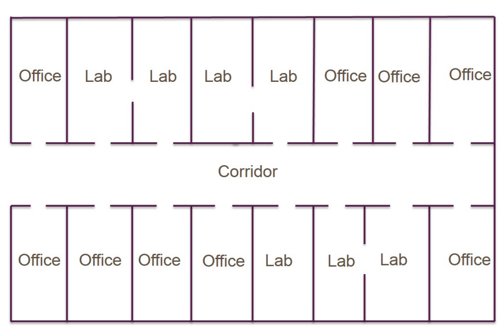


Figure 5.1: A typical school building with different types of rooms

5.3.2 UAV model

UAVs in this work start with a certain amount of budget (e.g., energy, time etc). They are equipped with sensors that allow them to sense the room type or target existence. When a UAV arrives at a node, it can take one of the three actions at each step: 1) observe the room type, which corresponds to mapping a room; 2) observe the target existence, which corresponds to searching; and 3) go to one of its neighbor node, which is exploring. Each action will return some information gain to the UAV. The UAV makes a decision based on the information reward function and the weight of each goal. Every action will cost a robot a certain budget c. Each UAV has a communication range c_r. If the distance between two UAVs is less than c_r, they can share the explored map as well as the knowledge of rooms.

5.3.3 Sensor Model

Different sensing actions return different information about the environment. For a particular room r_i, each UAV has a belief probability state that contains the probability vector $[P_{t1}, ..., P_{tn}]$, $n = |T|$. The UAV also has the probability of the target existence $[P_T, P_F]$, where T indicates *True*, F indicates *False*. However, sensor observations are noisy. Each time one UAV takes a observation, it has a small possibility to sense the incorrect information about a room. Similarly, target observation may be imperfect. The accuracy of the sensor depends on the real sensor model and the target classifier on the UAV.

5.3.4 Bayesian Update

Before the mission begins, UAVs are set up with priors on the state of the environment. The priors provide the probability that one type of the room contains a target. Note there can be more than one target in the environment. In our simulation, the types of room are: office, lab, and corridor. Each type of the room has an initial probability that it contains a target. The initial prior is shown in Tab 5.1

Table 5.1: Initial prior of target existence

Room Type	Corridor	Office	Laboratory
Probability of Target Existence	0.1	0.8	0.6

Based on the priors and the noise set, the probability of a room type after a target sense is updated as:

$$P(R^{t+1}|T) = \frac{P(T|R)}{P(T)} P(R^t) \tag{5.1}$$

where T is the set of room types and $P(R^{t+1}|T)$ is the prior at time $t+1$. Based on the same principle, the probability of target existence is updated after a sening action as follows:

$$P(T^{t+1}|R) = \frac{P(R|T)}{P(R)} P(T^t) \tag{5.2}$$

The value of $P(T^{t+1}|R)$ is now set as the new prior $P(T^{t+1})$ at time $t+1$.

5.3.5 Reward Function

UAVs select actions based on the estimated reward of each action. In our setup, the estimated reward of a specific action is equal to the weight of that action times the information gain of taking that action. The information gain of observing a room type is based on the uncertainty reduction of the node. The uncertainty is quantified by the entropy of the room type. Initially, each node has a uniform distribution over all room types. After each observation, the entropy of room r_i gets increased, which means the uncertainty of the room type is reduced. Thus the information gain function for mapping a node is defined as:

$$I_m = -\sum_{t \in T} P_t \log(P_t) \tag{5.3}$$

where T is the set of possible room types and P_t is the probability that this node's type is t. The information gain I_s of the target observation action is the same as observing the room type, except the set T is changed to the binary set $\{True, False\}$, and the P_t is changed to the probability of the target existence. The information gain of moving to a neighbor node is defined as:

$$I_e = \mathcal{I}\alpha^c (0 < \alpha < 1) \tag{5.4}$$

where $\mathcal{I}$ is a constant information gain when a node is visited for the first time. The information gain is decreased, along with increasing the visited count c of the node. The value α is the discounting rate. Based on the information gain functions, the estimated reward function of UAVs for exploring, mapping, and searching are: $\omega_e I_e, \omega_m I_m$, and $\omega_s I_s$, respectively. Note the sum of $\omega_e, \omega_m, \omega_s$ equals 1. Weights of sub-tasks are set by human controllers. The benefit of this mechanism is allowing the human controller control to influence UAV decisions based on the priors. For example, if the human controllers have no knowledge of the environment, then the weights can be set as Exploring: 1.0, Mapping 0, Searching: 0. The setting will drive the UAVs to visit every room as fast as possible. After the UAVs retrieve a rough map of the environment, the weights can be changed to Exploring 0, Mapping 1.0, Searching 0 to allow the UAVs map the world as clear as possible. If the human controller, on the other hand, want to find more targets instead of mapping the environment, the weight of searching can be set higher than other two sub-tasks.

5.3.6 Communication Loss Constraint

Each UAV has a fixed communication range c_r. As mentioned in Sec. 5.1, human controllers can set a specific time step S that one UAV can lose communication with the base station. For example, if S equals 0, that means the UAV cannot lose communication with the base station at any time. It is easy to see that if the value of S is small, the UAV cannot explore some further regions if they do not coordinate. Moreover, even if the S is big, UAVs will waste a lot of energy to travel back to the base station in order to retrieve the information. To solve this problem, we will present our solution in next section.

5.4 Algorithm Design

5.4.1 Proposed Algorithm

One objective of our system is for the human controllers at the base station to obtain the information from each UAV no more than time step S after its last communication. An existing solution to this problem is to only allow the UAVs go to the nodes that will not violate the constraint. We will describe this method as our baseline in the next section. The problem with this approach is an inefficient use of energy because UAVs are traveling back and forth to communicate with the base station. To achieve more efficient coordination, we propose a relay system that can retrieve information faster by allowing some UAVs to act as relay nodes.

Basically, there are two roles that each UAVs can perform: (1) explorers (E_u), and (2) relays (R_u). The E_us execute the tasks, and they do not need to directly communicate with the base station since there are R_us helping retrieve their information. Three problems need to be solved: (1) what is the next action of E_us; (2) if some of the E_u request relays, will these requests be satisfied; and (3) if yes, where to locate the R_u.

For the first problem, we propose a market-based solution. Initially, all UAVs are all E_u. Once one E_u requires a relay, it will broadcast its request. All UAVs who received this request will negotiate with the request robot. One UAV will examine the tradeoff between for giving up its current plan to be the relay. If a better global reward can be achieved, then the relay request will be fulfilled. Among all these new plans, a best one will be selected and the UAV will be the relay of the request relay.

To generate their plans, the exploring UAVs need to know the candidate relay locations. We formulate this problem as a *Steiner Minimum Tree Problem with Minimum Steiner Points and bounded edge length (SMT-MSP).* Intuitively, we try to find a best location for one R_u that minimized the sum of distances to its E_us and distance to its own relay (if exist) or base station. The SMT-MSP will give a good approximated location for the problem. Note that not all relays need to have a direct connection to the base station, and relays can be chained to form a connected network. SMT-MSP is proved as NP-Hard [55]. We modified the greedy method from [56] to solve the SMT-MSP.

Alg. 4 provides our algorithm.

Algorithm 4 ST-EMS

Require: UAV positions at time t, $\{p_1^t, ...p_n^t\}$ Explorer Set $\boldsymbol{E_u}$, Relay Set $\boldsymbol{R_u}$
Ensure: UAV positions at time $t+1$, $\{p_1^t, ...p_n^{t+1}\}$ Explorer Set $\boldsymbol{E_u}$, Relay Set $\boldsymbol{R_u}$

```
for e_i ∈ E_u do
    while e_i positions at time t + 1 is ∅ do
        MakePositionDecision(e_i)
        if relayRequest(e_i) then
            if HasRelay(e_i) then
                relay ← SMT-MSP(r_i)
            else
                relay ← LookForRelay(e_i, E_u, R_u)
        if relay then
            break
```

5.4.2 Baseline

As the baseline of this problem, the UAVs make decisions regardless of other UAVs' request. Each UAV will follow three steps: (1). Check the best action for next step that will not violate the communication loss constraint; (2) Communicate with other in-range UAVs to check if its decision is overlapped with anyone else; (3) if yes, search for another location to gain a better global utility.

5.5 Experimental Results

Results show our proposed algorithm outperforms the baseline algorithm in terms of utility gain. More significantly, robots perform the mission with weights on different sub-tasks. In this section, we will present our simulation setup then propose the results.

5.5.1 Simulation Setup

The environment of the problem is set as an indoor unknown environment. The environment is represented as a number of rooms. Rooms are connected by a edge if they are neighbors. The map we test our algorithm is shown in Fig. 5.2. In simulation, there are three kinds of rooms: office, laboratory and corridors. Each type of the room has a probability that it contains one target. There may be multiple targets, but each room can only contain one target.

To gain the environment information, UAVs are equipped with two kinds of

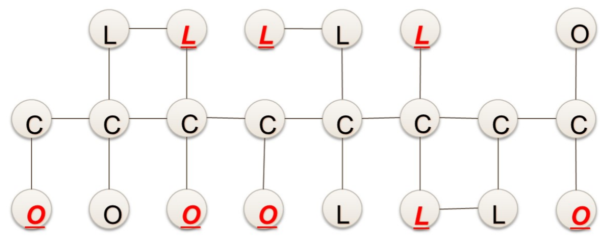


Figure 5.2: Simulated indoor environment with 22 rooms. O indicates office, L indicates Lab, C indicates Corridor. The room contains a target is underlined and colored.

sensors. One is used to map the environment (i.e., observe the type of the room), the other is used to search for targets. Both sensors have the possibility of producing erroneous measurements. In our simulation, the mapping sensing noise is set in Tab. 5.2, and the target sensing noise is in Tab 5.3.

Table 5.2: Noise of room type sense

	True Room Type		
Sense Results	Corridor	Office	Laboratory
Corridor	0.5	0.3	0.2
Office	0.05	0.9	0.05
Laboratory	0.1	0.1	0.8

Table 5.3: Noise of target sense

	Ground Truth	
Sense Results	Target	No Target
Target	0.5	0.3
No Target	0.05	0.9

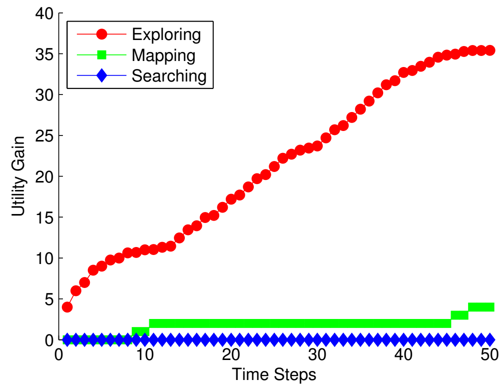
5.5.2 Results

We first present results under different weight settings for the proposed ST-EMS algorithm to show how the weight setting influences the UAVs' decisions. Then we compare the performance of the ST-EMS and baseline to present our relay mechanism's benefit. Lastly, we examine the influence of parameters changing (eg, communication loss constraint, number of robots, etc) to the outpout of the ST-EMS algorithm.

UAVs will get three types of utility during the mission: exploring utility, mapping utility, and searching utility. The performance of both algorithms under different weights is shown in Fig. 5.3. As shown in the figures, UAVs gain more utility for the higher weighted task. In Fig. 5.3(c), the exploring utility is higher since, in order to search more targets, UAVs need to visit the rooms before sensing the target in the rooms.

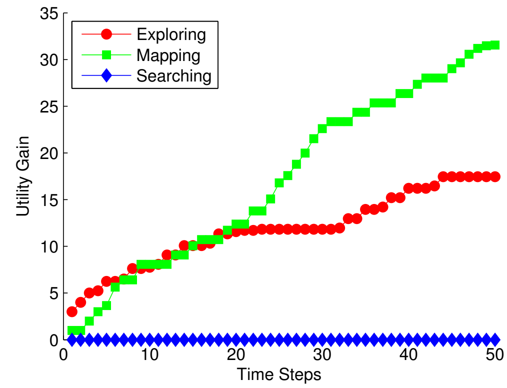
To evaluate the performance of ST-EMS, we compare the total utility gain with a 100 budget. Fig. 5.5 shows that under the same weights and communication loss constraint, ST-EMS outperforms the baseline approach consistently.

To show the scalability of our algorithm, we compare the results with the number of UAVs from 2 to 5. The simulation setup is set as Explore 0.3, Mapping 0.3, Searching 0.4. ST-EMS shows monotonic improvement in utility gain as the number of agents increases. One significant property of ST-EMS is that it allows the human controller to change the communication loss constraint in order to understand the environment better *during* the mission. To evaluate this property,

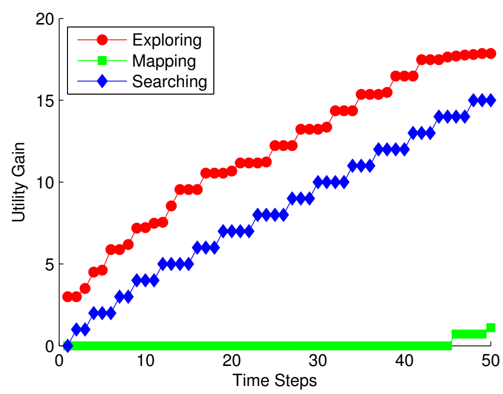


(a)

Red



(b)



(c)

Figure 5.3: Three kinds of utility gain under different weight setting for two UAVs: (a) Exploring: 0.9, Mapping: 0.05, Searching: 0.05 and (b) Exploring: 0.05, Mapping: 0.9, Searching: 0.05 (c) Exploring: 0.05, Mapping: 0.05, Searching: 0.9.

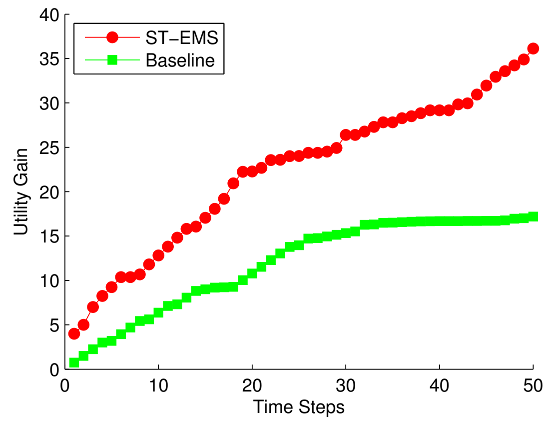


Figure 5.4: The global utility gain of ST-EMS and the baseline. The weight vector is set as [Exploring 0.8, Mapping 0.1, Searching 0.1].

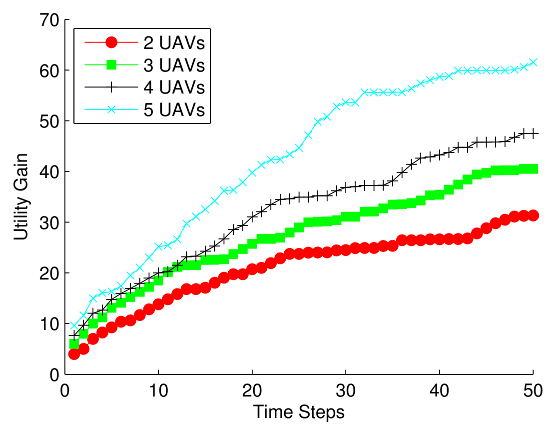


Figure 5.5: The global utility gain of ST-EMS with different number of UAVs. The weight vector is set as [Exploring 0.3, Mapping 0.3, Searching 0.4].

we compare the results with different communication loss constraints. Fig.5.6 shows the results that with the constraint increasing, UAVs are capable to explore more area, thus achieving higher information gain.

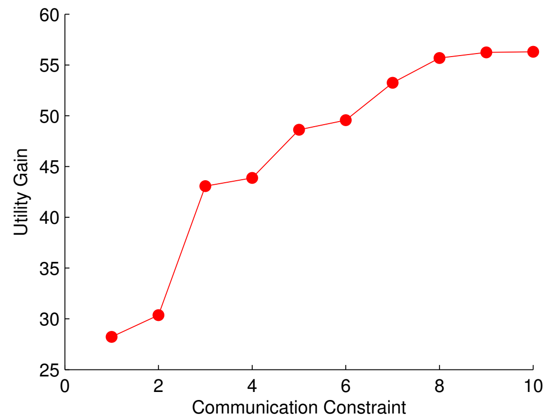


Figure 5.6: The global utility gain of ST-EMS with different communication constraint. The weight vector is set as [Exploring 0.3, Mapping 0.3, Searching 0.4].

5.6 Discussion

In this work we present an effective muli-UAV system to explore, map, and search in unknown environments. Based on our setup, the human controller can adjust the weights that drive the UAV team to focus more on one of the sub-task. Moreover, in order to keep the human controller informed of the environment features, a communication loss constraint can be set to force the UAVs to communicate with the base station. These two properties allows the human controller to change the weight setting during the mission if needed.

Chapter 6: Conclusion and Future Work

In this work we have analyzed multi-robot coordination and its applications in task allocation and information gathering problems. The main contributions in our work are demonstrated in three domains.

First, we presented an efficient bin-managing system supported by a team of robotic bin carriers. This work can potentially improve the efficiency of tree fruit harvest. For future work, we need to test the reliability and efficiency of the algorithm in real orchards. Another interesting future research area is robot coordination with human pickers. For example, we can explore learning methods to determine the humans' picking speeds, potentially yielding more efficient interaction between humans and robots.

Second, we defined a new instance of multi-robot routing for information gathering problems. We proposed to use distributed sampling algorithm for task distribution and receding horizon strategy for path planning. Future research efforts can be directed toward several aspects, including: (1) Design of a more robust task distribution algorithm. In the current implementation, task distribution is performed only once at the beginning. This is due to complexity that arises because each robot finishes its tour at a different time. When some robots are finished and the others are not, using the distributed sampling algorithm for task distribution at the current step is not straightforward. The problem becomes more complicated as the environment is dynamic (i.e., the cost of a subgraph in distributed sampling at the current time step is different from the cost at a later time). To overcome this, a form of auction may be incorporated to the task distribution algorithm.

We finally proposed an efficient multi-UAV coordination for indoor environment exploration and mapping. Our method allows for weights on one or more subtasks. Using our relay mechanism ST-EMS algorithm, human controllers can set up a communication constraint that allows the UAVs to lose communication with the base station no more than a certain time. These two properties provide better knowledge of the environment to the human operators. Therefore, operators can change their setting during the mission if the mission goal is changed.

For this scenario, many interesting research questions remain. One question is whether relay UAVs can still explore, map, and search for targets while they are relaying to other UAVs. Another potential improvement can be achieved if during the mission the UAVs can automatically change their weights based on previous experiences with human operators.

In this thesis, we reduced the difficulty of solving multi-robot problems by: (1) considering a subset of the environment to reduce the complexity; (2) employing an underlying representation to reduce the robots' action space; (3) predicting a horizon of future actions to reduce the uncertainty. These three components allowed efficient multi-robot coordination across different domains. In the future, we can potentially improve performance with (1) improved future predictions; (2) better task decomposition and allocation; and (3) integration with human operators.

Bibliography

[1] C. Phan and H. Liu, "A cooperative UAV/UGV platform for wildfire detection and fighting," in *System Simulation and Scientific Computing, 2008. Asia Simulation Conference - 7th International Conference on*, pp. 494–498, Oct 2008.

[2] R. D'Andrea, "Guest editorial: A revolution in the warehouse: A retrospective on kiva systems and the grand challenges ahead," *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 9, pp. 638–639, 2012.

[3] C. Luo, A. Espinosa, D. Pranantha, and A. De Gloria, "Multi-robot search and rescue team," in *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, pp. 296–301, Nov 2011.

[4] L. E. Parker, "Current research in multirobot systems," *Artificial Life and Robotics*, vol. 7, no. 1-2, pp. 1–5, 2003.

[5] G. Beni, "The concept of cellular robotic system," in *Intelligent Control, 1988. Proceedings., IEEE International Symposium on*, pp. 57–62, IEEE, 1988.

[6] T. Fukuda and S. Nakagawa, "Dynamically reconfigurable robotic system," in *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, pp. 1581–1586, IEEE, 1988.

[7] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa, and H. Matsubara, "RoboCup: A challenge problem for AI," *AI magazine*, vol. 18, no. 1, pp. 73–85, 1997.

[8] P. R. Wurman, R. D'Andrea, and M. Mountz, "Coordinating hundreds of cooperative, autonomous vehicles in warehouses," *AI magazine*, vol. 29, no. 1, pp. 9–20, 2008.

[9] N. Hazon, G. Kaminka, *et al.*, "Redundancy, efficiency and robustness in multi-robot coverage," in *Robotics and Automation (ICRA). 2005 IEEE International Conference on*, pp. 735–741, IEEE, 2005.

[10] E. Stump and N. Michael, "Multi-robot persistent surveillance planning as a vehicle routing problem," in *Automation Science and Engineering (CASE), 2011 IEEE Conference on*, pp. 569–575, IEEE, 2011.

[11] R. Luna and K. E. Bekris, "Efficient and complete centralized multi-robot path planning," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 3268–3275, IEEE, 2011.

[12] H. G. Tanner and A. Kumar, "Towards decentralization of multi-robot navigation functions," in *Robotics and Automation (ICRA). 2005 IEEE International Conference on*, pp. 4132–4137, IEEE, 2005.

[13] Z. Yan, N. Jouandeau, and A. A. Cherif, "Sampling-based multi-robot exploration," in *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, pp. 1–6, VDE, 2010.

[14] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, "Collaborative multi-robot exploration," in *Robotics and Automation (ICRA). 2000 IEEE International Conference on*, vol. 1, pp. 476–481, IEEE, 2000.

[15] Z. Yan, N. Jouandeau, and A. Ali-Chérif, "Multi-robot heuristic goods transportation," in *Intelligent Systems (IS), 2012 6th IEEE International Conference*, pp. 409–414, IEEE, 2012.

[16] A. Yamashita, T. Arai, J. Ota, and H. Asama, "Motion planning of multiple mobile robots for cooperative manipulation and transportation," *Robotics and Automation, IEEE Transactions on*, vol. 19, no. 2, pp. 223–237, 2003.

[17] M. Jäger and B. Nebel, "Dynamic decentralized area partitioning for cooperating cleaning robots," in *Robotics and Automation (ICRA), 2002 IEEE International Conference on*, vol. 4, pp. 3577–3582, IEEE, 2002.

[18] J. Wan, L. Zhong, and F. Zhang, "Cooperative localization of multi-UAVs via dynamic nonparametric belief propagation under GPS signal loss condition," *International Journal of Distributed Sensor Networks*, vol. 2014, 2014.

[19] A. Gonzalez-Ruiz and Y. Mostofi, "Cooperative Robotic Structure Mapping Using Wireless Measurements - A Comparison of Random and Coordinated Sampling Patterns," *IEEE Sensors Journal*, vol. 13, pp. 2571–2580, April 2013.

[20] Z. Wang, M. Zhou, and N. Ansari, “Ad-hoc robot wireless communication,” in *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, vol. 4, pp. 4045–4050, Oct 2003.

[21] S. Gil, D. Feldman, and D. Rus, “Communication coverage for independently moving robots,” in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 4865–4872, IEEE, 2012.

[22] J. Alonso-Mora, S. Haegeli Lohaus, P. Leemann, R. Siegwart, and P. Beardsley, “Gesture based human-multi-robot swarm interaction and its application to an interactive display,” in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 5948–5953, IEEE, 2015.

[23] M. S. Couceiro, R. P. Rocha, and N. M. Ferreira, “A novel multi-robot exploration approach based on particle swarm optimization algorithms,” in *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, pp. 327–332, IEEE, 2011.

[24] B. Hamner, M. Bergerman, and S. Singh, “Autonomous orchard vehicles for specialty crops production,” in *Proceedings of ASABE Annual International Meeting*, July 2011.

[25] N. Kalra, D. Ferguson , and A. T. Stentz, “Hoplites: A market-based framework for planned tight coordination in multirobot teams,” in *Robotics and Automation (ICRA), 2005 IEEE International Conference on*, pp. 1170 – 1177, April 2005.

[26] D. A. Graham and R. C. Marshall, “Collusive bidder behavior at single-object second-price and english auctions,” *Journal of Political Economy*, vol. 95, no. 6, pp. 1217–1239, 1987.

[27] J. B. Kruskal, “On the shortest spanning subtree of a graph and the traveling salesman problem,” *Proceedings of the American Mathematical society*, vol. 7, no. 1, pp. 48–50, 1956.

[28] E. Balas, “The prize collecting traveling salesman problem,” *Networks*, vol. 19, no. 6, pp. 621–636, 1989.

[29] B. Gerkey and M. Mataric, “Sold!: auction methods for multirobot coordination,” *Robotics and Automation, IEEE Transactions on*, vol. 18, pp. 758–768, Oct 2002.

[30] D. Goldberg, V. Cicirello, M. B. Dias, R. Simmons, S. Smith, and A. Stentz, "Market-based multi-robot planning in a distributed layered architecture," in *Multi-robot systems: From swarms to intelligent automata: Proceedings from the 2003 international workshop on multi-robot systems*, vol. 2, pp. 27–38, 2003.

[31] C. Tovey, M. Lagoudakis, S. Jain, and S. Koenig, "The generation of bidding rules for auction-based robot coordination," in *Multi-Robot Systems. From Swarms to Intelligent Automata Volume III* (L. Parker, F. Schneider, and A. Schultz, eds.), pp. 3–14, Springer Netherlands, 2005.

[32] J. Binney and G. S. Sukhatme, "Branch and bound for informative path planning," in *Robotics and Automation (ICRA), 2012 IEEE International Conference*, pp. 2147–2154, April 2012.

[33] J. Yu, M. Schwager, and D. Rus, "Correlated orienteering problem and its application to informative path planning for persistent monitoring tasks," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pp. 342–349, IEEE, 2014.

[34] Gurobi Optimization, Inc, http://www.gurobi.com/documentation/6.0/refman.pdf, *Gurobi Optimizer Reference Manual*, 6 ed.

[35] R. Jothi and B. Raghavachari, "Approximating the k-traveling repairman problem with repairtimes," *Journal of Discrete Algorithms*, vol. 5, no. 2, pp. 293 – 303, 2007.

[36] T. Schouwenaars, J. How, and E. Feron, "Receding horizon path planning with implicit safety guarantees," in *American Control Conference, 2004. Proceedings of the 2004*, vol. 6, pp. 5576–5581, IEEE, 2004.

[37] G. Hollinger and S. Singh, "Proofs and experiments in scalable, near-optimal search by multiple robots," in *Proc. Robotics: Science and Systems Conference (RSS)*, June 2008.

[38] J. K. H. John Tisdale, Zu Kim, "Autonomous path planning and estimation using UAVs," *IEEE Robotics and Automation Magazine*, vol. 16, no. 2, pp. 35–42, 2009.

[39] M. Garey, D. Johnson, and L. Stockmeyer, "Some simplified np-complete graph problems," *Theoretical Computer Science*, vol. 1, no. 3, pp. 237 – 267, 1976.

[40] Y.-H. Kim and D. A. Shell, "Distributed robotic sampling of non-homogeneous spatio-temporal fields via recursive geometric sub-division," in *Robotics and Automation (ICRA), 2014 IEEE International Conference*, pp. 557–562, IEEE, June 2014.

[41] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *Bell system technical journal*, vol. 49, no. 2, pp. 291–307, 1970.

[42] M. Mirzaei, F. Sharifi, B. Gordon, C. Rabbath, and Y. Zhang, "Cooperative multi-vehicle search and coverage problem in uncertain environments," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pp. 4140–4145, Dec 2011.

[43] M. Kothari, I. Postlethwaite, and D.-W. Gu, "Multi-UAV path planning in obstacle rich environments using rapidly-exploring random trees," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pp. 3069–3074, IEEE, 2009.

[44] M. Bryson and S. Sukkarieh, "Co-operative localisation and mapping for multiple UAVs in unknown environments," in *Aerospace Conference, 2007 IEEE*, pp. 1–12, IEEE, 2007.

[45] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, "Coordinated multi-robot exploration," *Robotics, IEEE Transactions on*, vol. 21, no. 3, pp. 376–386, 2005.

[46] M. Mirzaei, F. Sharifi, B. W. Gordon, C. A. Rabbath, and Y. Zhang, "Cooperative multi-vehicle search and coverage problem in uncertain environments," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pp. 4140–4145, IEEE, 2011.

[47] A. Marjovi, J. G. Nunes, L. Marques, and A. de Almeida, "Multi-robot fire searching in unknown environment," in *Field and Service Robotics*, pp. 341–351, Springer, 2010.

[48] S. K. Gan and S. Sukkarieh, "Multi-UAV target search using explicit decentralized gradient-based negotiation," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 751–756, IEEE, 2011.

[49] B. Charrow, N. Michael, and V. Kumar, "Active control strategies for discovering and localizing devices with range-only sensors," in *Algorithmic Foundations of Robotics XI*, pp. 55–71, Springer, 2015.

[50] Y. Yang, M. M. Polycarpou, and A. A. Minai, "Multi-UAV cooperative search using an opportunistic learning method," *Journal of Dynamic Systems, Measurement, and Control*, vol. 129, no. 5, pp. 716–728, 2007.

[51] S. Gil, S. Kumar, D. Katabi, and D. Rus, "Adaptive communication in multi-robot systems using directionality of signal strength," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 946–968, 2015.

[52] Y. Pei and M. W. Mutka, "Steiner traveler: Relay deployment for remote sensing in heterogeneous multi-robot exploration," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 1551–1556, IEEE, 2012.

[53] A. Aydemir, A. Pronobis, M. Gobelbecker, and P. Jensfelt, "Active visual object search in unknown environments using uncertain semantics," *Robotics, IEEE Transactions on*, vol. 29, pp. 986–1002, Aug 2013.

[54] T. Krajnik, M. Kulich, L. Mudrova, R. Ambrus, and T. Duckett, "Where's Waldo at time t? using spatio-temporal models for mobile robot search," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 2140–2146, May 2015.

[55] D. Chen, D.-Z. Du, X.-D. Hu, G.-H. Lin, L. Wang, and G. Xue, "Approximations for Steiner trees with minimum number of Steiner points," *Journal of Global Optimization*, vol. 18, no. 1, pp. 17–33, 2000.

[56] D. Du and X. Hu, *Steiner Tree Problems In Computer Communication Networks*. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2008.