A Decomposition Strategy for Optimal Coverage of an Area of Interest using Heterogeneous Team of UAVs

Mohsen Mehdizade

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science at

Concordia University

Montreal, Quebec, Canada

July 2012

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Mohsen Mehdizade

Entitled: "A Decomposition Strategy for Optimal Coverage of an Area of Interest Using Heterogeneous Team of UAVs"

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science

Complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair
Dr. R. Raut

_____ Examiner, External
Dr. K. Demirli (MIE) To the Program

_____ Examiner
Dr. M. R. Soleymani

_____ Supervisor
Dr. K. Khorasani

Approved by: _____
Dr. W. E. Lynch, Chair
Department of Electrical and Computer Engineering

_____ _____
July 31,2012 Dr. Robin A. L. Drew
Dean, Faculty of Engineering and
Computer Science

Abstract

In this thesis, we study the problem of optimal search and coverage with heterogeneous team of unmanned aerial vehicles (UAVs). The team must complete the coverage of a given region while minimizing the required time and fuel for performing the mission. Since the UAVs have different characteristics one needs to equalize the ratio of the task to the capabilities of each agent to obtain an optimal solution. A multi-objective task assignment framework is developed for finding the best group of agents that by assigning the optimal tasks would carry out the mission with minimum total cost.

Once the optimal tasks for UAVs are obtained, the coverage area (map) is partitioned according to the results of the multi-objective task assignment. The strategy is to partition the coverage area into separate regions so that each agent is responsible for performing the surveillance of its particular region. The decentralized power diagram algorithm is used to partition the map according to the results of the task assignment phase. Furthermore, a framework for solving the task assignment problem and the coverage area partitioning problem in parallel is proposed. A criterion for achieving the minimum number of turns in covering a region a with single UAV is studied for choosing the proper path direction for each UAV. This criterion is extended to develop a method for partitioning the coverage area among multiple UAVs that minimizes the number of turns.

We determine the "best" team for performing the coverage mission and we find the optimal workload for each agent that is involved in the mission through a multi-objective optimization procedure. The search area is then partitioned into disjoint

subregions, and each agent is assigned to a region having an optimum area resulting in the minimum cost for the entire surveillance mission.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Unmanned Aerial Vehicles (UAVs) are power driven aircraft without flight crew members on board that carry sensors, cameras, and communication equipment. The early motivation for developing the UAVs was due to the concern of the United States Air Force regarding loosing pilots in hostile country. The use of UAV over a manned air vehicle has significant advantages such as superior maneuverability, lower risk to human operators, weight savings, and lower development costs.

Unmanned aerial vehicles operate in various environments including but not limited to: atmospheric research [1], animal biodiversity and wildlife tracking [2], border patrol and reconnaissance [3], assessment of power lines and pipelines, traffic and accident surveillance [4], emergency and disaster monitoring [5], search and rescue [6], aerial photography [7], among others.

The use of groups of unmanned aerial vehicles (UAVs) for performing complex tasks has been an area of signicant interest recently. In the past decade, the focus of researchers has changed from developing control strategies and algorithms for a single UAV performing a mission to development of practical strategies that will allow

a group of UAVs to cooperate and perform multiple tasks. The research effort in both civilian and military domains have been focused on the development of effective multi UAVs control algorithms in order to exploit the full potential of a team of UAVs performing various tasks and reaching the mission objective. The potential advantages of multi-agent systems such as reduction of the operating time and cost, and robustness to single agent failures are the motivating factors justifying their use in performing a desired mission.

One of the applications of multi-agent UAV systems is complete coverage of an area of interest. Recent natural disasters such as the Haiti earthquake and the Indonesian Tsunami have highlighted the need for an efficient and timely applicable search and rescue strategy. Accurate and timely information regarding the place of injured people are crucial for effective search and rescue missions. Use of a team of UAVs is among the best options for these problems since the UAVs can perform their tasks in minimum amount of time, and can be utilized in dangerous environments without endangering a human pilot.

The main objective of research into the cooperative UAV control problem is to develop and evaluate strategies (algorithms) for a team of UAVs working together to perform mission specific tasks in an extended area under varying operating conditions and constraints. Collecting and processing data under constrained resources such as limitations in the fuel, restrictions in time, and constraints in the search region should be reflected in developing a strategy for cooperative coverage by multi-agent UAVs. For a given mission scenario, the cooperative UAV control problem formulation encompasses the following common sub-problems: Path Planning, Trajectory Generation, and Task Allocation. The complexity of these sub-problems depends on the chosen mission scenario, the capabilities of the UAVs, the complexity of mission

environment, and the constraints added by the mission designer.

The research in multiple UAV search and coverage problem is aimed at developing and evaluating strategies for a team of UAVs searching an environment of known dimensions for (stationary and/or mobile) targets under varying operating conditions and constraints. Searching an area of interest using a group of vehicles is a problem that finds applications in both military and civilian domains. Some of the possible missions that could benefit from multiple UAVs conducting search are: search-and-rescue operations [6], search-and-destroy missions for previously detected enemy targets, seek-destroy missions for land mines, and surveillance missions such as border patrol.

The problem of searching an unknown environment has been actively studied in classical search theory, where problems such as optimal distribution of resources, maximization of the probability of revealing static targets, and establishment of optimal values of various search parameters have been concentrated on. Moreover, additions have been made to the search theory literature by inclusion of mobile, multiple, and intelligent targets in the problem formulation, however, solution strategies have only been developed for a single searcher, and hence are not applicable to multiple searchers.

1.1 Literature Review

A multiple UAV search problem can be formulated in a number of different ways. The variations in the problem formulation are due to the different objectives of the UAV team, the imposed constraints under which the UAVs are operating, the assumptions made about the UAVs' capabilities, and the varying criteria in location of operation. These variations in the problem formulation necessitate unique solution strategies

based on the problem formulation.

The capabilities of UAVs are important elements in deciding how to allocate the tasks between individual UAVs. The solution strategies that are developed for homogeneous team of UAVs are often not applicable (or not optimal) in missions that involve heterogeneous UAVs. Hence, existence of heterogeneous UAVs necessitates efficient task assignment strategies.

The operation setting is a deciding factor on the dynamics of the solution strategy. The search environment may have obstacles and no fly zones such as mountains, that makes it a non-convex region for UAVs. Existence of multiple, stationary and/or mobile, intelligent, known and/or unknown targets and threats can dictate whether static or dynamic solution strategy is suitable for solving the search problem.

Some of the prominent constraints in the UAV literature are constraints on a vehicle's maneuverability, range of speed, fuel bank, endurance time, and communication abilities. Examples of assumptions made are the following: (1) Vehicles may or may not be capable of wireless communication, (2) The information base of the vehicles, representing the vehicles state, might be centralized or decentralized, (3) The sensors of the vehicles might have the same or different accuracy and ranges, and the vehicles may or may not be identical in terms of their abilities to perform tasks (homogenous or heterogeneous), and (4) The vehicles may or may not be autonomous (capable of independent decision making without human guidance).

There are two types of search and coverage problem in the literature. One is the static coverage where the problem is to locate a set of sensors to cover the entire region [8], [9], [10], [11]. In this type of coverage the range of sensors are usually

comparable with the area of workspace, and the goal is to find the optimal position for a team of sensors to maximize a reward function. In [12] authors have proposed a decentralized control strategy for placing the agents in their optimal position. A generalized Voronoi diagram (Power diagram) is applied to partition the workspace into subregions where workload is equal among all subregions.

The second type of coverage is dynamic coverage, where the agents move and visit the non-surveyed areas in order to cover every point in the workspace [13], [14]. The area of sensor footprint in the dynamic coverage is much less than the total area of the workspace. In [15], the authors addressed the problem of coverage with guaranteed collision avoidance. In [16], the authors proposed a coverage algorithm by covering the area or the map with disks of radius equal to the radius of the UAVs' sensor footprints.

Another area in multi-agent search literature is persistent coverage. The problem of persistent coverage is a special type of dynamic coverage and it is distinct from most other sweeping and monitoring scenarios in the literature because in persistent coverage problem every point in the area must be revisited every T^* units of time by at least one agent. In this type of coverage the team of vehicles consistently re-covers a given area and the agents must persistently move to satisfy the objective. A survey of sweep coverage is given in [17]. The challenge in the persistent coverage is in controlling the vehicles to move so that their footprints cover all points in the environment consistently. The vehicle must spend more time in those positions where the environment changes quickly, without ignoring the locations where the environment changes more slowly. The authors in [18] consider the situation in which robots are constrained to move on fixed paths and their speed must be controlled. In other related works, a region is persistently covered in [13] and [19] by controlling the robots to move at constant speed along predefined paths.

The problem of dynamic coverage with heterogeneous agents is an interesting research subject [20], [21]. Most of the literature in dynamic coverage that assumed to have homogeneous agents cannot be extended for scenarios including heterogeneous team of agents. Given that these algorithms use sensor's footprint radius or the speed of the homogenous UAVs as an important component and information of the algorithm, they fail when agents have different characteristics [15]. Another major limitation of the existing algorithms on dynamic coverage with heterogeneous UAVs is that of assigning optimal workload to UAVs based on their capabilities [22].

The search coverage problem for non-homogeneous vehicles can be divided into sub-tasks that are then assigned to each vehicle. The space of possible sub-tasks grows with an increase in the size of the search space, the number of vehicles, and the levels of changes in their capabilities. Complexity of the coverage region such as the existence of obstacles, or the non-convexity of the map has direct impact on complexity of the solution to the problem.

In [24], the authors have addressed the problem of multiple UAVs carrying out a search and surveillance mission based on the uncertainty map of a given unknown environment. The uncertainty map represents the *a priori* knowledge of the location of targets, and is comprised of real numbers between 0 and 1. The search environment is divided into identical hexagonal cells, where each cell has an associated uncertainty value, which represents the extent of the lack of information about a cell. The UAVs are directed to fly multiple sorties (or missions). For each mission, the objective of the UAVs is to maximize the reduction in uncertainty of the environment under a limited fuel constraint, which also constrains the length of the search path each vehicle takes since the vehicles have to return to their respective base stations for re-fueling at the

end of each mission. The search algorithm, based on the k-shortest path algorithm [25], directs individual UAVs to search (independently of their team members) through the area of maximum uncertainty in the environment while satisfying constraints on their endurance time and on the search path.

In [26], the authors introduced a game theoretical model in which each agent updates its uncertainty map through the mission by using the current position and route information of the other UAVs. Therefore, the UAVs always have the same uncertainty map, and they are aware of the past path and present position of their team members. At each time step, the UAVs aim to select their future paths such that the reduction in uncertainty of the environment is maximized. The authors have proposed three game theoretical search strategies to direct the UAVs to choose their future paths to maximize the reduction in uncertainty in the environment.

In [27], an agent based negotiation scheme was presented. The agents can only communicate with their neighboring vehicles and they have a limited sensor range. The objective of each UAV in the team is to coordinate with its neighboring UAVs to select search routes such that the reduction in uncertainty of the environment is maximized. Every agent selects the next cell in its search route after sharing the information with its neighbors at each time step and performing a negotiation process. This process continues until the search area is entirely explored.

In [28], a search-theoretic approach based on the concept of rate of return maps was used to develop cooperative search plans for a team of UAVs. The rate of return map shows the benefit of searching a cell with a small increment of effort. The UAVs in [28] are exploring a given environment, divided into cells, for stationary targets. The objective in this search theoretic problem is to find the optimal amount of effort

in the environment within a given constraint on effort.

Voronoi diagram [29] plays an important role in multi-agent missions in search and coverage. Suzuki and Okabe [30] have discussed various applications of Voronoi diagrams. Some static algorithms aim to divide the region by using established shapes such as triangles and polygons, while other approaches use Voronoi diagrams positioned on various points in the region to perform a space partitioning. Voronoi diagram is also adopted to solve a coverage problem in [32]. However, the strategy does not completely cover the environment. The authors in [9] develop a strategy for dividing a polygonal region into sub-regions of equal area by using Voronoi diagrams. Their work would result in optimal workload sharing for homogenous agents.

The authors in [31] consider the problem of locating M facilities on the unit square. The facilities must meet the demands of customers. In this scenario when a customer has a demand it will utilize the closest facility. The capacity of each facility must be sufficient to satisfy all accepted demand. Given the assumption that all the facilities are similar, the capacity of each facility will be determined by the busiest facility. Therefore, any facility having less demand than the busiest one will have some unused capacity. By using this assumption the authors addressed the problem of selecting the locations to minimize the total cost by finding locations that minimize the demand at the busiest facility.

One of the most significant research conducted into the use of Voronoi diagrams in multi-agent coverage is in [11]. The authors have proposed a gradient descent control law that depends only on the information of position of the robot and of those robots that are located in the neighboring Voronoi cells. Their proposed control law can be computed without global synchronization and it leads to optimal coverage for a group

of homogenous vehicles.

The work in [34] has used vehicle positions as the set of generators for the Voronoi diagram and it analyzed the Voronoi partitioning based on energy functions. The authors developed a Voronoi partition that requires the minimum energy for a vehicle to move from one point to another in a constant flow environment. They also used Voronoi partition to design efficient task allocation algorithms. Each vehicle is assigned to targets that fall into its Voronoi region. By properly controlling vehicles motion the group requires minimum average energy in servicing stochastic tasks with slow-rate Poisson distribution arrival.

Moarref and Sayyadi in [35] addressed the problem of optimal facility positioning on a continuous plane. They proposed an asynchronous, multi-agent robotic system approach that minimizes the weighted sum of the Euclidean distances of a set of target points to their nearest facility. In their approach, when the target points move to other locations, the position of each facility moves toward the geometric median of its corresponding Voronoi cell.

Existence of mobile, intelligent targets and threats in the search environment imposes the need for developing dynamic solution strategies. In the presence of uncertain and adverse environment with intelligent targets and threats, dynamic modification of the team mission strategy and adjustments in individual UAV plans need to be considered. The authors in [36] considered the problem of dynamic reassignment of tasks among a cooperative group that exchanges relevant information asynchronously. In [37] sensor relocation in mobile sensor networks is discussed. The authors propose relocating redundant mobile sensor nodes to fill the coverage holes. They used a cascade movement method to balance the energy cost and the repairing

time in the team. In [39] the use of a small team of mobile robots is discussed to replace failed sensors in a large-scale static sensor network.

Developing robust strategies is another important area in multi-UAV missions. The authors in [40] have considered the effects of uncertainty in the data for task assignment problem. They have developed a strategy that integrates two approaches for improving the robustness of the assignment algorithm. One approach is to design task assignment plans that are robust to the uncertainty in the data in order to reduce the sensitivity to errors. A second approach is to re-assign the tasks each time that the data is updated, which leads to the best plan given the current information. However, this approach may cause instability if the updates are performed rapidly. In [45], the authors studied the performance of a team of UAVs in attack and destroy missions subject to actuator faults in one or more agents. The effect of fault on the performance of the team is studied by comparing the results of completing the mission with a faulty agent and the result of dropping out the faulty agent from the available resources for completing the mission.

Another important aspect in a real set-up is that the region of interest may possibly contain some obstacles, i.e., the region is a non-convex environment. Some works have been devoted to deal with this issue. In [44] Breitenmoser *et al.* have suggested a two stage control strategy to drive the agents to the nearest point to the center of their partition. This method uses the classical Voronoi coverage with the local path-planning algorithm TangentBug [41], to handle the motion of the robots around obstacles and corners. They used Lloyd algorithm [42] to update the goal position of vehicles at each step, while TangentBug determines the path to the goal position. The authors in [43], [44] have suggested transforming a special form of non-convex region into the convex region, and then using the methods for convex

region for covering the region. They have shown how a diffeomorphism can be used to transform a non-convex problem into a convex one. The drawback for this work is that there is no control on the partitioned area in non-convex region, as when transforming the equally partitioned area in transformed convex region into the original non-convex map, there is no general way to predict the regions.

Surveillance problem is a variation of search mission. In search missions, the goal is to locate targets that are present in the search environment. However, for a surveillance mission, the UAVs have the information about the location of the targets and their goal is to closely observe the given target (a geographical region for border patrol, a region of specified radius in open-ocean for maritime surveillance, the perimeter of a wildfire, etc.) by utilizing their resources (camera, sensor, information of other UAVs). Civilian applications of surveillance missions include monitoring oil fields, pipelines, forest fires, oil spills, and tracking wildlife, whereas military and homeland security applications include border patrol, maritime surveillance, monitoring the perimeter of nuclear power plants.

Perimeter surveillance is collecting and processing information at all points of the perimeter. The team of agents is set to cooperatively collect data about the state of the perimeter. For example, in case of forest fire monitoring [47] the team of UAVs attempt to capture images along the perimeter of the fire and then distribute them among the team members to upload the data about the location of the changing fire perimeter to a base station as frequently and with as little delay as possible. In [48], perimeter surveillance algorithms have been developed for a team of UAVs, which is set to cooperatively monitor and track the spread of a forest fire. The authors measured the performance of the proposed centralized and decentralized monitoring algorithms by comparing the frequency and time delay of information being updated

at the base station.

Solving path planning problem is an important part of search mission with unmanned aerial vehicles. The problem in path planning is to generate a straight-line waypoint path from a UAV's current position to its desired position while satisfying the mission objectives and cooperation constraints. By considering the kinematic constraints of a UAV, such as constraints on velocity range, minimum turning radius and their effect on UAVs maneuverability, one can no longer use simple straight lines for generating the path of UAV(s) and it becomes necessary to solve a Trajectory Generation problem. In trajectory generation problem the objective is to develop a real-time trajectory that allows a UAV to travel through a series of waypoints in a time optimal, and that satisfies the kinematic and dynamic constraints of a UAV [49].

A path planning algorithm and a centralized cooperation scheme has been developed in [47], which is extended in [48] to include a decentralized cooperative surveillance scheme, and a real-time tracking algorithm that allows the team of UAVs to track the perimeter of a fire. The number of agents in the team and the perimeter of fire are known to each UAV in the mission. The proposed approach is capable of monitoring a changing fire perimeter, and the ability to transfer and update the collected data with maximum frequency/time delay ratio to the base station.

In [50], different path planning and trajectory generation algorithms were compared based on the following criteria: avoidance of generated trajectories from known threats, consideration of dynamic constraints of a UAV, and computational efficiency of algorithm. The path planning and trajectory generation methods that have been investigated are Rectilinear Grid [51], Voronoi Grid, Mass-Spring-Damper System [52], Chain-Link System [53], Voronoi Grid Approximation [54], Polynomial Basis

Functions, Cubic Spline Basis Functions [55], and Voronoi Decomposition Approach. In [54], the Voronoi Grid approach is utilized to create multiple waypoint paths that satisfy the mission objectives and cooperation constraints. These paths are generated from a vehicle's current position to its goal. After generating waypoint paths graph search algorithms, such as A* [56], Dijkstra [57], or k-best path algorithms [58] are used to search the Voronoi diagram and find the optimal path with respect to the mission objective. In [59] a decentralized algorithm for cooperative movement of a team of autonomous vehicles is suggested to minimize the waiting time/service ratio of stochastically-generated targets. The vehicles have a limited range of velocity and travel within a convex environment, and the targets were generated by using a spatio-temporal Poisson models.

Performing the missions efficiently is a crucial factor when the agents have some constraints on the amount of fuel or the mission should be completed in certain amount of time. The authors in [33] addressed the problem of static coverage optimization with networks of mobile sensors. The objective of the mission is to locate sensors in the environment so as to minimize the team servicing cost. The probability of an event occurring in a region is given by a density function and is available to all the sensors. Furthermore, the cost to service a location is known to all the agents. Furthermore, the area of the assigned region to each robot is a pre-specified value because of load balancing considerations.

In [38] the authors propose a strategy for minimizing the number of sensors for coverage task. The environment is divided into equal-size square cells which is either open or occupied by a static obstacle. The location of obstacle is not known to the agents and the agents should travel around the environment to discover and respond to obstacles. The optimum number of mobile sensors is determined

by selecting robots speed which is calculated based on the remaining fuel and remaining time before a deadline. The rule for determining the robots speed is that the robot should move with the most energy efficient speed if it has sufcient time to complete the coverage, and the robot should travel faster if the deadline is imminent.

Ahmadzadeh *et al.* studied coverage aerial surveillance in [60], [63]. The mission goal is to maximize the amount of covered area within pre-set deadline. The authors addressed restrictions in sensor range that is imposed by body-fixed cameras. In [64] and [65] a decentralized strategy is proposed for determining the path of UAVs in the search mission. The objectives for the UAVs in determining their paths are to avoid approaching neighboring UAVs, distance from search boundaries, avoid re-visiting an area, and maintain the current heading. However, they have not provided any proof on the performance of the team in a mission. In other words, They have not shown that with this strategy the team can cover the whole region of interest.

The reference [66] presents a method to deploy multiple robots into a field for exploration purposes by considering the energy and time constraints of the mission. They presented a deployment strategy to increase the average covered area among the team of robots. Their method assigns a rectangle area to each robot to cover and keeps the minimum covered area in different groups. The authors extend their strategy in [67] by relaxing the constraint on the similarity of speed among the robots and including the existence of obstacles in the environment.

The work in [68] studied the problem of optimal task allocation in multi-agent networks. The agents have different sets of capabilities that are suitable to perform different tasks. The team objective is divided into a set of sub-problems and the optimal group of agents is allocated to each sub-problem in order to complete it in

shortest amount of time.

In [22], a team of heterogeneous mobile robots is set to cover a region. The robots are considered to be heterogeneous in that the sensor footprints are different. The proposed strategy generates the trajectory for the robots to travel to their final positions without colliding with other robots. The authors use a power diagram for partitioning the region, and they use the radius of the robot sensors as the weight of Voronoi generators. The drawback of this method is that the generated partitions do not have equal area/capability for the agents; thus, it does not lead to the best positioning for the mobile sensors. Bazler [8] considered the problem of capacity-constrained Voronoi diagram in the continuous space. The problem is to partition a map into different sub-partitions with predefined areas. Bazler proposed two approaches with ordinary Voronoi diagram and additive and multiplicative Voronoi diagrams.

1.2 Problem Statement

Heterogeneous robot cooperation technologies have special importance in the multi-robot research. In real world applications, the agents of a robot team are often different in design, structure, and sensor configuration as well as intelligence. Hence, they are not homogeneous systems. In presence of heterogeneous agents the problem of task assignment is more complicated than the scenarios when all agents have similar characteristics. When a homogeneous team of agents is tasked to do a mission an efficient way to carry out the mission is to assign each agent the same workload as those assigned to others. However, this would not lead to an efficient solution when heterogeneous agents are involved in the mission. In fact, utilizing all the agents in a mission may not be the best strategy in some scenarios. Based on agent capabilities,

some might not be required to participate in the mission.

In this thesis, we address the problem of dynamic coverage of an area of interest by a team of heterogeneous UAVs. A team of heterogeneous UAVs is tasked to visit a region of interest ($\Omega \subset \Re^2$) and perform a complete coverage search. Every point in the region of interest should be visited by at least one of the agents during the mission. A point is visited if it is inside the sensor footprint of a UAV at any time during the mission. Each UAV has a sensor, which collects data about the environment in which it is traveling. The UAVs are heterogeneous in the sense that they have different sensor footprints and different range of speeds. The mathematical formulation of the problem includes the following components and constraints:

- m Non-Homogenous UAVs: each vehicle has a range of forward speed $v_{imin} < v_i < v_{imax}$, where v_{imin} and v_{imax} denote the minimum and maximum speed of UAV_i respectively.
- Coverage Region $\Omega \subset \Re^2$: is a compact and convex subset of $\Re^2$, and has a total area of D.
- Assigned Region Ω_i: is the UAV_i coverage region which is a subset of Ω.
- Radius of Sensor Footprint r_i: each UAV is equipped with a sensor. The sensor footprint for UAV_i has a radius of r_i.

There are two important factors in the mission. First, the time each UAV needs to complete its search, and second the amount of fuel that is consumed for each UAV. The goal of the team of UAVs is to completely visit the given region while minimizing the cost incurred by the team. The mission cost includes two cost functions representing the fuel cost J_f and the time cost J_t. The fuel cost of the team is equal to the sum of

the fuel cost for each UAV J_{f_i}, which is a function of UAV's fuel consumption rate and the length of the path covered by the UAV. The fuel consumption rate is proportional to the aerodynamic drag force, which is proportional to the velocity squared. Hence, the fuel required by the i^{th} UAV to search its assigned region, is given by $J_{f_i} = v_i^2 l_i$. We use the sum square of all UAVs' finishing time for determining the time cost of the team. Therefore, the costs J_f and J_t can be expressed as:

$$\begin{aligned} J_f &= \sum_{i=1}^{m} v_i^2 l_i \\ J_t &= \sum_{i=1}^{m} t_i^2 \end{aligned}$$

where v_i, l_i, t_i are the velocity, the length of path, and the mission time of UAV_i, respectively.

1.3 Proposed Solution

The problem of searching an unknown environment has been actively studied in classical search theory, where problems such as optimal distribution of mobile sensors, maximization of the probability of detection of stationary targets, and establishment of optimal values of various search parameters have been concentrated on [17]. While other considerations have been made to the search theory literature by inclusion of mobile, multiple, and intelligent targets in the problem formulation, solution strategies have only been developed for a single search, and hence are not applicable to multiple searches. Developing strategies for multiple searches requires that one takes into consideration all the agents in the scenario. This will lead to huge computational cost in time critical missions where the time optimality is a big concern. In optimal case of search with multiple UAVs, one needs to minimize the amount of overlapped coverage

among the agents. Hence, the trajectory generation of each agent has direct effect on the path of other agents. The best way to minimize this redundant coverage is to isolate each agent task from the others. This has been addressed in the literature for homogenous teams of agents [19]. However, these methods would not produce optimal results when the agents have different capabilities. The idea in this thesis is to develop a framework that make single search strategies possible to use in multiple searches scenarios. We propose a task assignment framework that will simultaneously assign an optimal task to all the UAVs, and to isolate their task from the rest of the team.

The problem of dynamic coverage can be divided into 3 sub-problems, namely (*i*) task allocation, (*ii*) space decomposition, and (*iii*) path generation. We address the region decomposition and the task allocation of the coverage mission in two phases. In the first phase, the objective is to find the optimal speeds and coverage areas for all the UAVs via a multi-objective optimization strategy. The use of multi-objective optimization makes considering different goals and costs for the mission, and gives the mission designer the possibility to choose from a set of possible solutions based on the importance or priority of each objective in the mission. Another advantage of using multi-objective optimization is that adding and removing an objective from the set of objectives does not change the solution strategy. The multi-objective optimization results in a set of different possible assignments for the team and the mission designer can determine the favorite assignment based on the mission criteria and also past experiences. Following the optimization phase and after choosing a solution as the preferred result of the optimization phase, the second phase would divide the region of interest into subregions according to the mission designer favorite set of task assignments. This phase assigns each UAV a subregion of Ω that matches with the solution of the first phase, i.e. each agent's area of coverage is equal to

its optimal coverage area from the first phase. By the end of the second phase all the UAVs are assigned to known regions and the optimal speeds for surveying the assigned regions are also prescribed to them. Hence, both task allocation and space decomposition subproblems are solved. For the path generation we can use any single search path generation algorithm since each agent's, subarea is isolated from the rest of the agents and one agent's trajectory does not affect the generation of the other trajectories.

1.4 Contributions of the Thesis

The focus of the research in this thesis is the wide-area coverage, which includes tasks such as search and rescue, security patrols, forest-fire monitoring, aerial mapping and reconnaissance missions. In search and rescue missions it is very important to search the entire region in minimum time, while in applications such as aerial mapping other costs such as fuel might be more important to the user. The framework presented here attempts to minimize the mission cost while searching and covering the entire map.

This thesis presents a new decomposition strategy that can be used to solve coverage problems with different objectives in presence of multiple heterogeneous agents. Multi-objective optimization techniques were chosen to model the task assignment problem to generate optimal tasks and also due to the intuitive nature of modeling these problems as a set of objective functions with a set of constraints.

The research presented in this thesis implements a task assignment strategy for heterogeneous aerial vehicles that is especially suited for large area surveillance problems. The system chooses optimal area for each vehicle and finds the optimal speed corresponding to the assigned area that minimizes the mission cost. Afterwards,

the system partitions the map based on the optimal areas for each UAV. The output of the system is a subregion with determined speed of coverage for each vehicle that is assigned to the coverage task. The advantages of using this framework are described below.

We should point out that our method is capable of handling different types of UAVs for performing the coverage task. The UAVs have different range of speed and their sensor footprints are also different. Another advantage of our method is its consideration of multi-objective functions, which allow the mission designer to make decisions based on all the important factors in the mission. Our map partitioning method prevents UAVs from having overlaps in their path coverage. Furthermore, we use area decomposition after determining the optimal share of each UAV. Performing the area decomposition before determining the capacity of each UAV leads to forcing the UAVs to choose between the generated regions and thus results in an non-optimal shares for agents. Determining the optimal workload for each agent according to the mission objectives allows us to perform the mission with minimum cost. Thus, the optimality of the proposed strategy makes it suitable for implementation it in the coverage missions.

This dissertation details the following contributions:

- The decomposition of coverage mission into task allocation, space decomposition and path generation presented in this thesis offer cost optimality and the possibility for real-time implementation that most existing coverage algorithms lack. The current coverage algorithms in the literature can be divided into two parts. One is focused on generating the optimal solutions for the coverage mission. However, the drawbacks of these methods are their huge computational cost. Thus, these algorithms cannot be implemented on a real UAV. The optimality of

coverage strategies can be understated when the focus is on developing real-time algorithms. By decoupling the UAVs paths from one another in our method, we improve the computational complexity of the UAV paths. Furthermore, isolating UAVs area of coverage guarantees that no redundant visits is made by any UAV in the mission. Thus, this helps in generating optimal solutions.

- The framework developed in this thesis makes use of a single agent search literature. Although the coverage problem is extensively studied in the single agent literature, however, these algorithms cannot be extended into multiple agent scenarios. With isolating the UAVs area of coverage and consequently their paths, we can use the literature that is related to a single agent coverage to select a suitable trajectory-planning algorithm for our method. The trajectory planning in our framework is the final phase of the coverage mission. In our framework after performing the multi-objective optimization and space partitioning each agent has the information about its travel speed and the sub-region of the map that it should search and cover. Thus, at this point the problem has been reduced to trajectory generation of single agent that is well researched in the literature.

- Using a multi-objective optimization problem formulation to address the task assignment problem makes it possible to change the objective functions and add/remove objectives based on the requirement of the mission and concerns of the user. In this thesis we considered the completion time of the mission and the overall fuel used to perform the mission as the two main objectives of the mission. However, it is possible to add more objectives or constraints to the scenario, such as limiting the fuel of each vehicle to its fuel bank or having some

constraints on the finishing time of one or more UAVs.

- Another advantage of using the multi-objective optimization for the task assignment is the selectivity of most suitable agents for the mission. Using all the UAVs to perform the mission would not always result in the best solution with regards to the mission objectives. When the agents are homogeneous one is more likely to have better performance by involving all the agents in the mission, however, in scenarios containing heterogeneous UAVs the best team for performing the mission varies by the changes in the mission objectives. When the importance of the mission is to complete its task as fast as possible it is best to utilize more agents than when the priority of the mission is to minimize the fuel.

The above contributions and attributes are shown to be satisfied through extensive computer simulation results.

1.5 Conclusions

In this chapter, we have presented and reviewed the multiple UAV coverage problem in detail. The multiple UAV search and coverage problem is aimed at developing and evaluating strategies for a team of UAVs searching an environment of known dimensions for (stationary and/or mobile) targets under varying operating conditions and constraints. The problem of searching an unknown environment has been actively studied in classical search theory, where most of the strategies have been developed for a single search agent. However, recent development in multi-agent systems and their potential advantages such as reduction of the operating time and cost, and robustness to single agent failures has motivated many researchers to develop algorithms for multi-agents search and coverage problem.

A literature review on different types of search and coverage problems is presented in this chapter. A multiple UAV search problem can be formulated in a number of different ways. The variations in the problem formulation are due to the different objectives of the UAV team, the imposed constraints under which the UAVs are operating, the assumptions made about the UAVs capabilities, and the varying criteria in the location of the operation.

The focus of this thesis is on developing a new framework for the coverage problem with heterogeneous UAVs. In our strategy the problem is broken down into 3 subproblems: task assignment, space decomposition, and path generation. In the task assignment phase the best agents are selected and are tasked to cover a specific share of the entire region with specified speed. Then the region of coverage is partitioned into subareas based on the results of the task assignment phase. Finally, for path generation the paths with minimum turn are selected for the UAVs.

Chapter 2

Background Information

In this chapter we review some basic concepts from multi-objective optimization, space partitioning, and path generation that are used throughout this thesis.

This chapter is organized as follows. The multi-objective optimization problem and the methods for solving these problems are presented in Sections 2.1 and 2.2, respectively. The space partitioning with Voronoi diagrams are then reviewed in Sections 2.3. In Section 2.4, we present the minimum turn trajectory planning for multiple agent systems.

2.1 Multi-Objective Optimization

Multi-objective optimization is concerned with solving problems that have two or more, often conflicting, criteria. In the single objective optimization problem there is only a single globally optimal minimum or maximum as the solution and every two points are comparable to one another, i.e. one solution is either better, worse or equal in comparison to another solution. However, in the multi-objective optimization all the points are not always comparable and there may not exist the best solution with respect to all the objectives.

The goal in the multi-objective optimization is to find a set of solutions known as Pareto set [97], [98]. None of the solutions in Pareto set are better than others and all are considered as viable and acceptable solutions. For a Pareto point when one attempts to enhance one of the objectives, other objectives will suffer as a result. Multi-objective optimization problems can be found in various fields: product and process design, finance, aircraft design, oil and gas industry, automobile design, and actually wherever optimal decisions need to be taken in the presence of trade-offs between two or more conflicting objectives. Maximizing profit and minimizing the price of a product; maximizing performance and minimizing fuel consumption of a vehicle; and minimizing weight while maximizing the strength of a particular component are examples of multi-objective optimization problems.

If a multi-objective problem is well formed, there should not be a single solution that simultaneously minimizes each objective to its fullest [98]. In each case an objective must have reached a point such that when attempting to optimize the objective further other objectives suffer as a result. Finding such a solution, and quantifying how much better this solution is compared to many other such solutions, is the goal that one attempts to satisfy when setting up and solving a multi-objective optimization problem.

2.1.1 Multi-Objective Optimization Problem Formulation

In mathematical terms, we can write a multi-objective optimization problem as:

$$\begin{aligned} \min_{x} \quad & F(x) = [f_1(x), f_2(x), ..., f_k(x)] \\ s.t. \quad & \\ & g_t(x) \leq 0 \qquad t = 1, 2, ..., m \\ & h_s(x) = 0 \qquad s = 1, 2, ..., p \\ & x_l \leq x \leq x_u \end{aligned} \tag{2.1}$$

where $x \in R^n$ is a vector of design variables (also called decision variables), n is the number of independent variables x_i, $F(x)$ is a vector of objective functions where $f_i(x) : R^n \rightarrow R$ is the i^{th} objective function (also called criteria, pay-off functions, cost functions, or value functions), k is the number of objective functions, x_l and x_u are the lower and the upper bounds of the decision vector, respectively, g_t and h_s correspond to the inequality and the equality constraints, respectively, m is the number of inequality constraints, and p is the number of equality constraints.

The *feasible design space* X is the set of points where the conditions for the constraints are not violated, that is

$$X = \{x \mid G(x) \leq 0 \; and \; H(x) = 0 \; and \; x_l \leq x \leq x_u\} \tag{2.2}$$

where $G(x) = \{g_1(x), ..., g_m(x)\}$ and $H(x) = \{h_1(x), ..., h_p(x)\}$.

The *feasible criterion space* Z (also called the feasible objective space) is the map of

feasible design space in the objective space, that is

$$Z = \{F(x)|x \in X\} \tag{2.3}$$

For the rest of this thesis, we use the term "design space" for the feasible design space and "objective space" for the feasible objective space, unless explicitly expressed otherwise.

2.1.2 Basic Concepts

The objective in solving the multi-objective optimization problem in equation (2.1) is to find the optimal points in the design space where their corresponding points in the objective space have the minimum value. In single objective optimization the criteria for optimality is the value of one objective function and one can always compare the value of one point to another, one solution is either better, worse or equal in comparison to another solution. Often, in multi-objective optimization there is no single global solution where all the objective functions attain their optimum. Therefore, it is necessary to determine a set of points that satisfy a preset definition for an optimality. The definition for an optimal point is the concept of *Pareto optimality* [100], and is defined as follows:

Definition 1. *Pareto Optimality: A decision vector in the design space* $x \in X$ *is Pareto optimal iff there does not exist another vector* $y \in X$ *such that* $f_i(y) \leq f_i(x)$ *for all* $i = 1, ..., k$ *and* $f_j(y) < f_j(x)$ *for at least one index* j.

There is another type of optimality for a decision vector in the multi-objective optimization known as "weak Pareto Optimality". The algorithms for solving the multi-objective optimization problems usually provide points that may not be Pareto

optimal but are weakly Pareto optimal. The definition for the *weakly Pareto optimality* is as follows:

Definition 2. *Weak Pareto Optimality: A decision vector in the design space* $x \in X$ *is weakly Pareto optimal iff there does not exist another vector* $y \in X$ *such that* $f_i(y) < f_i(x)$ *for all* $i = 1, ..., k$.

The difference between Pareto optimal point and weakly Pareto optimal point is that for a point to be weakly Pareto optimal there should not exist another point in design space that improves *all* of the objective functions simultaneously. However, the condition for Pareto optimality is that there should not exist a point that improves *at least one* objective function without detriment to another objective function. Pareto optimal points are weakly Pareto optimal, but the reverse is not always correct.

Definition 3. *A point* $x_1 \in X$ *is a solution of multi-objective optimization problem (2.1) if there is no other* $x_2 \in X$ *where*

$$\begin{aligned} & f_i(x_2) \leq f_i(x_1) \quad \forall i \in \{1, 2, ..., k\} \\ & \exists\, j \in \{1, 2, ..., k\} \quad f_j(x_2) < f_j(x_1) \end{aligned} \tag{2.4}$$

Each point in the solution set is a Pareto point, and the set of all the solution points is often called Pareto front. All the Pareto optimal points map into the boundary of the objective space Z [101].

In Figure 2.1 the Pareto point and the Pareto front in the 2-D objective space

are shown. As shown the thick line contains all the Pareto optimal objective vectors.

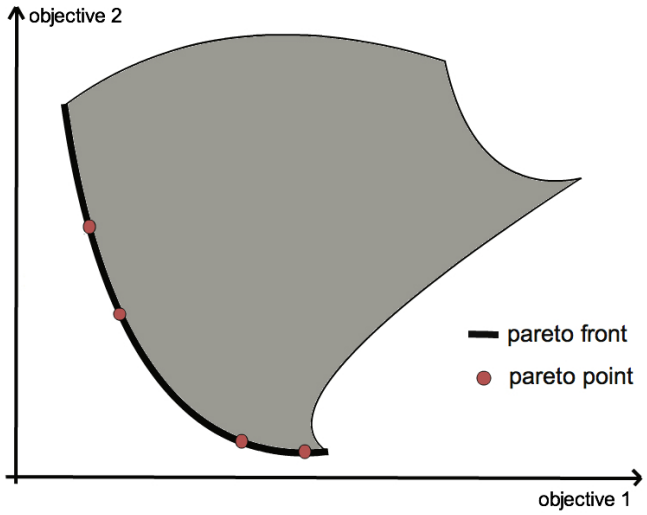

Figure 2.1: Pareto front and Pareto point in the 2-D objective space.

The concept of Pareto optimality is coupled with the solution of multi-objective optimization, however in practical applications one may not always need to find Pareto optimal solution. Therefore, the concept of dominance is introduced. The concept of dominance is very similar to Pareto optimality, however, it is used more generally, and is specially useful in algorithms that try to find multiple Pareto optimal points at once. The concept of dominance is defined as follows.

Definition 4. *Dominance: A vector of objective functions in the objective space* $F(x) \in Z$ *is non-dominated iff there does not exist another vector* $F(y) \in Z$ *such that* $F(y) \leq F(x)$ *with at least one* $F_i(x) < F_i(y)$*. Otherwise,* $F(X)$ *is dominated.*

In multi-objective optimization one tries to optimize all the objective functions simultaneously, however often the objective functions are conflicting and when one function attains its minimum, the other functions have values higher than their

minimum. This has led us to define the point in the entire objective space where all the objectives reach their optimal value. This point is called the *utopia point* (also called as *ideal point*) and is defined as follows [106].

Definition 5. *Utopia point: A point in the objective space F^o is an utopia point iff* $f_i^o = \min_x \{ f_i(x) \mid x \in X \}$ *for each* $i = 1, ..., k$.

F^o is generally infeasible. However, many algorithms for finding the solution of the multi-objective optimization try to find a Pareto point as close as to the utopia point. Utopia point has also some applications in transforming the objective function into more useful form which we will discuss later in this section.

2.1.3 Multi-Objective Optimization Methods

The term "min" in the multi-objective optimization problem (2.1) indicates that we aim to minimize all the objective functions simultaneously, i.e. we want to find a point where every objective function achieves its optimum. If there is no conflict among the objective functions, then finding such a solution is possible. However, in general the objective functions are conflicting and there may not exist the best solution with respect to all the objectives. Hence, the solution to the problem (2.1) is usually a set of Pareto points. Thus, instead of being a unique solution to the problem, the solution to a multi-objective problem is a possibly an infinite set of Pareto points. In this case by attempting to enhance one of the objectives, other objectives will detriment as a result.

Having more than one objective function in an optimization problem results in additional degrees of freedom. Without constraining these degrees of freedom, instead of having a single optimal point, the solution of the optimization problem

would be a set of solution points. However, in practical applications we often have to decide between the solutions of the multi-objective problem and have only one of the Pareto points as the final solution. For choosing one point between the Pareto points we need to determine a preference criteria between solution points. Determining these preferences is done by a Decision Manager (DM) who is assumed to have in depth knowledge about the problem and is able to provide preference information related to the objectives and/or different solutions in some form.

The role of the DM can be articulated in different phases of the solution process. Accordingly, the method for solving the multi-objective optimization can be categorized into four classes based on responsibility of the DM in the solution process:

1. If DM specifies its preference information before obtaining the Pareto points the method is called *a priori* method [104].

2. If the information about the preference of DM is obtained after solving the multi-objective problem and presenting the Pareto solutions, the method is called *a posteriori* method [103].

3. When the DM has no-preference toward any of the Pareto points the method for solving the multi-objective optimization is called *no-preference* method [99].

4. If the DM interacts and changes/adopts its preference information when an iterative algorithm is used to find the Pareto points in the multi-objective problem, the method is called *interactive* method [105].

2.1.3.1 Methods with *a priori* Articulation of Preference

In the *a priori* methods, the user must provide preference information prior to obtaining the Pareto set. The user provides its preference in terms of goals or the

relative importance of different objectives. The solution process attempts to find a Pareto optimal solution satisfying the user preference as best as possible. This is a straightforward approach, however, its drawback is that the DM has to choose its preferences before knowing its options and possible solutions. The weighted global criterion method, weighted sum method, lexicographic method, weighted min-max method, exponential weighted criterion, weighted product method, and bounded objective function method are example of the *a priori* methods [104].

2.1.3.2 Methods with *a posteriori* Articulation of Preference

If a general representation of the set of Pareto points is generated and then the user is asked to select the most preferred one among them, the method is called *a posteriori* [103]. In some cases, it is hard for the DM to determine a clear approximation of its preference. Therefore, it is advantageous to allow the DM to choose from a general representation of the solutions and view the options before making a decision. In these methods instead of putting preference on the objective functions, the DM only considers which solution is most appealing. The DM can select the most appealing solution based on the objective function values in the criterion space or design variable values in the design space.

The weighted methods are common in generating a representation of the Pareto optimal set by solving multiple sequential optimization problems with different parameters. Other algorithms such as physical programming, normal boundary intersection method, normal constraint method, and genetic algorithms are also useful for *a posteriori* articulation of the DM preference. Regardless of the method that is used for generating the Pareto points, the solution must be presented to the DM in graphical or tabular form. Both of the graphical and tabular representations of a solution give an overview of different solutions to the DM. However, graphical presentation of

solutions is generally limited to three-dimensional representation of Pareto surface. Furthermore, in tabular representation of solution it may be hard to select the final solution if the number of objectives, variables, and the solution points are large.

Choosing the number of points to represent the Pareto set to the DM is an important factor in *a posteriori* methods. Selecting more solution points would provide a better representation of the Pareto set for the DM, however it requires additional computational time. Using fewer points for representing the Pareto set in tabular form makes it more user-friendly for the DM, however, it may result in an incomplete representation of the Pareto set and omitting possibly better solutions.

2.1.3.3 No - Preference Methods

In the no-preference method the DM has no favoritism toward any of the points in the solution set and takes no part in the solution process. In this case, the task is to find some unbiased solution without any preference information. In no-preference method the final solution is selected based on some assumption about a reasonable compromise between the objectives instead of asking the user for preference information [99]. In all the other classes, the user is assumed to take part in the solution process. Most of the no-preference methods are special cases of the *a priori* methods, where the preference parameters are set to be the same value for all objectives. Global criterion methods, Nash arbitration, and Rao's method are among the methods in this category [103].

2.1.3.4 Interactive Methods

In interactive methods [105], an iterative solution algorithm is formed and repeated. After each iteration, the current information and the current solution points are given to the DM and it is asked to provide preference information for the next iteration. Methods in this class gives the DM the possibility of specifying and adjusting

preferences information between iterations while at the same time the DM learns about the solution patterns in the problem as well as about ones own preferences.

2.1.4 Normalization of Objective Functions

In order to find the Pareto points the objective functions do not need to be transformed, however, it is often beneficial to transform the objective functions into more useful forms. Usually the objective functions have very different ranges for their values, thus this makes it hard for the DM to compare their values and put accurate significance on the objective functions. The goal of the function transformation is to assume that all the objective values take place within some comparable range. The function transformation is specially utilized in *a priori* methods that scalarize the objective functions. Here, we present some of most frequent function transformations within the multi-objective optimization framework.

One possible way to transform the objective functions is to normalize each function by its absolute highest value in the design space [107]:

$$f_i^t(x) \;=\; \frac{f_i(x)}{|f_i^{max}|} \tag{2.5}$$

where $|f_i^{max}|$ is the absolute maximum of $f_i(x)$ in the design space (note that it is assumed $f_i^{max} \neq 0$). This approach produces non-dimensional objective functions bounded between 1 and -1.

Another method for transforming the objective functions is by using the *utopia* point. This method calculates the relative deviation of each objective from its minimum value, that is

$$f_i^t(x) \;=\; \frac{f_i(x) - f_i^o}{|f_i^o|} \tag{2.6}$$

This method provides a non-dimensional objective function with the lower value of zero. However, the upper value for the transformed objective is unbounded [113].

A variation of the previous method is to divide each objective value by its minimum value which results in non-dimensional objective functions with a lower limit of one [108], that is

$$f_i^t(x) \;=\; \frac{f_i(x)}{f_i^o} \qquad\qquad f_i^o \;>\; 0. \tag{2.7}$$

The last method for objective function transformation that we introduce here is to normalize each objective function by using its minimum and maximum value. As a result the transferred objective functions have a range of zero and one [109], [110], [112], that is

$$f_i^t(x) \;=\; \frac{f_i(x) - f_i^o}{f_i^{max}(x) - f_i^o} \tag{2.8}$$

The *utopia* point is often approximated in these methods. Therefore, the accuracy of the range of the transferred objective function is dependent on the accuracy of the approximation.

Following the introduction to the general categories of multi-objective optimization methods we will introduce two methods that are selected for solving the multi-objective optimization problem in this thesis. In the following the weighted sum method and the genetic algorithm are presented.

2.1.5 Weighted Sum Method

To solve a multi-objective optimization problem, a standard method is to transform the problem into a single objective problem. This is done by assigning a weight to each objective function and adding all the objectives to form a new single objective function. This method is known as the weighted sum (WS) [111], [98], and it allows one to use a single objective optimization algorithm. However, its drawback is that the WS method can only find one point in the Pareto front for every set of weights. It also requires that the user decides on the importance of each objective function prior to determining the possible solutions. This method is used to convert the problem into a form where one objective function can be optimized by a single objective optimization solvers available.

In the weighted sum method, we solve the following problem

$$\begin{aligned} \min_{\vec{x}} \quad & \sum_{i=1}^{k} \omega_i \, f_i^t(x) \\ s.t. \quad & \\ & g_t(x) \leq 0 \\ & h_s(x) = 0 \\ & x_l \leq x \leq x_u \end{aligned} \tag{2.9}$$

where $\omega_i \geq 0$ for all $i = 1, ..., k$, and typically $\sum_{i=1}^{k} \omega_i = 1$.

The solution to the above minimization problem can be shown to be weakly Pareto optimal and, furthermore, Pareto optimal if $\omega_i > 0$ for all $i = 1, ..., k$ if the solution is unique [98]. It is advisable to normalize the objective functions so that their different magnitudes do not bias the method. The weighted sum method is often used as *a priori* method when the DM is asked to provide the information about the preference in terms of weight parameters. However, it can be utilized as *a posteriori*

method by calculating the solution points for different weights to depict the Pareto set.

It is important in multi-objective optimization problems that the solution method would be able to find all the Pareto points. In this respect, the WS method has a shortcoming in solving non-convex problems, where it can not find any Pareto point in the non-convex part of the Pareto front. The reason behind this is the fact that the weighted sum method is usually implemented as a convex combination of objectives (where the weights are non negative and their sum is one). Thus, it may happen that some Pareto optimal solutions of non-convex problems cannot be found no matter how the weights are selected. To address this shortcoming the authors in [115] have developed a variation of WS method that will find the Pareto points in the non-convex parts of the Pareto front. To find these Pareto points in each of the non-convex parts of the Pareto front, their method will add a constraint to the optimization problem that will restrict the feasible objective space to a region that only has that specific non-convex part of the Pareto front. This method is known as adaptive weighted sum (AWS). As shown in Figure 2.2, the regular WS method only finds the Pareto points in the boundary sections 1, 3 and 5, and it is incapable of finding any Pareto points in the non-convex parts of the boundaries 2 and 4. Figure 2.3 shows how the adaptive weighted sum method finds the Pareto points in the regions 2 and 4. For the region 2, the AWS method attempts to find the Pareto optimal points by restricting the objectives to be in the region 2. This can be done by adding two constraints to the optimization problem that restricts the first objective to be less than the value of the first objective at the point $p2$ and the second objective to be greater than the value of the second objective at the point $p1$.

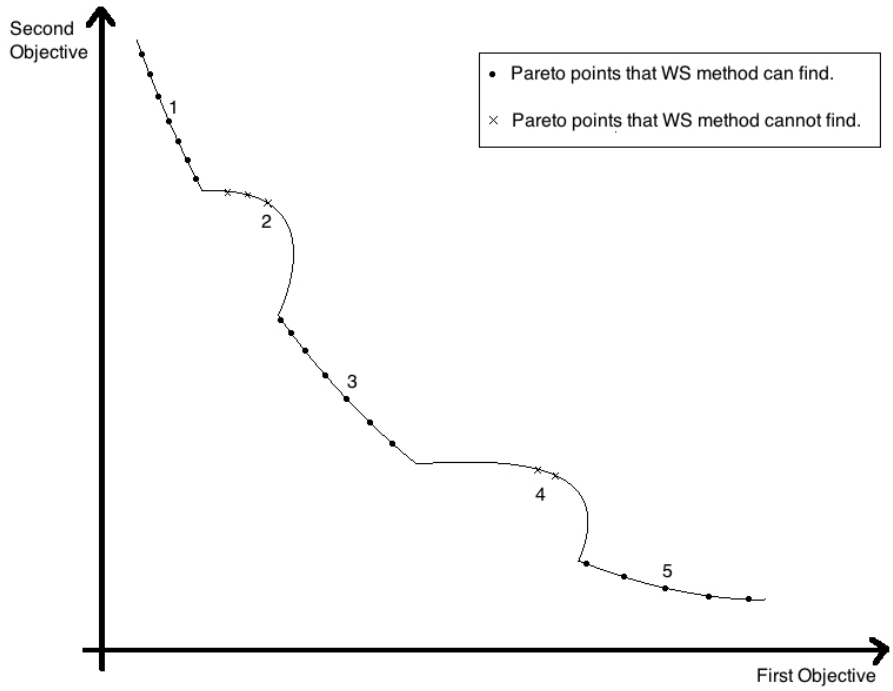

Figure 2.2: Boundary of the non-convex objective space. WS method can only find the Pareto points that are in the convex part of the boundary [115].

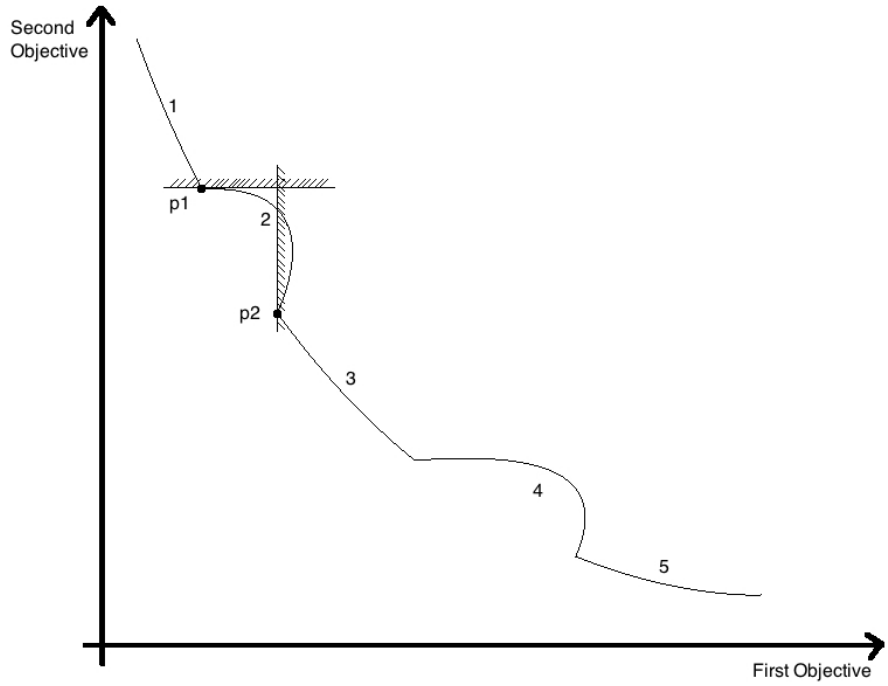

Figure 2.3: Example of how the AWS method finds the Pareto points in the non-convex part of the boundary [115].

2.1.6 Genetic Algorithm

Most of the methods in the *a priori*, *a posteriori*, and *no-preference* categories formulate the multi-objective optimization problem into the single objective optimization problem and then solve it using the standard optimization techniques. However, there are other approaches such as the *genetic algorithms* (GA) that are capable of solving the multi-objective problems directly. The genetic algorithm is introduced by Holland [117] and is used in many applications to solve optimization problems. Since the GA does not use the gradient information in solving the optimization problem its effectiveness is emphasized in problems that have non-smooth or non-differentiable objective functions and constraints. The basic idea behind the GA comes from the Darwin's theory of natural selection and uses the language of microbiology in process of producing the solutions.

Genetic algorithm starts the optimization with a population of decision vectors usually created at random within a specified lower and upper bounds on each variable. Afterwards, the GA performs an iterative operation of creating a new population from the current population. This iterative procedure consists of 4 main operators, namely: selection, crossover, mutation and elite-preservation. The GA stops the optimization when one or more termination criteria are met.

For solving the multi-objective optimization problems, GA attempts to give an idea on the extent and shape of the Pareto front by finding a set of well-distributed Pareto-optimal points. One must note that multi-objective genetic algorithm optimization principles are different from the classical optimization methodologies. GA is a direct search procedure that does not use gradient information in its search process. Thus, this allows it to be applied to a wide variety of optimization problems. Due to this reason, for a structured and well-behaved optimization problems it is better to use

a more standard approaches such as linear or quadratic programming as the GA has higher computational time and is not competitive with gradient-based optimization procedures.

GA is among the global optimization techniques; i.e. when GA is utilized to solve a single objective optimization it converges to the global solution in contrast to the gradient based methods that might find a local solution. The advantage of GA over other global optimization techniques for multi-objective optimization such as [118], [119], [120] is in its capability to work with a population of points. Most of the classical optimization algorithms update one solution in each iteration, however, the GA works with a population of points and uses more than one solution in each iteration which enables it to find multiple optimal solutions. This advantage makes GA very effective in finding the solution of multi-objective optimization problems. Another advantage of GA is that it has the ability to normalize objective functions within an evolving population using the population-best minimum and maximum values. The drawback of utilizing the GA is the memory and computational cost associated with performing each iteration. In this regard it is suitable for depicting the Pareto set for the DM, and thus, it is categorized as a method with *a posteriori* articulation of preference. The most popular method in the GA is the Fast and Elitist Nondominated Sorting Genetic Algorithm (also known as the NSGA-II) [121].

2.2 Map Partitioning

A commonly used resource allocation strategy for multi-agent systems is the partitioning policy. Partitioning policy is to divide the general task of mission into subtasks and then assign each agent to one of the subtasks. While other methods for resource allocation are available [122], [123], [124], [125], the partitioning policy

has the advantage of efficiency, ease of design and ease of analysis as a result of isolating the task of each agent from other agents. In search and coverage applications the partitioning policy comes in the form of dividing the area of interest, i.e. map partitioning. Map partitioning can be done by trivial methods that work in simple regions such as square or circular regions [31]. An alternative method for partitioning the region of interest is to decompose the map by small units such as squares or hexagons, and then assign a group of these units to each agent [16]. The method that we use for map partitioning in this thesis is by using the Voronoi and the power diagrams. These methods divide the region of interest into the exact number of agents in the mission. When using these methods *a priori* information about the area of each region is needed. In the following the basic concepts on the Voronoi and the power diagrams are presented and reviewed.

2.2.1 Voronoi Diagram

Voronoi diagram [127] partitions a plane into convex polygons by using some points as generators. Each of the resulting polygons contains exactly one generating point and the distance of each point in a given polygon to its generator is smaller than to any other generator. The concept of the Voronoi diagram was introduced by Descartes in 1644. Dirichlet also used it in 1850. However, it was Voronoi who studied and extended these concepts to higher dimensions. Voronoi diagram is also known as the Dirichlet tessellation [129]. The polygons are also called Dirichlet regions, Thiessen polytopes, or Voronoi polygons. The mathematical definition of the Voronoi diagram is as follows:

Definition 6. *Assume that* $G = (g_1, ..., g_m)$ *is a set of distinct points in the region* Ω. *The Voronoi diagram of* Ω *generated by the points* G, *denoted by* $V(G) =$

$\Big(V_1(G), ..., V_m(G)\Big)$, *is defined by*

$$V_i(G) \;=\; \{x \in \Omega \mid \|x - g_i\| \;\leq\; \|x - g_j\|,\; j \neq i\} \tag{2.10}$$

In other words, the Voronoi diagram divides the region Ω into m cells where if g_i is the nearest point in G to $x^* \in \Omega$ then x^* is in the Voronoi cell of g_i. Voronoi cells and their generators are shown in Fig. 2.4.

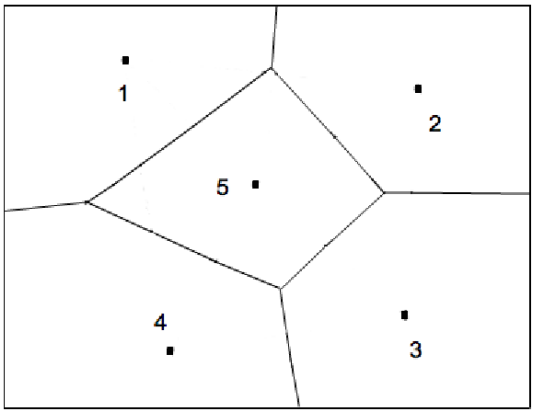

Figure 2.4: Example of a Voronoi diagram partitioning the 2-D space into five regions

One can easily show that each Voronoi cell is a convex set, and thus a Voronoi diagram of Ω is a convex partition of Ω. Two Voronoi cells are called neighbor if they have more than one points in their boundary that belongs to both cells. In Fig. 2.4 the Voronoi cells 1 and 2 are neighbors but the Voronoi cells 1 and 3 are not neighbors.

For $g_i, g_j \in G, j \neq i$, we define the bisector between g_i and g_j as

$$b(g_i, g_j) = \{\; x \in \Omega \mid \|x - g_i\| \;=\; \|x - g_j\| \;\} \tag{2.11}$$

The face $b(g_i, g_j)$ bisects the line segment joining g_i and g_j, and this line segment is orthogonal to the face (Perpendicular Bisector Property). The boundary of each Voronoi cell is either the boundary of Ω or the bisector of two generators.

The Voronoi diagram has found many applications in multi-agent systems. For example, it is used in optimal sensor allocation problem where the position of sensors are the generators of the Voronoi diagram. It is also used in the attack and destroy missions by multi-agent UAVs in presence of static radar sites, where the location of the radar sites are generators of the Voronoi diagram and the UAVs routes are the boundary of the Voronoi cells to avoid all of the radar sites equally.

2.2.2 Power Diagram

The only control element in Voronoi diagrams is the position of generators. If one wants to change the Voronoi partitions, the only solution is to change the location of the generators. This is a shortcoming in problems where the location of generators are fixed but the resulting Voronoi diagram are not satisfactory. Thus, a generalization of the Voronoi diagram is developed which is known as Power diagram. It adds weights to each generator which makes it possible to alter the Voronoi cells without changing the location of the generators. The mathematical definition of the power diagram is as follows:

Definition 7. *We refer to the pair* (g_i, ω_i) *as a power point and define*

$$G_W = \Big((g_1, \omega_1), ..., (g_m, \omega_m)\Big) \tag{2.12}$$

Assume that each point $g_i \in G$ *has assigned an individual weight* $\omega_i \in \Re$. *Let* $W = (\omega_1, ..., \omega_m)$. *We define the power distance as*

$$d_p(x, g_i, \omega_i) = \|x - g_i\|^2 - \omega_i \tag{2.13}$$

The power diagram of Q *generated by power points* G_W, $V(G_W) = \Big(V_1(G_W), ..., V_m(G_W)\Big)$, *is defined by*

$$V_i(G_W, \Omega) \quad = \quad \big\{x \in \Omega | \|x - g_i\|^2 - \omega_i \leq \|x - g_j\|^2 - \omega_j, j \neq i\big\} \qquad (2.14)$$

We refer to G_W as the set of power generators of $V(G_W)$, and $V_i(G_W)$ as the power cell or region of dominance of the i-th power generator. Moreover, we call g_i and ω_i the position and the weight of the power generator (g_i, ω_i), respectively. The power diagram is a convex partition of Ω. Note that when all the weights are the same the power diagram coincides with the Voronoi diagram.

A geometric interpretation of the power diagram for positive weights is as follows. A weighted point (g_i, w_i) is a circle with g_i as its center and w_i as its radius. For a point x outside this circle $d_p(x, g_i, \omega_i) > 0$, and it represents the square of the length of the tangent line from x to the circle. Example of a power diagram is shown in Figure 2.5.

One of the applications of power diagrams is in the map partitioning. In [12] the authors have shown that for any given convex region Ω, and any set of points G in Ω, there exists a set of weights W to partition the map with equal area in each partition.

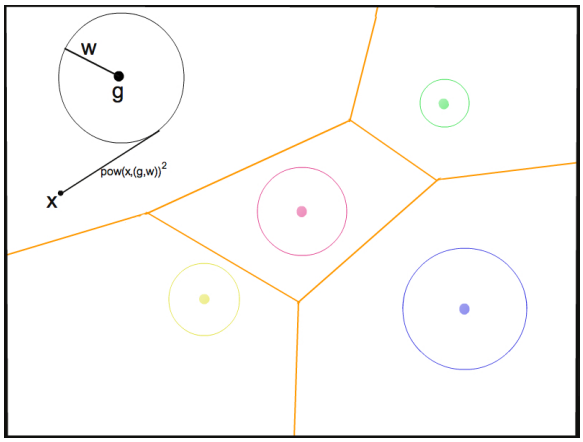

Figure 2.5: Example of a power diagram partitioning the 2-D space into five regions.

2.2.3 Conclusions

We have reviewed some basic concepts and methods on multi-objective optimization problem. The material that is reviewed in this chapter are used in Chapter 3 for solving the task assignment problem in coverage mission. As mentioned in the proposed solution section in Chapter 1, after solving the task assignment for a team of heterogeneous UAVs in the coverage mission, we partition the map to determine the share of each UAV on the map. We have introduced some basic concepts and methods on the space decomposition problem. Two partitioning methods are reviewed, namely the Voronoi diagram and the Power diagram, which divide the map into convex and disjoint subregions. The materials reviewed in this chapter are used in Chapter 4 for solving the map partitioning problem in coverage mission.

Chapter 3

Task Assignment in a Team of Heterogeneous UAVs

Mobile and static sensor networks have large number of applications for exploration, environmental monitoring, safety and recovery operations. Small low power mobile devices are expected to be the next generation of networks. Regardless of their field of application it is always important to maximize the field of coverage by these sensor networks. However, as a result of the size of sensors, these network agents have limited resource to communicate, compute, and move. Thus, power management is a key factor in developing these systems. Time management is also crucial in mobile sensor applications. When mobile sensor networks are utilized in safety and recovery operations, or in search and rescue missions the time is the most important factor in the mission. When the team agents have different capabilities, balancing the workload for each agent based on its capability significantly affects the mission time cost. However, limited work has considered both the power and the time management in heterogeneous multi-agent networks.

In this thesis we address the problem of dynamic coverage of an area of in-

terest by a team of heterogeneous UAVs. The UAVs are heterogeneous in the sense that they have different sensor footprints and different ranges of speeds. The team is tasked to visit a region of interest ($\Omega \subset \Re^2$) and perform a coverage search. A point is visited if it is inside the sensor footprint of an UAV at any time during the mission. Every point in the region of interest should be visited by at least one of the agents during the mission. The mathematical formulation of the problem includes the following components and constraints:

- N_v Heterogeneous UAVs: each vehicle has a range of forward speed $v_{imin} < v_i < v_{imax}$, where v_{imin} and v_{imax} denote the minimum and maximum speed of UAV_i, respectively.

- Coverage Region $\Omega \subset \Re^2$: is a compact and convex subset of $\Re^2$, and has the total area of D.

- Assigned Region Ω_i: is the UAV_i coverage region which is a subset of Ω, and has the area of d_i.

- Radius of Sensor Footprint r_i: each UAV is equipped with a sensor. The sensor footprint for UAV_i has a radius of r_i.

The solution methodology that is developed in this thesis divides the problem into 2 phases. In the first phase, the objective is to find the best group of UAVs that are able to finish the mission with the minimum time and fuel. This is done by performing a multi-objective optimization process. The decision variables of optimization problem are the speeds of the UAVs and their coverage area.

The result of the multi-objective task assignment optimization is a set of different possible assignments for the team. The user or the mission designer must

determine the favorite assignment based on the mission criteria and also past experiences. Since the UAVs are different and have do not have the same capability, some of the UAVs may not be included in the solution of the task assignment optimization problem. Choosing the final solution of the task assignment determines the optimal amount of coverage area for the UAVs, but their coverage region is not determined yet. The coverage region is determined through the map partitioning phase. After solving the task assignment optimization and choosing a solution as the preferred task, the map partitioning algorithm would divide the region of interest into subregions according to the mission designer priority set of task assignments. This phase assigns a subregion of Ω to each UAV that matches with the solution of the task assignment phase. The task assignment optimization phase is studied in this chapter, and the map partitioning is presented in Chapter 4.

This chapter is organized as follows. The mathematical formulation of mission objectives is presented in Section 3.1. The task assignment problem formulation and the solution methodology are presented in Sections 3.2 and 3.3, respectively. In Section 3.4, the simulation and results are presented. In Section 3.5, the chapter conclusion is provided.

3.1 Defining the Objective Functions

The first step in the multi-objective optimization problem is to determine all the objectives and constraints. In this thesis we are interested in search and coverage missions by a team of heterogeneous UAVs. The goal is to cover the entire region in minimum time, while minimizing the amount of fuel in the mission. Hence, there are two objectives in the mission: one is concerned with the cost of fuel (J_f), and the

other is addressing the time it takes for the team to perform the mission (J_t), that is

$$J_{team} = [J_f, J_t] \tag{3.1}$$

Due to the conflict between performing the mission in minimum time and using the minimum amount of fuel, one cannot find a set of task assignments that fully satisfies both of these objectives.

The following conditions are assumed for the surveillance mission:

(I) No point is visited more than once by one UAV.

(II) No point is visited by more than one UAV.

Condition (I) implies that each UAV investigates its region efficiently, i.e. it visits each point once. From this condition one can conclude that the area covered by the i^{th} agent can be expressed as the total length of its path multiplied by the diameter of its sensor footprint, that is $d_i = 2\, l_i\, r_i$, where d_i is the coverage area of UAV_i, l_i is the length of its path and r_i is its sensor's footprint radius.

While condition (I) focuses on the optimality of the coverage for each UAV individually, condition (II) addresses optimality of the team's assignment by making sure that UAV assignments do not overlap with one another. In other words, the sum of the coverage areas for all agents must be equal to the total area of the search space. By assuming the above conditions one can formulate an optimization problem with UAVs' speed and coverage area as its decision variables. In the following, the objectives are defined assuming these two conditions hold.

3.1.1 First Objective Function: Minimizing Fuel

The fuel consumption rate for each UAV is proportional to the aerodynamic drag force, which is proportional to the velocity squared. Hence, the fuel required by the i^{th} UAV to search its assigned region is a function of the vehicle's velocity and the length of the path covered by the vehicle through the mission. We denote the fuel objective for each UAV as J_{f_i}. The function J_f is the fuel cost of the team and it is equal to the sum of the fuel cost for each UAV. The individual fuel cost can be expressed as:

$$J_{f_i} = v_i^2 l_i \tag{3.2}$$

where:

v_i : The velocity of the i^{th} UAV constraint to be between $v_{imin} < v_i < v_{imax}$.

l_i : The length of the i^{th} UAV path.

Therefore, for the entire team the fuel cost function is

$$J_f = \sum_{i=1}^{N_v} J_{fi} \tag{3.3}$$

The length of the i^{th} UAV path (l_i) can be calculated based on its sensor radius and the total area of its coverage, that is

$$l_i = \frac{d_i}{2r_i} \tag{3.4}$$

where:

d_i : The area of region assigned to the i^{th} UAV.

r_i : The range of sensor footprint of the i^{th} UAV.

Therefore, the fuel cost for each UAV and the entire team are:

$$J_{f_i} = \frac{d_i v_i^2}{2r_i} \tag{3.5}$$

$$J_f = \sum_{i=1}^{N_v} \frac{d_i v_i^2}{2r_i} \tag{3.6}$$

3.1.2 Second Objective Function: Minimizing Time

The function J_t is the time cost incurred by the team of UAVs for carrying out its surveillance and coverage mission. A suitable time cost function must have the following criteria:

a) Two missions with different total time of coverage should have different time costs, and the mission with longer mission time should have higher cost.

b) Two missions with the same total time of coverage but different individual mission time should have different time costs, e.g. If there are two UAVs to perform the coverage mission, one prefers to finish the mission in 10 second (both UAVs take 10 second to finish their coverage) in comparison to one UAV finish the mission in 15 second and the other one finish the mission in 5 second.

One should note that the average UAVs completion time ($J_t = \sum_{i=1}^{N_v} \frac{t_i}{N_v}$) lacks the second criteria, while choosing the maximum mission completion time ($J_t = \max_{i=1}^{N_v} t_i$) lacks the first condition.

We will use the mean square of UAVs completion time for calculating J_t. The cost used for this purpose can be written as

$$J_t = \sum_{i=1}^{N_v} \frac{t_i^2}{N_v} \tag{3.7}$$

Each UAV's mission time, t_i, depends on the length of its path, l_i, and the speed of UAV, v_i, that is

$$t_i = \frac{l_i}{v_i} \tag{3.8}$$

The length of each UAV's path is a function of its assigned region's area and the radius of its sensor, namely

$$l_i = \frac{d_i}{2r_i} \tag{3.9}$$

$$t_i = \frac{d_i}{2r_i v_i} \tag{3.10}$$

Therefore, the time cost for the entire team is:

$$J_t = \sum_{i=1}^{N_v} \frac{d_i^2}{4N_v r_i^2 v_i^2} \tag{3.11}$$

3.2 Problem Statement

Following the definition of the objective functions we can now formulate the problem as follows:

Problem 3.1: *Given a team of N_v heterogeneous UAVs and a region Ω with a total area D, determine the best group of UAVs along with their optimal speeds and coverage areas that minimizes both the fuel and the time objective functions in the coverage mission.*

The Problem 3.1 can be stated as the following optimization problem:

$$\min_{\vec{v},\vec{d}} \quad \left[\sum_{i=1}^{N_v} \frac{d_i v_i^2}{2r_i} \, , \, \sum_{i=1}^{N_v} \frac{d_i^2}{4N_v r_i^2 v_i^2} \right] \tag{3.12}$$

subject to:

$$v_{i_{min}} \leq v_i \leq v_{i_{max}}$$
$$\sum_{i=1}^{N_v} d_i = D, \quad d_i \geq 0$$

where d_i is the coverage area of UAV_i, and $v_{i_{min}}$ and $v_{i_{max}}$ are the vectors containing the lower and the upper bounds of UAV's speeds, respectively.

From equation (3.12), one can note that manipulation of the velocity and coverage area determine the total cost of the mission. An increase in the velocity would result in an increase in the fuel cost but a decrease in the mission time. Similarly, an increase in the coverage area would result in increase in the distance covered by an UAV, which in turn increases its fuel and time cost, but this increase in the area of one vehicle would result in decrease of another UAV(s) area(s), therefore the cost of fuel and time of another vehicle would decrease.

3.3 Solution Methodology

The solution for Problem (3.1) is a set of Pareto points. For having one final solution a mission designer must choose between the set of Pareto points based on the importance of each cost function. For example, if the mission is search and rescue then minimizing mission time should have more importance than conserving fuel. However, when the mission is delivering a product, the mission cost is set in favor of the fuel cost. In choosing the methods for solving the optimization problem (3.12), we have selected the multi-objective genetic algorithm (GA) and the weighted sum (WS) methods.

Solving using the GA has the advantage of finding multiple solutions at once. The disadvantage of the GA is its long computational time in comparison to the WS method. Since in the GA the representation of the Pareto front is obtained there is no need to redo the optimization when the preference of the mission designer changes throughout different times.

There is no single solution to the optimization problem (3.12) that minimizes all the cost functions. Hence, the final solution must be chosen through involvement of a user/decision manager. The final solution can be selected according to the preference information given by the user. If the preference information is provided before obtaining any of the Pareto solutions, the method is called *a priori*, and it would search for the final solution without finding any other Pareto solution. On the other hand, the user may not be able to provide the preference information before evaluating the possible solutions. This usually happens when the user has not enough insight about the problem and possible solutions. In this case, *a posteriori* method is utilized to obtain a representation of the Pareto set for the user/decision manager. In *a posteriori* methods the final solution is no longer chosen based on the relative importance of the objectives, instead it is chosen directly by the decision manager by evaluating the solutions in the Pareto set.

The weighted-sum method is the most widely used *a priori* approach in solving multi-objective optimization problems. Its extensive usage is due to being easy to understand and simple to implement. Furthermore, the weights in this approach reflect the preference of the decision manager in its simplest form by putting relative importance among the objectives. The WS method is mostly used as *a priori* method, i.e., before obtaining the possible solutions of the problem, the decision manager is asked to provide the preference information regarding the relative importance of

objective functions. In this regard the WS method will find the optimal solution, which satisfies the decision managers preference.

The WS method could also be utilized as *a posteriori* method in which it is employed to find a representation of the Pareto points. This representation is presented to the decision manager, and s/he will choose the suitable Pareto solution as the final answer of the problem. If the problem is convex, the WS method is a suitable choice to depict the Pareto set. However, the WS method has a disadvantage in depicting the Pareto set in non-convex multi-objective optimization problems. The WS method cannot find the Pareto points that are lying in the non-convex part of the objective space. Since the time objective function (J_t) in problem (3.12) is non-convex, we will use another *a posteriori* method to depict the Pareto front.

In choosing the *a posteriori* methods for solving the above optimization problem we have selected the multi-objective genetic algorithms (GA) method [121]. Using the GA has the advantage of determining the Pareto front. Hence, there is no need to re-solve the optimization problem when the preference of mission designer changes at different mission times.

GA solves the optimization problem with the concept of domination and crowding distance. A solution a dominates solution b if all the objective values for a are better than corresponding objective values for b. At each iteration of the algorithm the population is split into multiple fronts. The first front contains individuals that are not dominated by any other individuals in the population. The individuals in the i^{th} front are only dominated by the individuals in the $1, 2, ..., i-1$ fronts. After sorting the population into different fronts, each individual is assigned a rank equal to its front, i.e. the individuals in the first front are assigned with rank 1, and the individuals

in the i^{th} front are assigned with rank i^{th}.

Fig. 3.1 shows an illustrative example of domination in the 2-D objective space. The circle individuals are not dominated by any individual, hence they form the first front and their rank is 1. The square individuals are only dominated by the individuals in the first front. They form the second front and have a rank 2. The third front consists of remaining two star individuals which have a rank 3. Note that individuals in the third (last) front are dominated by the first and second fronts.

Crowding distance measures how close an individual is to the other individuals with the same rank. A set of solutions with large average crowding distance indicates large diversity in the population.

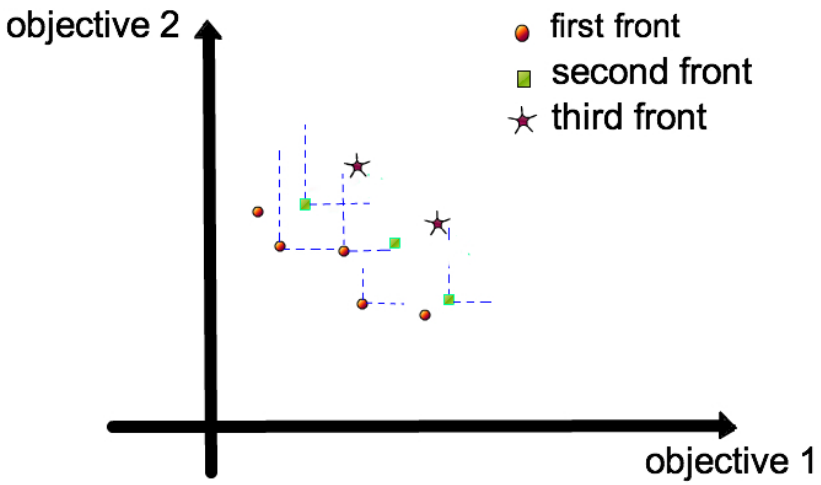

Figure 3.1: Grouping the individuals into different fronts [121].

Since GA does not work with the variables of the multi-objective problem directly, one must convert the decision variables into chromosomes with a binary code. Each

generation consists of N individuals that are represented in binary code as:

$$Ind_1 = \{110...0...1\} \tag{3.13}$$

$$Ind_2 = \{101...1...1\} \tag{3.14}$$

$$\vdots \tag{3.15}$$

$$Ind_N = \{011...0...1\} \tag{3.16}$$

The bit-string in each individual contains the values of the decision variables. To form a chromosome, the value of each decision variable is converted into a fixed length binary number and then the binary codes of the decision variables are concatenated. The constellation chromosome structure is shown in Figure 3.2.

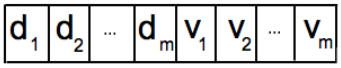

Figure 3.2: Data structure of the chromosomes used in the GA, where d_i s and v_i s denote the decision variables of the optimization problem.

After creating a random parent population (P_0) with N individuals, at each iteration of the GA we assign a rank to each individual equal to its front number. For selecting the pairs of individuals for reproduction, the tournament selection is used with the criterion based on the rank and the crowding distance, i.e. we prefer the individual with a lower rank, and in case of having equal ranks, the individual with higher crowding distance is preferred. After selecting the pairs of parents, the child set, Q_t, is produced by performing mutation and recombination [121]. The next step is to combine the sets of parent and child populations $R_t = P_t \cup Q_t$. The size of R_t is $2N$. The new parent population P_{t+1} is then formed by selecting the best N individuals among the population of R_t. The criterion in selecting the population

P_{t+1}, is the same as the criterion for the tournament selection. Furthermore, the proposed approach of Deb *et al.* in [121] is used for handling the constraints of the multi-objective optimization problem. Figure 3.3 shows the multi-objective genetic algorithms that is used in this chapter.

Initialize population with N individuals (P_0)
$i = 0$
while(stopping criteria has not met)
    Generate child from population(Q_i)
    Combine parent and child$(R_i = P_i \cup Q_i)$
    Rank R_i based on domination $F = (F_1, F_2, ...)$
    $P_{i+1} = \{\}$
    k=1
    while $(|P_{i+1}| + |F_k| < N)$
        $P_{i+1} = P_{i+1} \cup F_k$
        $k = k + 1$
    end
    sort F_k based on crowding distances
    $P_{i+1} = P_{i+1} \cup F_k\{1 : (N - |P_{i+1}|)\}$
    $i = i + 1$
end

Figure 3.3: The multi-objective genetic algorithm [121].

The solution of the multi-objective optimization problem is a set of Pareto points. The decision manager will choose the final solution based on its preference. We have modeled the preference of the decision manager in the form of relative importance of the objective functions. This can be accomplished by selecting the percentage of each objective's importance in the final decision, that is through

$$J_{total} = \omega_f J_f + \omega_t J_t \tag{3.17}$$

$$0 \leq \omega_f, \omega_t \leq 1$$

$$\omega_f + \omega_t = 1$$

In forming the total mission cost in equation (3.17), it is important to normalize the objective functions, so that both of the objectives have the same order of magnitude. There are various methods for normalization as presented in Section 2.1.4. After normalizing the objective functions, the final solution is the Pareto point which has the minimum J_{total} for the mission.

It should be noted that the task assignment optimization problem is only necessary in presence of heterogeneous UAVs. If the UAVs are homogenous (that is, $r_i = r_j,\ v_{i_{min}} = v_{j_{min}},\ v_{i_{max}} = v_{j_{max}}$), the optimal coverage area for all of the UAVs would be the same (i.e., $d = \frac{D}{m}$). Similarly, the optimal speed has to be the same for all the UAVs. Consequently, there would be only a single variable (v) and the multi-objective optimization problem reduces to:

$$\min_{v} \quad \left[\frac{Dv^2}{2r}\ ,\ \frac{D^2}{4mr^2v^2}\right] \tag{3.18}$$

$$s.t.$$

$$v_{min} \leq v \leq v_{max}$$

For any set of $v_{min} \leq v_1 < v_2 \leq v_{max}$ we can write:

$$\frac{Dv_1^2}{2r} \ < \ \frac{Dv_2^2}{2r} \quad \rightarrow \quad J_f(v_1) \ < \ J_f(v_2) \tag{3.19}$$

$$\frac{D^2}{4mr^2v_1^2} \ > \ \frac{D^2}{4mr^2v_2^2} \quad \rightarrow \quad J_t(v_1) \ > \ J_t(v_2) \tag{3.20}$$

As can be seen from the above equations, neither of the objective sets $[\, J_f(v_1)\,,\ J_t(v_1)\,]$ and $[\, J_f(v_2)\,,\ J_t(v_2)\,]$ dominates the other. Therefore, all the points in this multi-objective optimization are Pareto points, and the minimization phase is redundant.

In coverage scenarios involving heterogeneous team of UAVs, it is important

to balance each UAV's share based on its capabilities. The task assignment framework that is developed in this chapter assigns optimal tasks to the best group of UAVs that ensure the minimum cost for the mission. It also handles time and power critical coverage missions effectively. It balances the task of all the agents, by selecting the amount of UAVs' coverage areas based on their capabilities.

The methods that are selected for solving the task assignment provide flexibility for the decision manager. If the decision manager is aware of the relative importance of the objective functions, the weighted sum method can be utilized to directly find the optimal set of assignments. On the other hand, the decision manager may prefer to study the possible solutions before choosing the final set of task assignments. In this case the multi-objective genetic algorithms can be employed to calculate a representation of the solutions, and the decision manager can make its decision by comparing among the results of the GA. Hence, the optimal set of tasks could be assigned to the agents using both methods.

3.4 Simulation Results

In this section, we apply the methodology that is presented in this chapter to two coverage scenarios. In the first scenario four UAVs are tasked to cover an area of square of $20km * 20km$. For selecting the optimal speeds and coverage areas for each UAV, the user must choose a Pareto point based on the importance of the objective functions. We use the GA as *a posteriori* and the weighted sum method as *a priori* method for obtaining the Pareto front. The focus of the first scenario is on the effect of the heterogeneity of the UAVs on their assignments.

In the second scenario 10 UAVs are assigned for the same task. The multi-objective optimization is used to determine the optimal task assignment solution. Both the GA and WS methods are used to obtain the best group of UAVs along with the optimal tasks for the UAVs that cooperate in the mission. In this scenario, it will be demonstrated that using all the UAVs for the coverage mission would not result in the optimal solution in all scenarios. The characteristics of the UAVs corresponding both scenario are listed in Table 3.1.

Table 3.1: UAVs specifications for both scenarios.

Scenario No.	Speed Range	Radius of Sensor Footprint
# 1	$80\ m/s < v_1 < 100\ m/s$	$r_1 = 35m$
	$80\ m/s < v_2 < 120\ m/s$	$r_2 = 56m$
	$70\ m/s < v_3 < 150\ m/s$	$r_3 = 49m$
	$60\ m/s < v_4 < 140\ m/s$	$r_4 = 84m$
# 2	$80\ m/s < v_1 < 110\ m/s$	$r_1 = 63m$
	$70\ m/s < v_2 < 115\ m/s$	$r_2 = 56m$
	$90\ m/s < v_3 < 125\ m/s$	$r_3 = 49m$
	$75\ m/s < v_4 < 130\ m/s$	$r_4 = 70m$
	$85\ m/s < v_5 < 120\ m/s$	$r_5 = 77m$
	$50\ m/s < v_6 < 80\ m/s$	$r_6 = 42m$
	$60\ m/s < v_7 < 85\ m/s$	$r_7 = 63m$
	$70\ m/s < v_8 < 90\ m/s$	$r_8 = 77m$
	$90\ m/s < v_9 < 110\ m/s$	$r_9 = 28m$
	$95\ m/s < v_{10} < 120\ m/s$	$r_{10} = 28m$

3.4.1 Scenario 1: Surveying with Four UAVs

The mission scenario consists of four UAVs surveying a large region with the goal of minimizing the time and the fuel cost for the mission. The range of speed for UAV_1 is $80m/s < v_1 < 100m/s$, the radius of its sensor footprint is 35 m, $r_1 = 35m$, for UAV_2

the range of speed is $80m/s < v_2 < 120m/s$, and the radius of its sensor footprint is 56 m, $r_2 = 56m$, for UAV_3 the range of speed is $70m/s < v_2 < 150m/s$, and the radius of its sensor footprint is 84 m, $r_3 = 49m$, and for UAV_4 the range of speed is $60m/s < v_2 < 140m/s$, and the radius of its sensor is 84 m, $r_3 = 84m$. The total area of coverage is $400km^2$.

The optimization phase is solved by using the WS and GA methods. The GA method is utilized as *a posterior* method. Hence, it tries to compute the Pareto front in its most general and diverse form as it does not have any information on the preference of one objective over another. To obtain a detailed representation of the Pareto front we have selected 160 individuals as the initial number of parents population in the GA method. Therefore, the number of population in each generation is also 160. At the final iteration of the GA, some of the individuals might not be in the first front, so by selecting 160 individuals, we make sure that the first front has enough population to spread through the entire Pareto front. However, this number directly affects the computational time of the GA procedure. In our simulations, the number of individuals in the first front in the last iteration of GA method was 56. Figure 3.4 shows the first front of the GA solution.

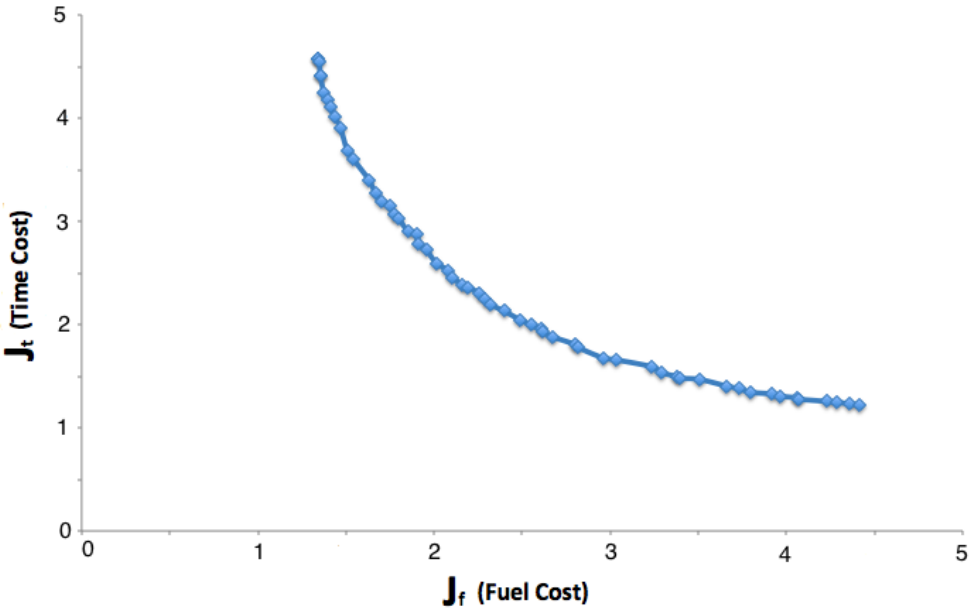

Figure 3.4: First front of the multi-objective genetic algorithms for Scenario 1. Both the fuel cost and the time cost are normalized by using equation (2.7).

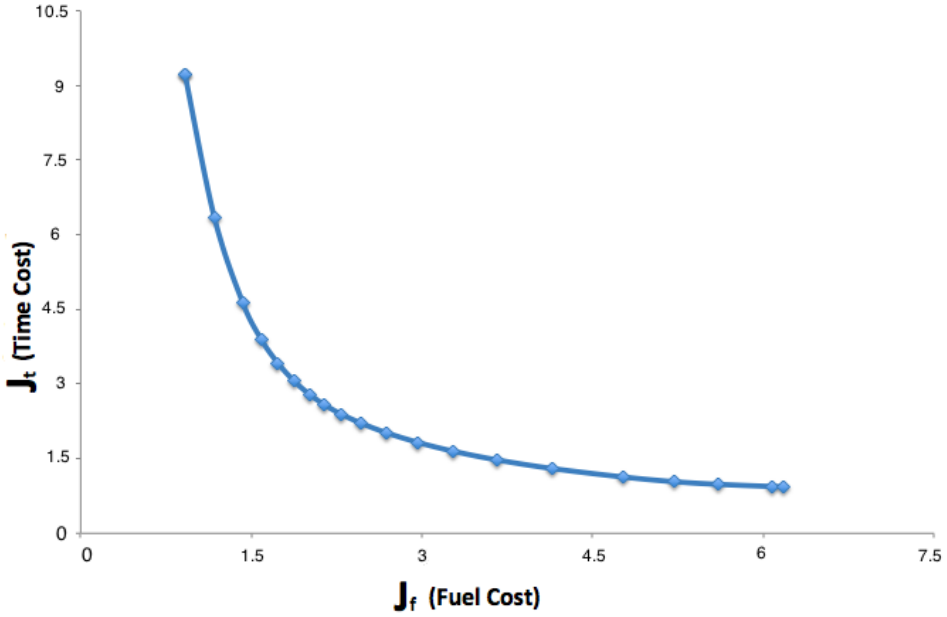

Figure 3.5: First front of the WS method for the Scenario 1. Both the fuel cost and the time cost are normalized by using equation (2.7).

The weighted sum method is also used to obtain the Pareto points for 21 different set of weights, where the weight for the fuel cost (ω_f in equation (3.17)) starts

as 1, while the weight for the time was set to 0, and the weight for the fuel decreases by 0.05, and hence the weight for the time increases by 0.05, in the next set of weights. Fig. 3.5 shows the first front of the WS solution. The corresponding values of the UAVs speeds and area of coverage for these points are shown in Table 3.2.

When the importance of the fuel is set to 1 (and accordingly importance of the time is 0), the entire map is assigned to UAV_4 which has the largest sensor footprint and the lowest minimum speed. However, when we incorporate the importance of the time in the mission cost, the optimal result requires cooperation of more than 1 UAV. The lowest share of the area is always assigned to UAV_1, because UAV_1 is the least capable between the four UAVs, due to the fact that its range of speed is a subset of the range of speed of all the other UAVs, and it has the smallest sensor radius.

To compare the results of the WS method in Fig. 3.5 and the GA method in Fig. 3.4, one can observe that both methods have depicted the Pareto front very well. The only difference between these two graphs is in the extreme cases where only one of the two objectives is important in the mission (the cases where either of ω_f or ω_t is zero). The WS method can easily find these Pareto points, however, the GA only search for the Pareto points where both objectives are important. As it can be seen in Fig. 3.5, the WS method can be exploited as *a posteriori* method in this scenario because it has found the Pareto front by only finding the solution of the problem to 21 set of different weights.

Figure 3.6 shows optimal coverage areas among the UAVs with different importance of the fuel and the time costs. One can see that by increasing the importance of the time in the objective of mission, the UAVs tend to cooperate more. The reason is for achieving the minimum time it is best that all the UAVs participate in the coverage

mission regardless of their fuel efficiency among the team. On the other hand, when the fuel is the main concern of the mission, the larger part of the area is given to the more efficient member of the team. Here, UAV_4 is more fuel efficient as the minimum speed of UAV_4 is less than other UAVs and its sensor has larger coverage as compared to the three other UAVs in the team, and when the importance of the fuel is set to 1, all of the region of interest is assigned to UAV_4. However, when the focus of the mission is to complete it as quickly as possible, the area assignment is based on the maximum speed of the UAVs, and the radius of the sensor footprint. In the first scenario the UAV_4 has the largest sensor footprint and it is the second fastest UAV among the team.

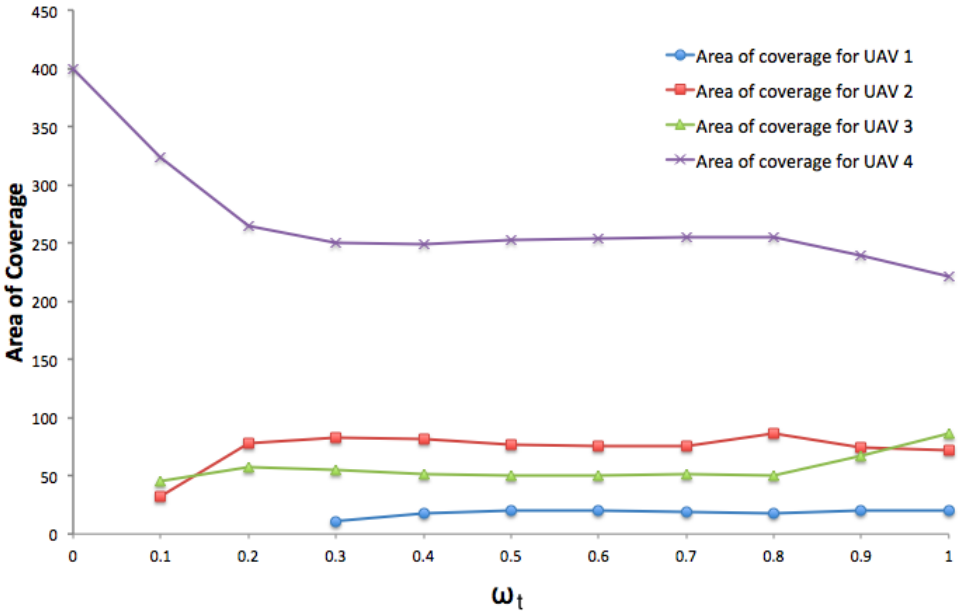

Figure 3.6: Changes in the area of coverage for each UAV corresponding to the changes in importance of the time cost.

Table 3.2: Optimization results obtained by using the WS method for Scenario 1

ω_t	ω_f	$d_1(km^2)$	$d_2(km^2)$	$d_3(km^2)$	$d_4(km^2)$	$V_1(m/s)$	$V_2(m/s)$	$V_3(m/s)$	$V_4(m/s)$
0.00	1.00	0.0	0.0	0.0	400.0	80	80	70	60
0.05	0.95	0.0	0.0	0.0	400.0	80	80	70	60
0.10	0.90	0.0	21.1	8.7	370.2	80	80	70	60
0.15	0.85	0.0	68.6	35.4	296.0	80	80	70	60
0.20	0.80	0.0	88.2	43.6	268.2	80	80	70	64
0.25	0.75	0.0	98.8	47.0	254.2	80	80	70	68
0.30	0.70	2.2	81.8	57.5	258.5	80	80	70	73
0.35	0.65	8.7	83.0	56.2	252.0	80	80	70	76
0.40	0.60	13.2	82.9	54.7	249.2	80	80	70	80
0.45	0.55	16.4	81.9	53.0	248.7	80	80	70	84
0.50	0.50	18.6	80.3	51.1	250.0	80	80	70	89
0.55	0.45	20.0	77.8	50.0	252.2	80	80	72	94
0.60	0.40	20.6	76.9	49.4	253.1	80	80	76	98
0.65	0.35	20.6	75.2	50.4	253.8	80	85	80	104
0.70	0.30	20.4	76.8	49.6	253.2	80	93	82	110
0.75	0.25	19.8	80.8	47.0	252.5	80	101	86	117
0.80	0.20	19.8	86.5	44.3	249.4	83	107	90	128
0.85	0.15	18.1	80.5	47.8	253.6	88	115	104	137
0.90	0.10	19.7	72.4	62.7	245.1	100	120	126	140
0.95	0.05	19.9	68.4	87.4	224.4	100	120	150	140
1.00	0.00	19.9	67.4	101.8	210.9	100	120	150	140

Figure 3.7 shows different optimal speeds among the UAVs with respect to different importance of the fuel and the time costs. By emphasizing on the mission time in the objective of the mission, the UAVs are assigned with faster speeds for coverage of their regions. Furthermore, when conserving fuel is the main objective of the mission, the UAVs use lower speeds in their mission. Note that all the UAVs use their maximum

speeds when the importance of time is set to 1 and importance of fuel is set to 0. When the importance of fuel is 1 the most fuel efficient UAV is assigned to cover the entire region with its minimum speed.

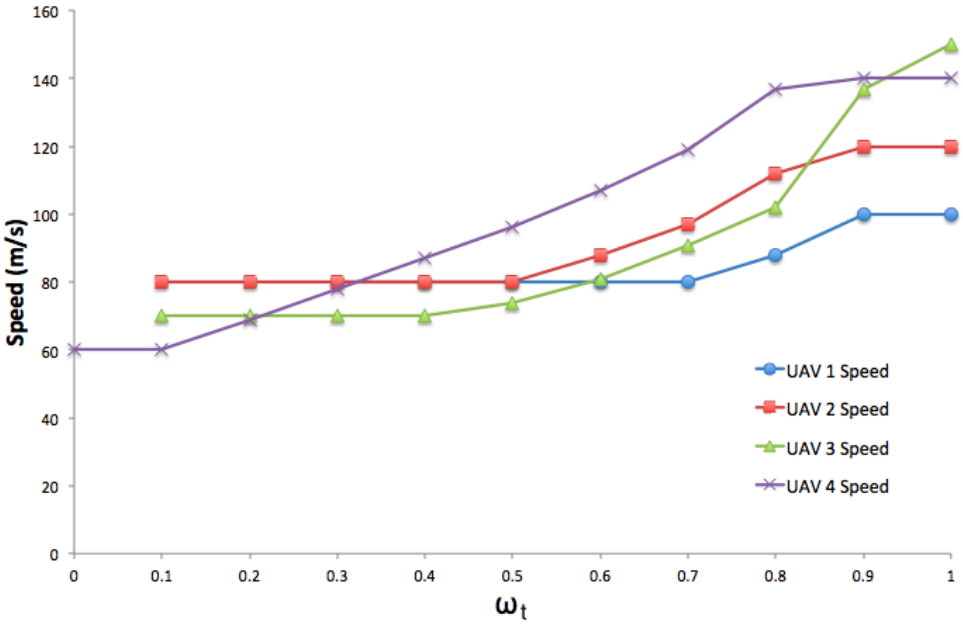

Figure 3.7: Variations in the optimal speed of each UAV corresponding to change in the importance of the time cost.

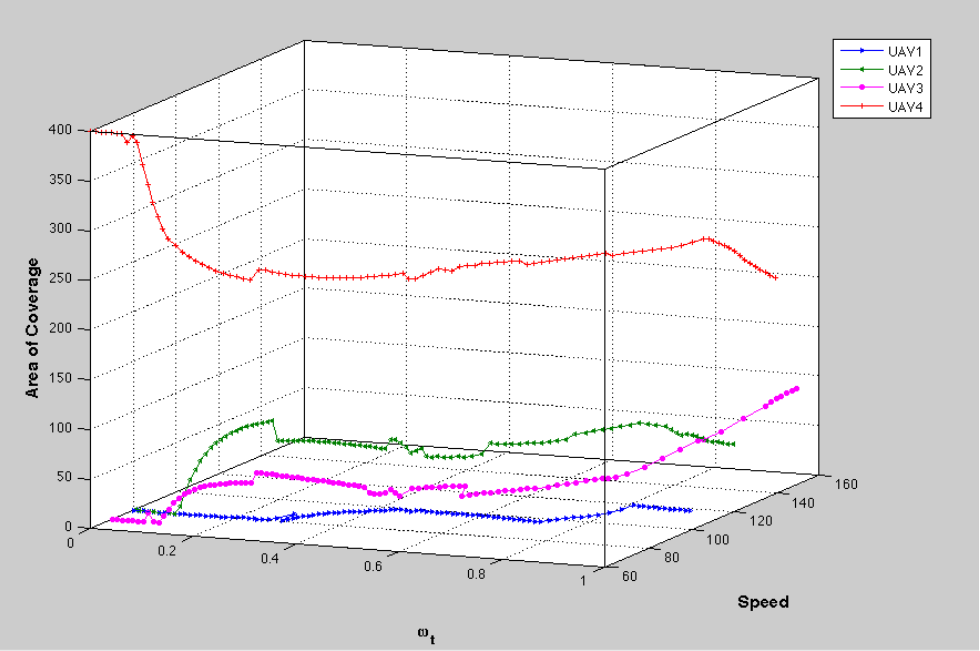

Figure 3.8: 3D plot of the optimization variables (speed and area of the coverage) with respect to different ω_t.

3.4.2 Scenario 2: Surveying with Ten UAVs

Ten heterogeneous UAVs with different sensor footprints and varying ranges of speeds are set to cover an area of square region of size $20Km * 20Km$. The first UAV is able to fly with a speed in the range of 80 to 110 m/s, and its sensor footprint has a radius of $63m$. The speed for the second UAV is limited between 70 and 115 m/s, and its sensor has a $56m$ radius footprint. The third UAV can fly with a varying speed between 90 and 125 m/s. The span of its sensor is $49m$. The fourth and fifth UAVs have a minimum speed of 75 and 85 m/s, respectively. Their maximum speed is also 130 and $120m/s$, and they have sensors with $70m$ and $77m$ footprint radius, respectively. The sixth UAV has the lowest minimum and maximum speed of 50 and 80 m/s, and its radius of sensor footprint is among the smallest one in the team with $42m$. The seventh UAV is able to fly with a speed in the range of 60 to 85 m/s, and its sensor footprint has a radius of $63m$. The eighth UAV is able to fly with a speed in the range of 70 to 90 m/s, and it has a sensor footprint with $63m$ radius. Both the

ninth and the tenth UAVs have the lowest radius of sensor footprint with only $28m$. The range of the speed for UAV_9 is 90 to 110 m/s, and it is 95 to 120 m/s for the UAV_{10}.

For this scenario we have solved the optimization phase only by using the WS method. The weighted sum method is used 21 times with different level of importance for each objective, with the weight for fuel cost (ω_f in equation (3.17)) starts as 1, while weight for the time was zero, and the weight for the fuel decreases by 0.05, and hence the weight for time increase by 0.05, in the next set of weights. Figure 3.9 shows the Pareto front of the WS solution. The corresponding values of the UAVs area and speed of coverage for these points are shown in Table 3.3 and Table 3.4, respectively.

Table 3.3: Optimal area of coverage that is obtained by using the WS method with different values of (ω_f, ω_t).

ω_f	ω_t	d_1 (km^2)	d_2 (km^2)	d_3 (km^2)	d_4 (km^2)	d_5 (km^2)	d_6 (km^2)	d_7 (km^2)	d_8 (km^2)	d_9 (km^2)	d_{10} (km^2)
0	1.00	48.2	41.7	38.2	80.2	85	11	26.9	48.3	9.7	10.8
0.05	0.95	49	41.3	35.4	78	86.9	11.7	30.1	52.9	7.3	7.4
0.1	0.90	49.9	39.4	27.5	76.2	91	12.6	33.9	58.6	5.4	5.5
0.15	0.85	50	35.7	23.6	70.6	93.9	13.6	38.2	64.4	5	4.9
0.2	0.80	48.3	33	24.7	67.3	90.7	14.5	41.8	71.4	4.5	3.9
0.25	0.75	47.4	33.1	24.6	65.3	87	14.3	44.9	77.4	3.5	2.5
0.3	0.70	47.1	32.9	24.7	65.1	84.9	13.8	46.3	82.5	2.2	0.5
0.35	0.65	47	32.7	24.2	64.9	86	14.1	47.1	83.4	0.6	0
0.4	0.60	49.4	32.3	22.1	64.4	87.5	13.8	47.2	83.3	0	0
0.45	0.55	49.6	33.4	20.9	64.1	89.5	13.5	46	83	0	0
0.5	0.50	50.6	34.4	16.8	65.7	90.1	13.4	45.8	83.1	0	0
0.55	0.45	50.6	35.7	12.8	67.8	91.8	13.3	44.7	83.5	0	0
0.6	0.40	50.3	36.7	7	69.9	91	13.5	45.2	86.4	0	0
0.65	0.35	49.7	35.2	0	72.6	89	14.3	46.6	92.5	0	0
0.7	0.30	47.4	34.5	0	73	84.1	15.2	49.8	96.1	0	0
0.75	0.25	42.7	33	0	73.6	79.6	16.3	53.8	101.1	0	0
0.8	0.20	35	29.9	0	74.1	73.4	18	60.5	109	0	0
0.85	0.15	19.9	25.1	0	75.2	62.5	21.3	73.1	122.8	0	0
0.9	0.10	0	18.1	0	74.5	34.2	26.5	91.3	155.4	0	0
0.95	0.05	0	0	0	38.1	0	35.6	127.4	199	0	0
1.00	0.00	0	0	0	0	0	146.2	253.8	0	0	0

Table 3.4: Optimal speed of the UAVs that is obtained by using the WS method with different values of (ω_f, ω_t).

ω_f	ω_t	V_1 (m/s)	V_2 (m/s)	V_3 (m/s)	V_4 (m/s)	V_5 (m/s)	V_6 (m/s)	V_7 (m/s)	V_8 (m/s)	V_9 (m/s)	V_{10} (m/s)
0	1.00	110	115	125	130	120	80	85	90	110	120
0.05	0.95	110	115	125	130	120	80	85	90	107	108
0.1	0.9	110	113	113	130	120	80	85	90	90	95
0.15	0.85	106	101	95	113	118	80	85	90	90	95
0.2	0.8	96	91	90	103	108	79	85	90	90	95
0.25	0.75	89	84	90	95	99	74	85	90	90	95
0.3	0.7	85	79	90	89	93	69	84	90	90	96
0.35	0.65	80	75	90	84	88	66	80	87	90	0
0.4	0.6	80	70	90	81	85	63	76	82	0	0
0.45	0.55	80	70	90	75	85	58	71	78	0	0
0.5	0.5	80	70	90	75	85	55	67	75	0	0
0.55	0.45	80	70	90	75	85	52	64	71	0	0
0.6	0.4	80	70	90	75	85	50	61	70	0	0
0.65	0.35	80	70	0	75	85	50	60	70	0	0
0.7	0.3	80	70	0	75	85	50	60	70	0	0
0.75	0.25	80	70	0	75	85	50	60	70	0	0
0.8	0.2	80	70	0	75	85	50	60	70	0	0
0.85	0.15	80	70	0	75	85	50	60	70	0	0
0.9	0.1	0	70	0	75	85	50	60	70	0	0
0.95	0.05	0	0	0	75	0	50	60	70	0	0
1.00	0	0	0	0	0	0	50	60	0	0	0

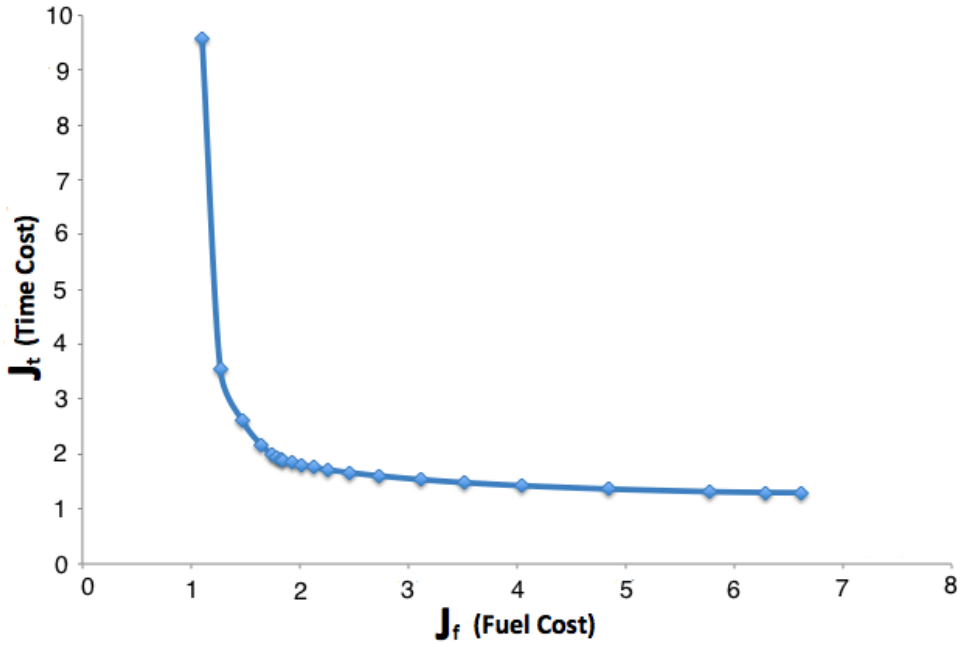

Figure 3.9: Pareto front that is obtained by the WS method for the Scenario 2. Both the fuel cost and the time cost are normalized by using equation (2.7).

When the importance of the fuel is set to 1 (and accordingly the importance of time is zero), the entire map is assigned to UAV_6 and UAV_7 which have the lowest minimum speed and have comparably good sensor footprint. However, when we incorporate the importance of the time in the mission cost, the optimal result requires cooperation of more UAVs. Figure 3.10 shows different optimal divisions of area of the interest among the UAVs with different importance of the fuel and the time costs. One can see that by increasing the importance of the time in the objective of the mission, the UAVs tend to cooperate more. The reason for getting the minimum time is that all the UAVs would participate in the coverage mission regardless of their fuel efficiency among the team. On the other hand, when the fuel is the main concern of the mission, the larger part of the area is given to the more efficient member of the team. Here, UAV_6 and UAV_7 are more fuel efficient as their minimum speed is less than the other UAVs. Accordingly, when the importance of the fuel is set to 1, all of the region of interest is assigned to UAV_6 and UAV_7. However, when the focus of the

mission is to complete it as quickly as possible, the area assignment is based on the maximum speed of the UAVs, and the radius of their sensor footprints.

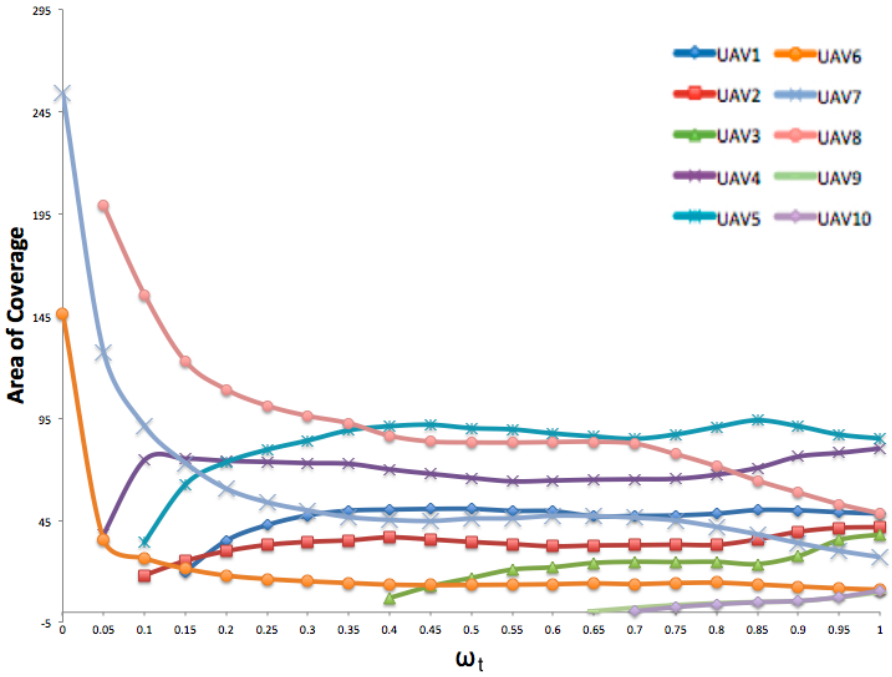

Figure 3.10: Variations in the area of coverage for each UAV corresponding to changes in the importance of the time cost.

Figure 3.11 shows different optimal speeds among the UAVs regarding the different importance levels of the fuel and the time costs. By emphasizing on the finishing time in the objective of the mission, the UAVs would be assigned with faster speed for coverage of their regions. Furthermore, when conserving fuel is the main objective of the mission, the UAVs use lower speed in their mission. Note that all the UAVs use their maximum speed when the importance of time is set to 1 and the importance of fuel is zero. When the importance of fuel is 1 the most fuel efficient UAV is assigned to cover the entire region with its minimum speed.

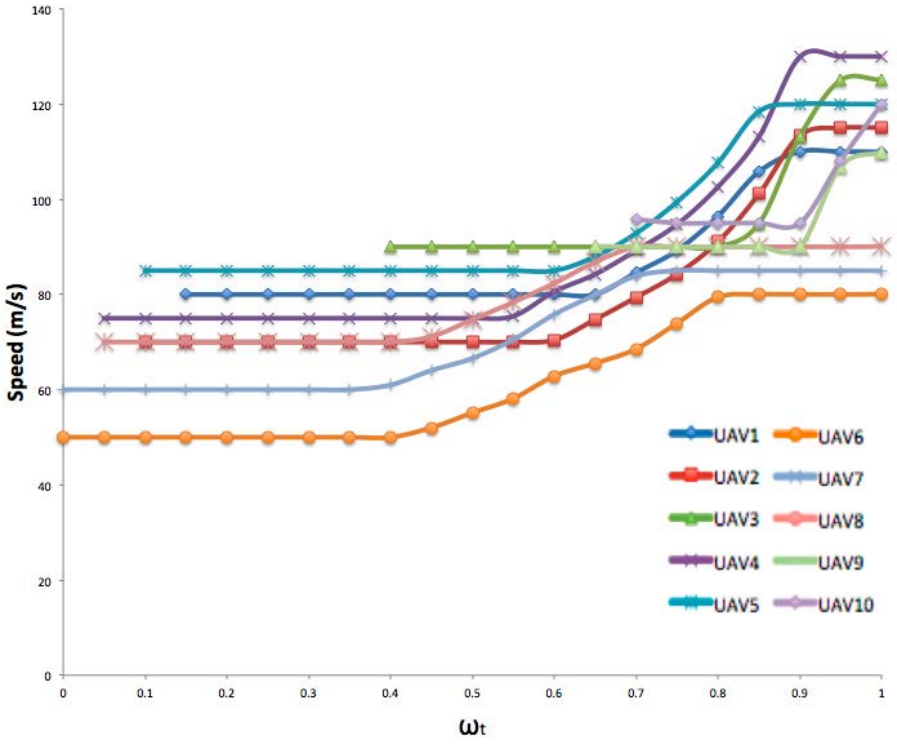

Figure 3.11: Variations in the optimal speed of each UAV corresponding to the changes in the importance of the time cost.

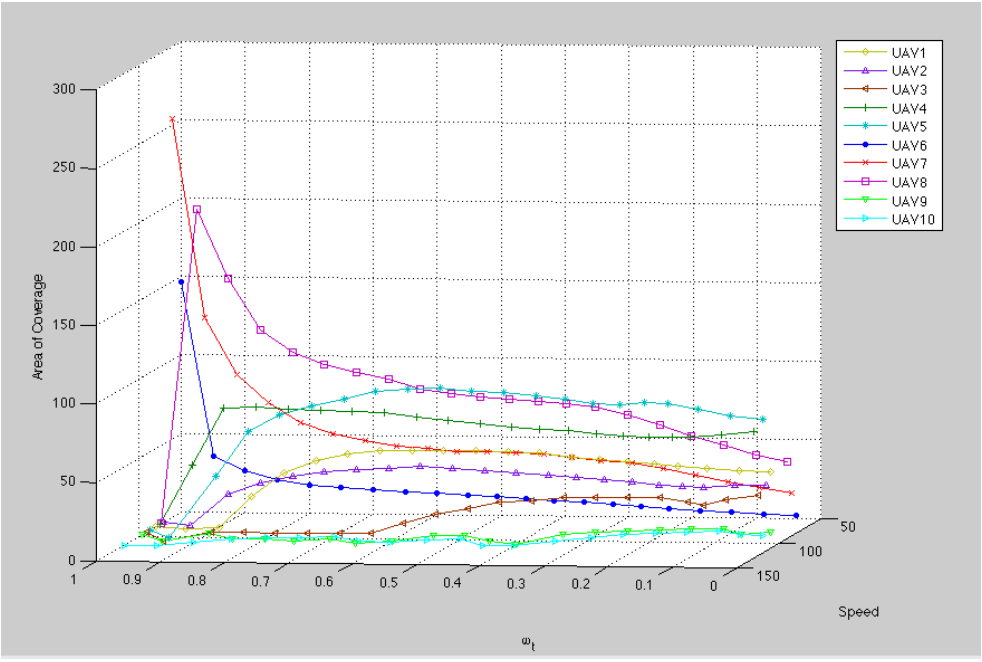

Figure 3.12: 3D plot of optimization variables (speed and area of the coverage) with respect to different ω_t.

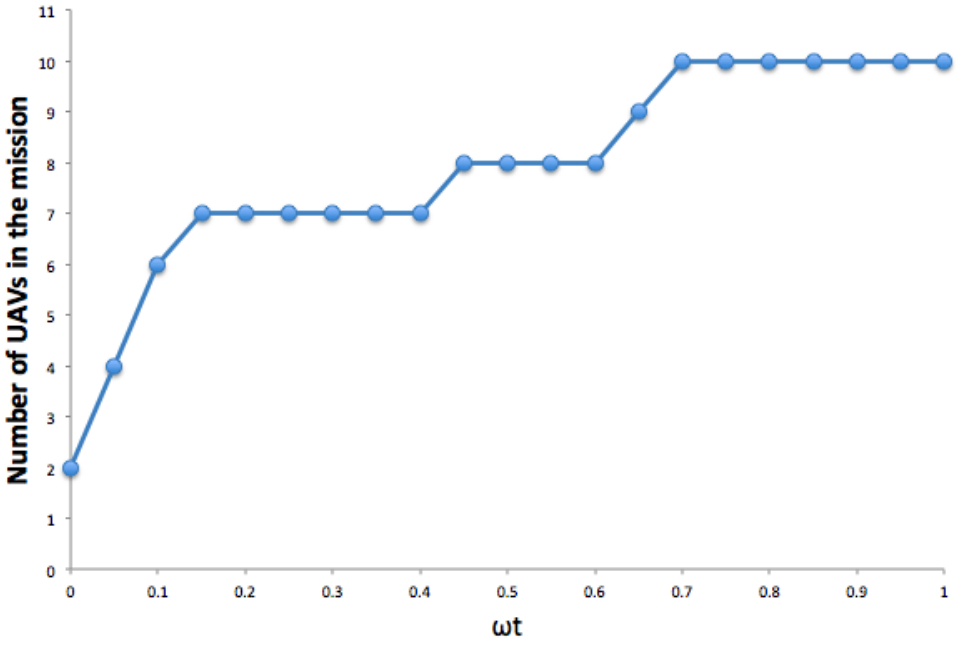

Figure 3.13: Number of the agents participating in the coverage mission for getting optimal result with respect to the priority of the fuel and importance of the time.

Figure 3.13 shows the optimal number of the UAVs for different priorities of the two objectives in the coverage mission. As the priority of time increases, more UAVs should participate in the mission in order to complete the coverage sooner. However, when the time is not the main concern in the mission, only the UAVs with higher fuel efficiency should do the coverage. It is interesting to note that if $\omega_t < 0.7$ there is no need to utilize all the ten available UAVs for carrying out the coverage. Note that this criterion is determined by the capabilities of the UAVs. If the UAVs have the same capabilities then there is no difference between them and for getting the optimal result one needs to always utilize all the UAVs in the mission. However, when the UAVs are heterogeneous, their differences results in performing the mission with the best suitable group among them. The UAVs with small radius of sensor footprint and large minimum of speed have higher fuel cost for covering a region in comparison with UAVs with large radius of sensor footprint and smaller minimum of speed. Hence, when the importance of conserving fuel is more than the importance of completing the

mission as fast as possible, it is likely that fuel efficient UAVs will handle the mission.

3.5 Conclusions

In this chapter, the problems of determining the best group of UAVs (in a team of heterogeneous UAVs) and finding their optimal speeds and coverage areas have been studied. A team of heterogeneous UAVs must cover a region of interest while the mission time and fuel consumption of the team should be minimized. Multi-objective optimization methods are used for solving the problem. The multi-objective genetic algorithm is used as *a posteriori* method to obtain the Pareto front when the importance of each objective is unknown. The weighted sum method is employed as *a priori* method. The methods in this chapter are employed on two example scenarios involving a team of heterogeneous UAVs, which must cover an square area of $20km * 20km$ obtaining the minimum mission cost for different importance levels of the fuel cost and the time cost.

Two cases are simulated. One is the mission with 4 UAVs in which the effects of the heterogeneity are studied on the area of each UAV' coverage mission. The collaboration among the team members is associated with the minimum combined time and the fuel cost of the team (team cost). In the second case, 10 UAVs are available for carrying out the coverage mission. In this case, it is demonstrated that for most sets of the fuel and the time importance levels the best team of agents does not include all the available UAVs. This is solely for the heterogeneous teams of UAVs, as in the homogeneous teams of agents, the team cost is always minimum when employing all the agents in the mission.

Chapter 4

Map Partitioning

Capabilities of UAVs are important elements in deciding how to allocate the tasks between individual UAVs. The solution strategies for dynamic coverage are mostly developed for homogeneous agents and they cannot be extended for heterogeneous team of agents. Given that these algorithms use sensors footprint radius or the speed of the homogenous UAVs as an important component and information of the algorithm, they are often not applicable (or not optimal) in missions that involve heterogeneous UAVs. In this chapter we address the problem of map partitioning for a team of heterogeneous UAVs. Two solution methods are presented for solving area specified map partitioning problem. The presented strategies along with the materials in Chapter 3, make a suitable framework for optimal coverage of a map with a heterogeneous team of agents.

This chapter is organized as follows. The problem formulation is presented in Section 4.1. A decentralized algorithm is introduced in Section 4.2 which uses the power diagrams to partition the map. In Section 4.3 the optimization phase in Chapter 3 is exploited in parallel with the decentralized map partitioning algorithm. A second approach for map partitioning is introduced in Section 4.4 which divides the map into area specific regions which causes the UAVs to make the minimum

number of turns during the mission. The simulations and results are presented and the performance of two methods are compared in Section 4.6. The chapter conclusion is also provided in Section 4.7.

4.1 Problem Formulation

After solving the multi-objective optimization and selecting one of the Pareto points as the final solution for the optimization phase, each UAV is assigned with an optimal speed and area of coverage. The next step is to divide the surveillance region Ω into m subregions $[\Omega_1, ..., \Omega_m]$ with areas equal to $[d_1, ..., d_m]$ where d_i $(i = 1, ..., m)$ are the final results of the optimization phase obtained in Chapter 3.

We are using partitioning method to assign each UAV a region with a per-specified area. The area of each region is considered as the workload of its corresponding agent. In this respect the partitioning method can be viewed as a workload sharing policy. The total workload for the team (that is the total area of the region of interest) must be divided between the UAVs in the mission. This is done through partitioning the region of interest $\Omega \subset \Re^2$ into m disjoint subregion Ω_i $(i \in [1, ..., m])$ whose union is Ω, where m is the number of agents.

As mentioned above in the coverage mission the workload of any region is its area. Hence, we can model the workload function $\Lambda : \Omega \rightarrow \Re$ that returns the area of a region, that is

$$\Lambda(\Omega) \quad = \quad \int_{\Omega} dx \tag{4.1}$$

The mathematical formulation of the problem is as follows:

Problem 4.1:

Having a positive vector $\vec{d} = [d_1, d_2, ..., d_m]$ *and a convex region* Ω *with a total area equal to* D ($\Lambda(\Omega) = D = \sum_1^m d_i$)*, divide the region into* m *disjoint convex subregions* $\Omega_i,\ i \in [1, ..., m]$ *with* $\bigcup_{i=1}^m \Omega_i = \Omega$ *and* $\Lambda(\Omega_i) = d_i$.

Note that in the multi-objective optimization problem there might be cases where some UAVs have been assigned with an area of coverage equal to zero. For handling this case we omit all the UAVs with zero area of coverage. The map partitioning considers only the positive values for the area of coverage.

4.2 Map Partitioning Using Distance Functions

In the previous chapter, a multi-objective optimization was performed to select the best team of UAVs for covering the map which results into least total cost of the mission. As a result, the participating UAVs in the mission were assigned with the optimal speed and the area of coverage. The next step is to partition the map and divide it into subregions according to the results of multi-objective optimization. There are a number of choices in selecting the method for the map partitioning. The required criteria for an applicable method in our case is the capability to partition the map into subregions with specified area. It is also desired that the shape of sub-regions to be convex, as large number of single search path planning methods are suitable for convex regions. One should note that as long as the map partitioning method is capable of dividing the map into disjoint areas with specific areas (according to the results of task assignment optimization) the optimality of the mission is guaranteed.

The Voronoi and the power diagrams are natural candidates for the partitioning Problem 4.1 due to their ability to partition the map into disjoint convex subregions. The researcher have used both of these two methods in multi-agent systems in different missions, such as the rendezvous, sensor placement, and attack and destroy missions. These two partitioning methods can also be used for partitioning the map into area specified regions. For every convex map Ω and for every positive vector $\vec{d} = [d_1, d_2, ..., d_m]$ with the condition of $\sum d_i = \Lambda(\Omega)$, there is a set of generators that divide the map into subregions with desired area. The partitioning can be also performed by power diagrams. In the following we review these methods briefly and compare the advantages of each method to choose the proper one for our scenario.

4.2.1 Voronoi Diagram

When a Voronoi diagram is used for partitioning a map the subregions are decided solely on the position of the Voronoi generators. One must change the position of the Voronoi generators in order to change the subregions, and consequently the areas of subregions. Hence, if we start from an initial locations for the generators we have to change the position of generators to get the desired partitioning. This can be formulated as an optimization algorithm where the optimization variables are the generator's positions.

For a set of generators position $G = (g_1, ..., g_m)$, the Voronoi diagram partitions the map into m subregions. Consider the function $\Psi : (\Omega, G) \rightarrow \Re$ that computes the mean square of the desired area for each generator over the actual area resulting from the current position of generators. Hence, $\Psi(\Omega, G)$ is defined as

$$\Psi(\Omega, G) = \sum_{i=1}^{m} \frac{d_i^2}{m\Lambda(V_i(G))} \tag{4.2}$$

where $G = \{g_1, ..., g_m\}, g_i \in \Omega, g_i \neq g_j$.

The Ψ assigns a positive value to every Voronoi partition. However, it gets its minimum value in special case. This is stated in the following lemma.

Lemma 1: The global minimum of $\Psi(\Omega, G)$ is obtained when $\Lambda(V_i(G)) = d_i$.

Proof. The proof is straightforward by noting that $\sum_{i=1}^{m} \Lambda(V_i(G)) = \sum_{i=1}^{m} d_i = D$ [12]. □

From Lemma 1, one can conclude that $\Psi(\Omega, G)$ obtains its minimum value only in the case of partitioning the map with the conditions in the Problem 4.1. Therefore, we can formulate a minimization problem on the value of Ψ function to find the location of the Voronoi generators that produce suitable partitions. Note that in order to change the value of $\Psi(\Omega, G)$ one must change the position of the Voronoi generators. Thus, the decision variables in the optimization problem should be the generators position

$$\min_{G} \quad \Psi(\Omega, G) \tag{4.3}$$

where $G = \{g_1, ..., g_m\}$ and $g_i \in \Omega$.

In the above optimization the decision variables are the generators location, and during the optimization the generators are relocated to find the position that results into the solution of the problem. The result of the above optimization problem generates a Voronoi partition where $V_i(G) = \Omega_i$, and $\Lambda(V_i(G)) = \Lambda(\Omega_i) = d_i$. Thus, it is a solution for map partitioning Problem 4.1.

An interesting solution strategy is proposed in [79] where a decentralized al-

gorithm is used to obtain the desired Voronoi generators which partition the map into regions with areas equal to $(d_1, d_2, ..., d_m)$.

4.2.2 Power Diagram

The other option for map partitioning is to use power diagrams. In using a Voronoi diagram for every set of generators there is only a single partitioned map as a solution. However, in Power diagrams due to having a degree of freedom in the weights of the power diagram, one can generate unlimited different partitions in a map by changing the weights of the power diagram and without changing the location of the generators. Hence, the advantage of using the power diagram over Voronoi diagram is that one can choose between the generators position and their associated weights to change the subregions and consequently the area of the subregions.

For a set of power generators $G_W = \{(g_1, \omega_1), ..., (g_m, \omega_m)\}$, the power diagram $V(G_W, \Omega)$ partitions the map into m subregions ($\{V_1(G_W, \Omega), ..., V_m(G_W, \Omega)\}$ or $\{\Omega_1, ..., \Omega_m\}$). Consider the function $\Gamma : (\Omega, G_W) \rightarrow \Re$ that yields the mean square of the desired area for each generator over the actual area resulting from the power diagram. Hence, $\Gamma(\Omega, G_W)$ is defined as

$$\Gamma(\Omega, G_W) = \sum_{i=1}^{m} \frac{d_i^2}{m\Lambda(V_i(G_W, \Omega))} \tag{4.4}$$

where $G_W = \{(g_1, \omega_1), ..., (g_m, \omega_m)\}$, $g_i \in \Omega, g_i \neq g_j$, and $V_i(G_W, \Omega)$ is the power diagram partitions of Ω.

Therfore Γ assigns a positive value to every power diagram partition. However, its minimum value is a special case. This is stated in the following lemma.

Lemma 2: The global minimum of $\Gamma(\Omega, G_W)$ is obtained when $\Lambda(V_i(G_W, \Omega)) = d_i$.

Proof. The proof is straightforward by noting that $\sum_{i=1}^{m} \Lambda(V_i(G_W, \Omega)) = \sum_{i=1}^{m} d_i$ [12]. □

One may note that $\Gamma(\Omega, G_W)$ obtains its minimum value only in the case of partitioning the map with the conditions of the Problem 4.1. Therefore, one can formulate a minimization problem on the value of the Γ function to find the proper values for the set of power generators (G_W). In order to change the value of $\Gamma(\Omega, G_W)$ one may change the position of power generators or the values of their associated weights. Therefore, if one formulates the optimization problem to minimize the value of the Γ function, the decision variables are the generators position and their weights. Specifically, we have

$$\min_{W} \quad \Gamma(\Omega, G_W) \tag{4.5}$$

where $G = \{g_1, ..., g_m\},\ g_i \in \Omega\ ,\ g_i \neq g_j$ and $W = (\omega_1, ..., \omega_m),\ \omega_i \in \Re$.

Thus, the result of the above optimization problem would generate the proper power diagram that satisfies the conditions for solution of Problem 4.1.

The power diagram can partition the map into regions with pre-specified areas only by changing the weights. This allows one to change the generators' positions during the process of the map partitioning. Having this degree of freedom is very useful in satisfying other constraints of the desired solution such as the shapes, and the approximate positions.

The power diagram $V(G_{W^*}, \Omega)$ is a solution to the optimization problem (4.5)

if $\Gamma(G_{W^*}, \Omega) = \sum_{i=1}^{m} d_i = D$, representing the global minimum of $\Gamma(G_W, \Omega)$. The result of the optimization Problem 4.5 is a suitable set of power weights $W^* = \{\omega_1^*, ..., \omega_m^*\}$ for our map partitioning problem.

For large sets of partitions the high dimensionality of the parameter vector entails a prohibitive runtime and numerical problems during the optimization. Not optimizing all the partition weights at once, but rather iteratively optimizing only one partition weight at a time improves the solution to the problem of optimization high dimensionality. This is conducted in [8] where an iterative method optimizes one sub-region at a time. Consequently, the dimensionality of the individual optimization problem is reduced. This enables a reliable and fast convergence of the optimization even for large sets of partitions. However, the convergence of the algorithm is not assured due to the fact that the optimization is performed on one sub-region at a time, and not on the global function $\Gamma(\Omega, G_W)$.

An alternative approach for solving the minimization Problem 4.5 is to optimize all the partition weights at once. The reference [12] employs this approach to find the suitable weights for the power diagram. By using the following gradient law the suitable power weights can be obtained, that is

$$\dot{\omega}_i = -\frac{\partial \Gamma(\Omega, G_W)}{\partial \omega_i} \tag{4.6}$$

The advantage of this approach over the first approach is in its guaranteed convergence [12]. However, the dimension of the problem is m in contrast to the dimension of the first approach which is 1.

The gradient of $\Gamma(\Omega, G_W)$ is computed as

$$\frac{\partial \Gamma(\Omega, G_W)}{\partial \omega_i} = \frac{1}{2mD^2(\Omega)} \sum_{j \in N_i} \frac{\delta_{ij}}{\zeta_{ij}} \left(\frac{d_j^2}{\Lambda(V_j(G_W, \Omega))^2} - \frac{d_i^2}{\Lambda(V_i(G_W, \Omega))^2} \right) \tag{4.7}$$

where N_i is the power cell that shares a face with the power cell $V_i(G_W, \Omega)$, δ_{ij} is the length of the shared segment of $V_i(G_W, \Omega)$ and $V_j(G_W, \Omega)$, and ζ_{ij} is the Euclidean distance between the generators' positions of $V_i(G_W, \Omega)$ and $V_j(G_W, \Omega)$, D is the total area of the map, $D = \Lambda(\Omega)$, and d_i and d_j are the desired values for the areas of the i^{th} and j^{th} power cells, respectively.

Remark 1: It is noteworthy to mention that in equation (4.7), the gradient of Γ with respect to the weight of the i^{th} generator (ω_i) depends only on the i^{th} power cell and its neighbors (power cells which share a face with the i^{th} power cell).

Remark 2: When the location or the weights of a power generator changes it only affects the shape of its power cell and its neighbors. In other words, changing the parameters of a power generator does not affect the outcome of the other power cells except its neighbors.

Equation (4.7) can be solved by using decentralized calculations. Since in this method the power generators locations are fixed, the only information that each agent needs to share with its neighbors is its current power weights and the area of its power cell. Consequently, each agent needs to receive these information from its neighbors. By having this information each agent is able to update its power weights and generate a new power cell. Equation (4.7) always converge to its optimal value [12]. The algorithm for a decentralized implementation of equation (4.7) is as follows.

Initialize the power generation information and map information in each agent, $UAV_i \leftarrow (\Omega, G_W)$. The initial position of generators are random but distinct, and the initial values for the power weights are all zero.

Agents calculate the Euclidean distance between the position of their generator and the position of their neighbors (ζ_{ij}).

while $\Lambda(V_i(G_W)) \neq d_i$ **do**

Agents calculate their power cells ($V_i(G, W)$).

Agents calculate the area of their power cell ($\Lambda(V_i(G_W))$).

Agents share the information about their cell area $\Lambda(V_i(G_W))$ and their power weights (ω_i) with the neighboring agents.

Agents calculate the length of boundary with each of their neighbors (δ_{ij}).

Agents update their power weights using equation (4.7)

end while

Figure 4.1: Pseudo code for a decentralized calculation of the optimal power weights

The decentralized implementation of this method requires a low computational cost. The agents can share the value of the power weights (ω_i) easily, and for sharing the required information about the area of the power cells each agent needs to calculate the boundary of its power cell from the power generators information of its neighboring power cells. That is, this calculation requires a low computational capability embedded in each agent as opposed to solving equations (4.5) and (4.7) in a centralized platform. The solution to equation (4.7) results in a power diagram $V(G_{W^*}, \Omega)$ that divides the map into m parts, $(\Omega_1, ..., \Omega_m)$ with areas equal to $(d_1, ..., d_m)$, obtained from the optimization phase $\big(\Lambda(\Omega_i) = d_i\big)$.

4.3 Parallel Map Partitioning and Task Assignment

In Chapter 3, we formulated a multi-objective optimization problem for determining the optimal share of the area of the map for each UAV. The solution candidates of the multi-objective optimization problem should be presented to a decision manager to select the final solution. This solution of the optimization phase is then used in the map partitioning algorithm described in the previous section. In scenarios where the importance of the objective functions are given as *a priori*, one can combine the procedure of obtaining the optimal area for each UAV through the multi-objective optimization and map partitioning using decentralized power diagram. When the importance of the objective functions is known *a priori* for the multi-objective optimization, the problem can be transferred into a single objective optimization by using the weighted sum method. An iterative algorithm can then be utilized to solve the optimization problem by updating the desired speed and area of coverage for each UAV.

The computational time of solving the task assignment optimization in this case is much less than the computational time that is required for calculating the power weights. Therefore, it is possible to run both algorithms at the same time, and send the transient results of the multi-objective optimization to the map partitioning algorithm. The task assignment algorithm updates the desired area of coverage for each UAV as it solves the optimization problem and these real-time results are then sent as parameters to the map partitioning algorithm. It is expected that by running both algorithms simultaneously the overall computational time will be reduced. The algorithm for the parallel task assignment and the map partitioning is as follows.

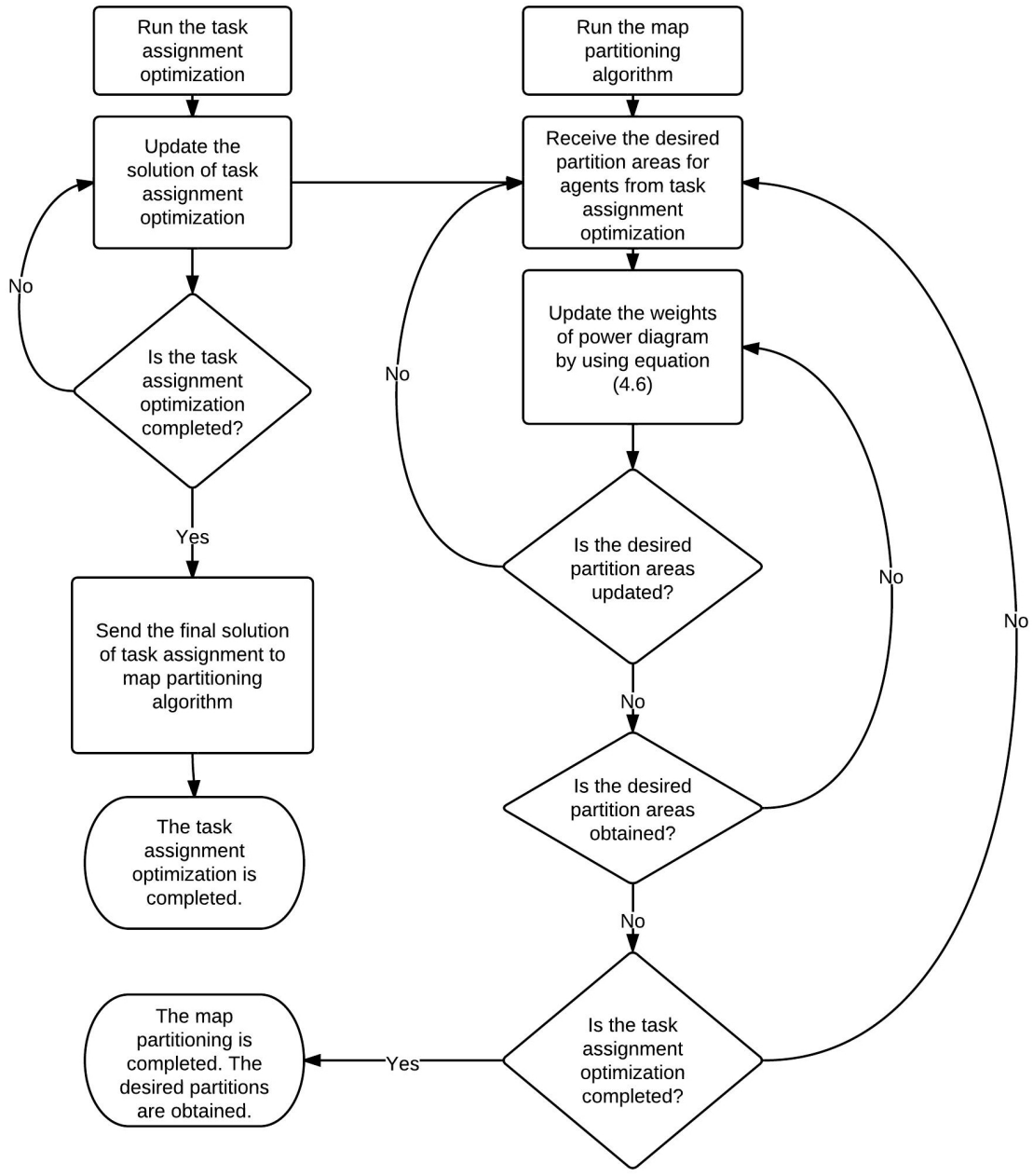

Figure 4.2: The flowchart of the parallel map partitioning and the task assignment.

4.4 Partitioning with Minimum Sum of Width

Performing a turn requires more power and is more time consuming than travelling in a straight path. For example, in the case of helicopters, it must slow down its speed, stay in hovering, perform the turn, and accelerate again to execute a successful turn.

Since the objectives of our coverage mission in this thesis are to minimize both the fuel cost and the time cost, choosing a partitioning method that imposes less turns for the agents is desirable.

In this section, we present an alternative method for partitioning the map for the coverage missions. The method used in this section partitions the map by considering the most suitable coverage path. The coverage mission in this thesis is divided into three sub-problems: task assignment, map decompositioning, and path generation. Following the solution to the task assignment problem the coverage speed and the amount of coverage area for all the UAVs are determined. From the condition (I) in Section 3.1 the length of the coverage path for each UAV can be calculated based on the radius of its sensor footprint, and its assigned area of coverage. Hence, regardless of the path generation strategy the length of the path is constant.

There are two factors that affect the total number of turns in covering specific area. The first factor is the topology of the region of coverage. As shown in Fig. 4.3 the minimum number of turns in two regions with the same area are different. Different map partitioning methods generate different subregions, and the minimum number of turns varies based on the topology of each subregion. The second factor that affects the number of turns is the path that each UAV selects to cover its region. As shown in Fig. 4.4 the total length of the two possible path generations is approximately the same, however, the number of turns is different. Hence, the path generation method directly affects the number of turns the UAVs require to perform.

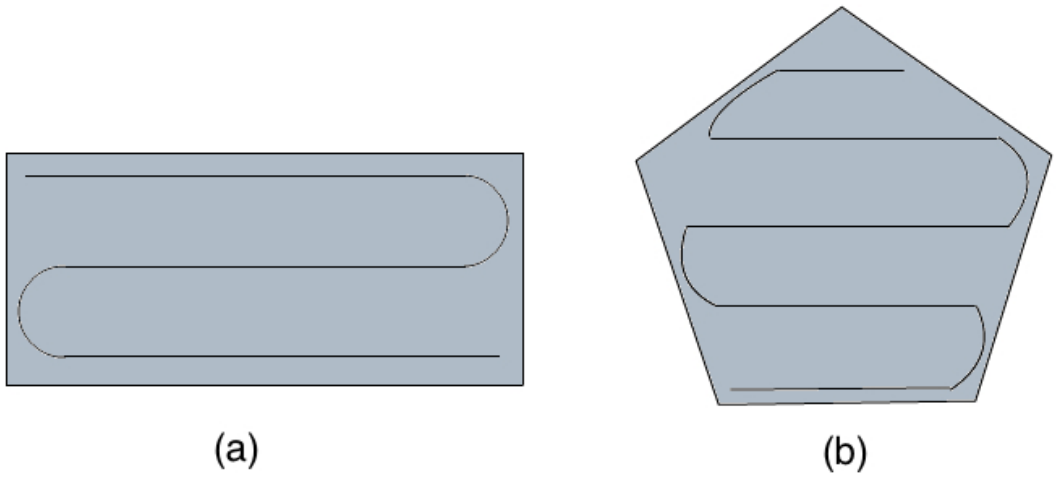

Figure 4.3: Effect of the shape of coverage region on the total number of turns. The area of coverage in both regions is equal but the number of turns in region (a) is less than the number of turns in region (b).

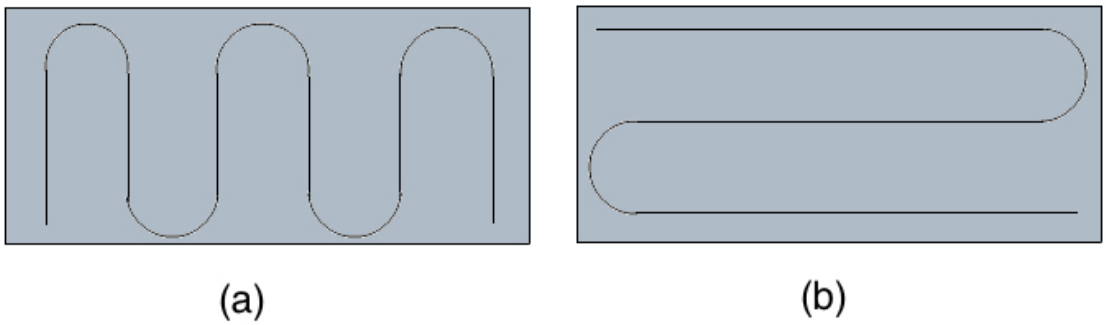

Figure 4.4: Effect of the coverage path on the total number of turns. The length of the path in both regions is equal but the number of turns in region (a) is more than the number of turns in region (b).

For any convex and bounded region in $2D$ space one can define a diameter function. The diameter of a region in direction ($\theta = \theta^*$) is the minimum distance between 2 parallel lines with the slope $\tan\theta^*$ that has the interior of the polygon in between. An example of a diameter function is shown in Fig. 4.5.

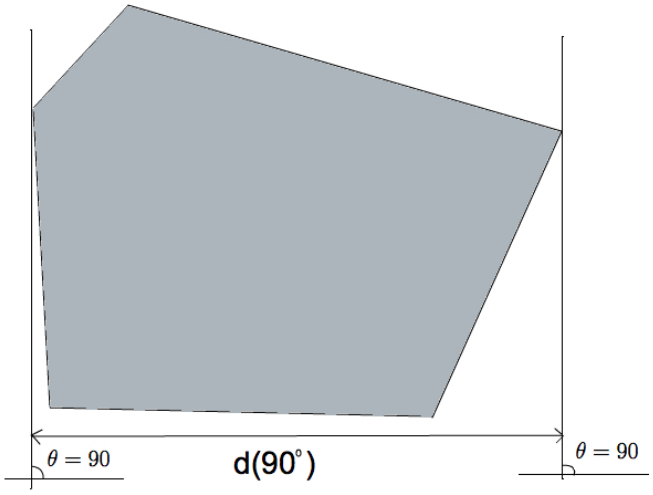

Figure 4.5: Example of a diameter of a polygon with $\theta = 90$.

The minimum diameter of a region is called the *span* or *width* of the region. For a single agent covering a polygon region, the path with the minimum number of turns is perpendicular to the direction of the minimum diameter of the polygon [95]. Therefore, to find the optimal direction for path of coverage, it is required to determine the width of the coverage region. The width of a polygon is always the distance between a vertex and an edge [92]. The maximum distance of each edge to all of the vertexes is its corresponding diameter of the polygon. For example in Fig. 4.6 the diameter of polygon with respect to V_1 is the $distance(V_1\ E_{34})$, and with respect to V_2 is the $distance(V_2\ E_{45})$. Among all the diameters associated with the edges, the minimum value is the width of the polygon, that is

$$width = \min_{i} \max_{k}\ distance(E_{i,i+1}, V_k) \tag{4.8}$$

$$\text{where} \quad i = \{1, ..., n\} \quad \text{and} \quad k = \{1, ..., n\}, \quad k \neq i,\ k \neq j$$

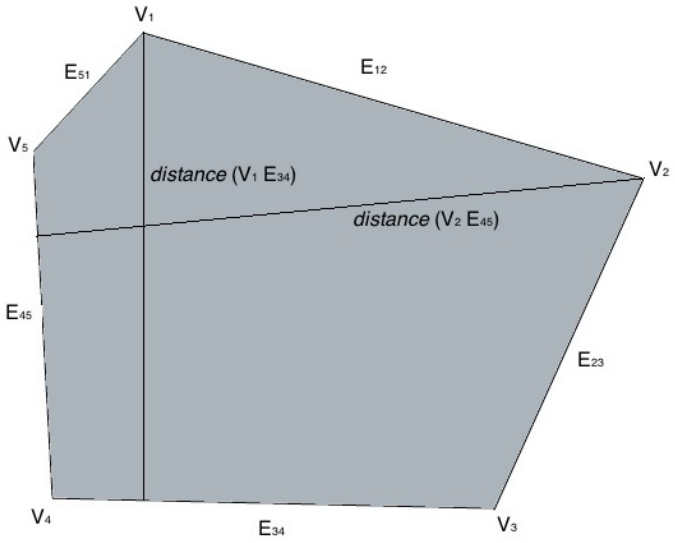

Figure 4.6: Example of a diameter of a polygon with $\theta = 90$.

For example to find the *width* of the polygon in Fig. 4.6, one should first determine the maximum distance of each vertex from all the edges. the

If an UAV uses a sweep method for its path and sweeps the region in perpendicular to the direction of the region' width, the total number of turns is minimized. Therefore, by selecting the proper sweep direction, the total number of turns in the coverage mission can be reduced. However, the sweep direction is not the only factor affecting the total number of turns. In order to minimize the total number of turns one must generate the subregions that need less turns. In Fig. 4.7, both of the coverage regions are partitioned into two subregions with the same area. For each subregion the proper direction of sweep is determined and the only difference between these 2 coverage scenarios is the map partitioning part. However, covering the region in Fig. 4.7 (a) requires less number of turns and thus is more efficient. Hence, partitioning the map properly is important in minimizing the number of turns for the mission.

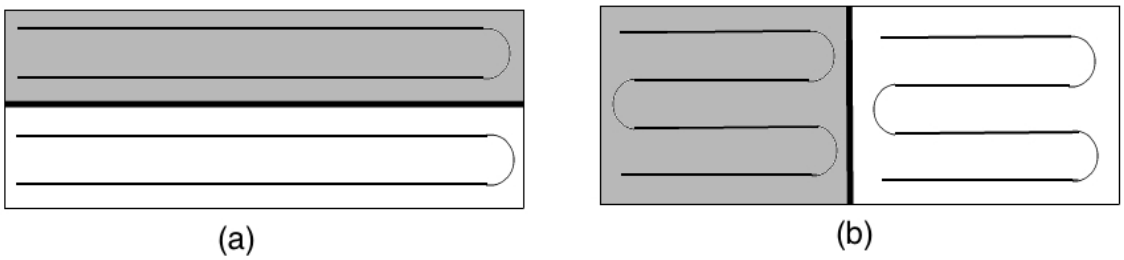

Figure 4.7: Effects of the map partitioning in the total number of turns in the coverage mission. The left region needs two turns to be covered by the two agents, and the right region needs 6 turns to be covered by the two agents.

When partitioning a map into multiple subregions, the aggregate widths of the subregions are greater or equal than to the width of the entire map. The equality occurs when the width direction of each subregion is parallel to the width direction of the entire map. Therefore, to minimize the total number of turns in the coverage mission one need to generate subregions that have the same width direction as the width of the entire map. Furthermore, in order to use this method for the coverage scenario in this thesis, the generated subregions must have pre-specified areas.

In the following the *minimum sum of width* partitioning method for obtaining the minimum total number of turns in the coverage mission is presented. We have assumed that the map is polygon in this scenario. The coverage area in real applications is also usually a polygon area, however, most of the convex non-polygon areas can be transformed into polygons by the polygonal approximation [149], [148].

The first step in the minimum sum of width method is to calculate the width of the entire map. This is to determine the direction of the sweep for the agents and also the direction of the map partitioning line. After determining the width direction, the map can be partitioned by a line perpendicular to the direction of the polygon's width. This partitioning line is parallel to at least one of the polygons sides. We assign the line on this side and move it towards the polygon so that this line is dividing the

polygon into two subregions. The first partition is generated when the area of the polygon between the two parallel lines (edge of the polygon and the partitioning line) is equal to the largest in the desired areas for the subregions. The same procedure is repeated for the second largest and this is continued until all the subregions are determined. Consequently, by utilizing this method for partitioning the map, the agents would need to perform the minimum number of turns when they are sweeping their respective coverage areas. The minimum number of turns that each agent needs to perform depends on the width of its coverage region and the radius of its sensor footprint, that is

$$N_{turn}(\Omega_i) = \lceil \frac{width(\Omega_i)}{2r_i} \rceil \tag{4.9}$$

where $\lceil x \rceil$ is the smallest integer that is not less than x. The total number of the turns of the team of UAVs in the mission is now given by

$$N_{turn}(\Omega) = \sum_{i=1}^{m} \lceil \frac{width(\Omega_i)}{2r_i} \rceil \tag{4.10}$$

4.5 Simulation Results

In this section, we apply the methodologies developed in this chapter to partition the coverage map of the second scenario presented in Chapter 3. In all the simulated scenarios it is assumed that the mission is to cover a square of $20km * 20km$ using a team of 10 heterogeneous UAVs. The UAVs specifications can be found in Table 3.1.

In Section 4.5.1, the decentralized algorithm shown in Fig. 4.1 is used to partition the map among the UAVs. The convergence of the algorithm is demonstrated along with the transient results of this algorithm as presented in Section 4.5.2. The results for the parallel computing of the task assignment and map partitioning are presented

in Section 4.5.3. The results for obtaining the minimum total turns is presented in Section 4.5.4 and the advantages and disadvantages of the methods are discussed in Section 4.5.5.

4.5.1 Map Partitioning with Power Diagram

In this section, we investigate the scenario of covering a square map of $20km * 20km$ with ten UAVs that are available to participate in the mission. Each UAV's share from the total area of Ω is assumed to be available after solving the multi-objective optimization task assignment problem (refer to Chapter 3). The solution of the optimization problem for determining the optimal area of coverage for each UAV can be found in Table 3.3. The optimal speed of coverage for each UAV is included in Table 3.4. After determining the optimal area of coverage for each UAV, the map partitioning phase must assign a subregion of Ω to each UAV with an area equal to its share obtained from the optimization phase. In the following we present the results of the map partitioning using decentralized power diagram algorithm in as shown Fig. 4.1 for different relative priorities of the time and the fuel in the mission goal.

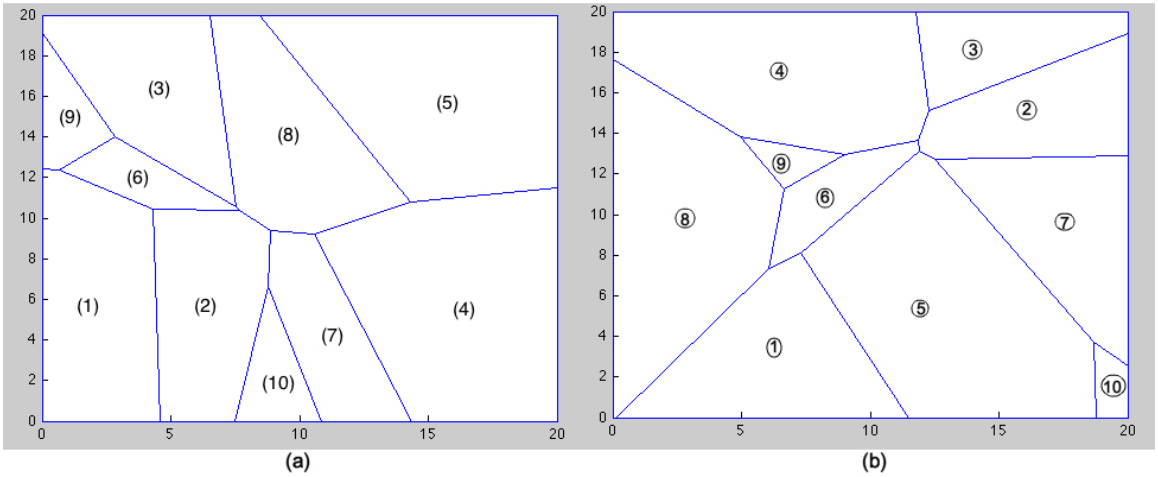

Figure 4.8: Map partitioning using the power diagram. The numbers show the assigned UAV for covering the subregions, a) Division of Ω into 10 parts. The areas are the result of the multi-objective optimization for $\omega_f = 0$ and $\omega_t = 1$, b) Division of Ω into 10 regions for $\omega_f = 0.2$ and $\omega_t = 0.8$.

Fig. 4.8 (a) shows the results of the map partitioning when $\omega_f = 0$ and $\omega_t = 1$). The mission has only one objective in this case which is the time cost. Consequently, all the UAVs are cooperating in the mission and are flying with their maximum speeds because the fuel cost is not important. The area of coverage for each UAV is determined based on its maximum speed and the range of its sensor footprint, i.e. the UAVs with faster maximum speeds and larger sensor footprints have larger areas as compared to the UAVs with slower maximum speeds and smaller sensor footprints. In this case UAV_5 and UAV_4 have the largest areas of coverage as they are the most time efficient agents for completing the coverage mission.

In Fig. 4.8 (b) the relative priority of the fuel cost is set to 0.2 and the relative importance of the time cost is set to 0.8. As the primary objective of the coverage mission is still the time cost all the UAVs are still involved in the coverage mission. However, only UAV_6, UAV_7 and UAV_8 are still flying with their maximum speeds. All the other seven UAVs are flying with speeds between their minimum and maximum speeds in order to obtain the optimal mission cost. In this case UAV_5 and

UAV_8 have the largest areas of coverage.

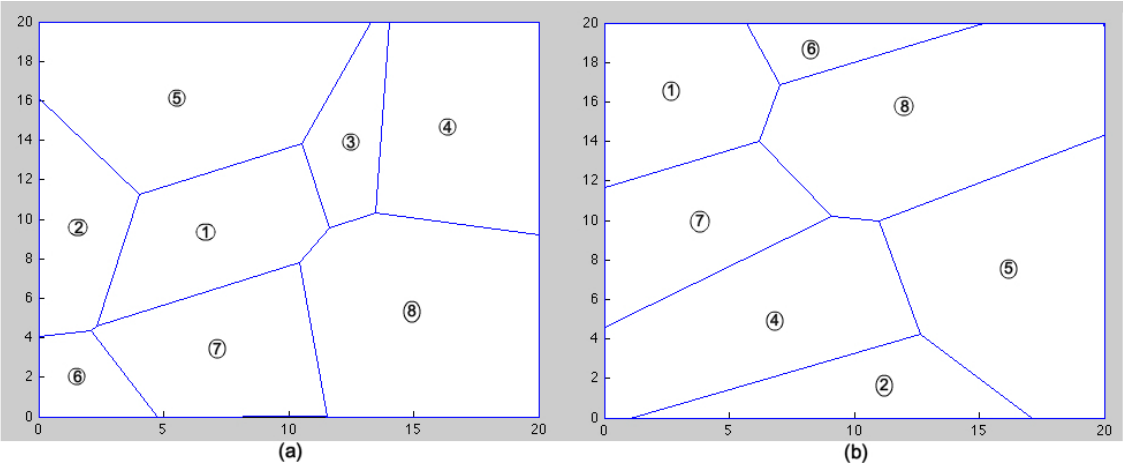

Figure 4.9: Map partitioning using the power diagram. The numbers show the assigned UAV for covering the subregions a) Division of Ω into 8 parts. The areas are the result of the multi-objective optimization for $\omega_f = 0.4$ and $\omega_t = 0.6$, b) Division of Ω into 7 regions for $\omega_f = 0.7$ and $\omega_t = 0.3$.

In Fig. 4.9 (a) the relative priority of the fuel cost is set to 0.4 and the relative priority of the time cost is set to 0.6. It can be seen that only 8 UAVs are assigned to the coverage subregions in this case. UAV_9 and UAV_{10} are not involved as their sensor footprints are very small compared to the other sensor footprints in the team. As the priority of the fuel has increased all the UAVs are flying with speeds slower than their maximum speeds to minimize the fuel and the time cost. In this case UAV_5 and UAV_8 have the largest areas of coverage.

Fig. 4.9 (b) shows the results of the map partitioning when $\omega_f = 0.7$ and $\omega_t = 0.3$. Due to the increase in the priority of the fuel, UAV_3 is also omitted from the mission, and only seven UAVs are involved in the coverage mission.

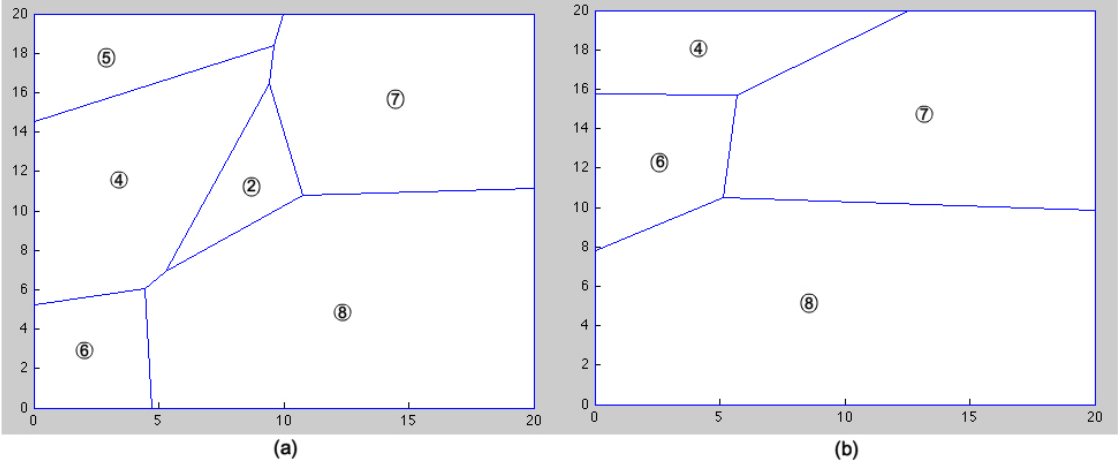

Figure 4.10: Map partitioning using the power diagram. The numbers show the assigned UAV for covering the subregions a) Division of Ω into 6 parts. The areas are the result of the multi-objective optimization for $\omega_f = 0.9$ and $\omega_t = 0.1$ b) Division of Ω into 4 regions for $\omega_f = 0.95$ and $\omega_t = 0.05$.

In Fig. 4.10 (a) the relative priority of the fuel cost is set to 0.9 and the relative priority of the time cost is set to 0.1. Only 6 UAVs are assigned to cover the subregions of the map as UAV_1 is also dropped from the best group of UAVs to perform the mission. All the UAVs in the mission are flying with their minimum speeds as the fuel cost is more important than the time cost in this case. Note that in this case UAV_8, UAV_7, and UAV_5 have the largest areas of coverage.

Increasing the relative priority of the fuel cost by another 5% results in dropping 2 other UAVs from the team for the coverage mission. Fig. 4.10 (b) shows the results of the map partitioning when $\omega_f = 0.95$ and $\omega_t = 0.05$. Only 4 UAVs are assigned to carry out the coverage mission in this case, and all of them are assigned with their minimum speeds.

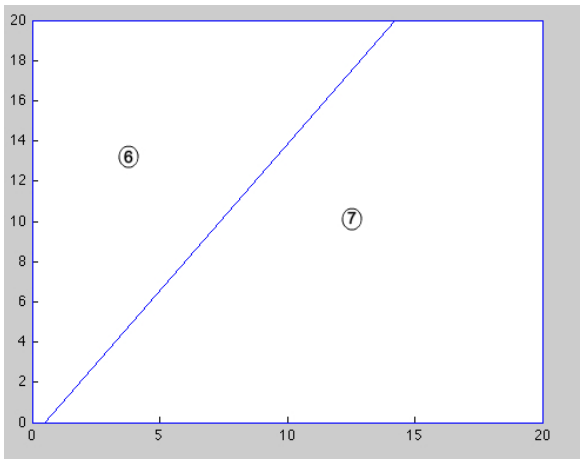

Figure 4.11: Map partitioning using the power diagram. The numbers show the assigned UAV for covering the subregions. Division of Ω is into 2 parts and the areas are the result of the multi-objective optimization for $\omega_f = 1$ and $\omega_t = 0$.

Fig. 4.11 shows the results of the map partitioning when $\omega_f = 1$ and $\omega_t = 0$. The mission has only one objective in this case which is the fuel cost. As a result, only the most fuel efficient UAVs participate in this case which are UAV_7 and UAV_6. The area of the coverage for each UAV is determined based on its minimum speed and size of its sensor footprint. It is interesting to note that all the map is not assigned to only one UAV, and still having more than one UAV would result in the minimum mission cost.

4.5.2 Transient Results of the Decentralized Power Diagram Map Partitioning

In this section, we present the transient results of partitioning the area of interest using the decentralized power diagram. The transient results of the scenario with $\omega_f = 0$ and $\omega_t = 1$ for the objective cost functions are presented. Given that the objective of the mission is to complete the coverage as fast as possible ($\omega_t = 1$), all the UAVs are participating in the mission with their maximum speeds. The decentralized power diagrams algorithm shown in Fig. 4.1 is implemented for this scenario. Each

agent shares the information about its power weights and the area of its power cell with all of its neighboring UAVs (two agents are neighbor if their associated power cells share a face) and then updates its power cell and its power weight. This process continues until the area of each agent' power cell reaches its desired value. Figure 4.12 shows the power weights throughout the design procedure.

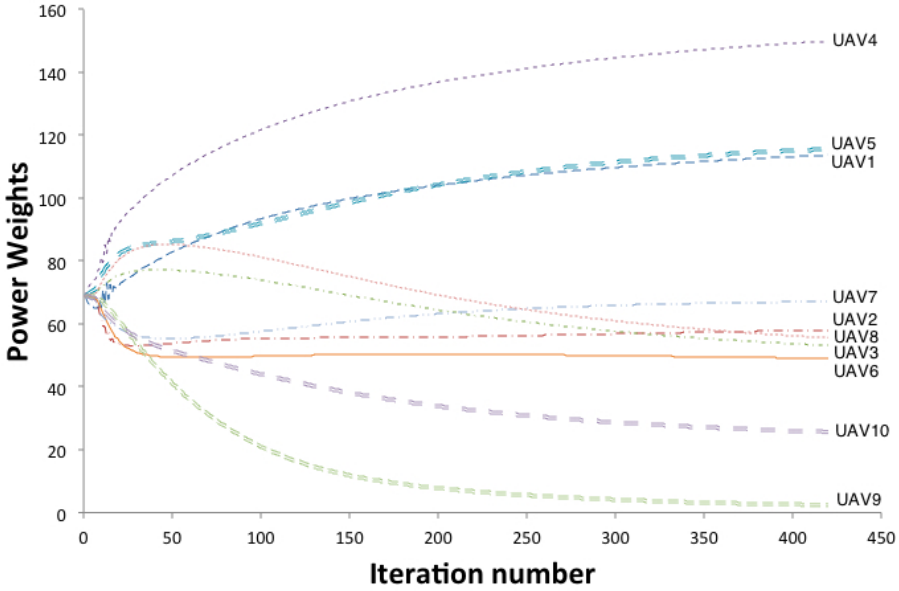

Figure 4.12: Changes in the power weights in partitioning the map using the decentralized power diagram algorithm.

Figure 4.13 (a) shows the power cells at the start of the process (iteration 25). The coverage regions of UAV_1, UAV_2, UAV_4, UAV_5 are smaller than the desired regions which is the result of the task assignment optimization problem in the first row of Table 3.3. The coverage regions of the remaining UAVs in the team are bigger than the desired regions. Hence, the UAV_1, UAV_2, UAV_4, and UAV_5 need to expand their power cells, and the rest of the UAVs need to shrink their subregions.

Figure 4.13 (b) shows the power cells at iteration 150. As the agents update their power weights using equation (4.7), their associated power cells are changing

constantly. Figure 4.14 (a) shows the power cells at iteration 300. The area of each power cell converges to the desired value after 420 iteration. The final power cells are shown in Figure 4.14 (b). The graph of the area of the power cells is presented in Fig. 4.15.

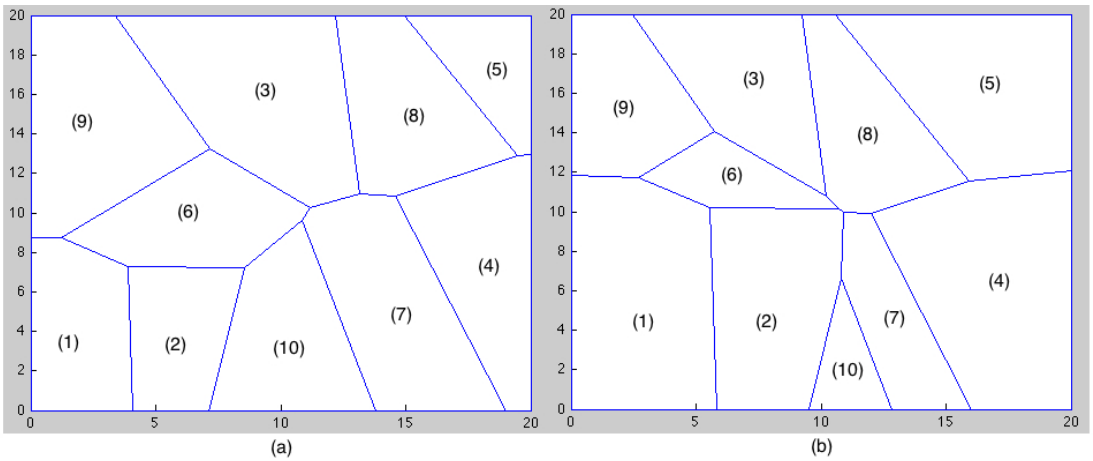

Figure 4.13: The results of the map partitioning using the decentralized power diagrams at iterations a) 25 and b) 150. The numbers show the assigned UAVs for covering each subregion.

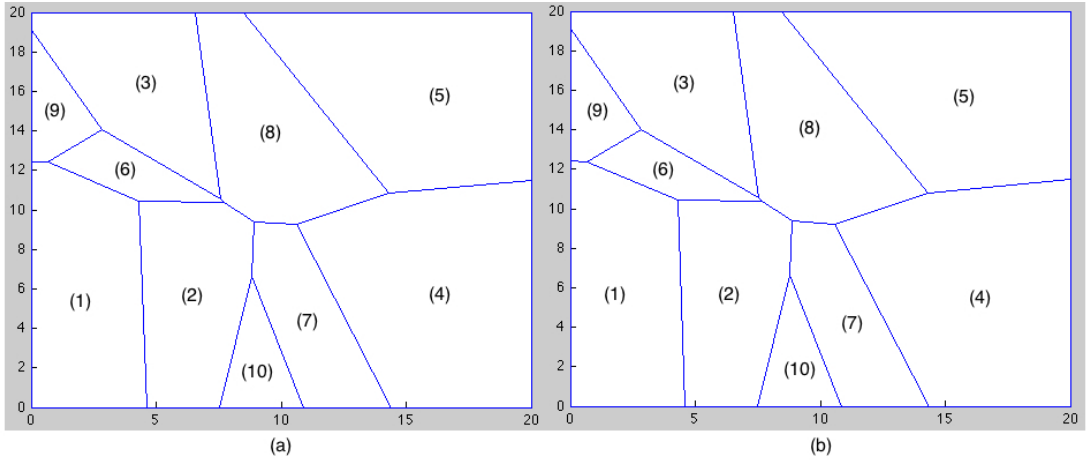

Figure 4.14: The results of the map partitioning using the decentralized power diagrams at iterations a) 300 and b) 420. The numbers show the assigned UAVs for covering each subregion.

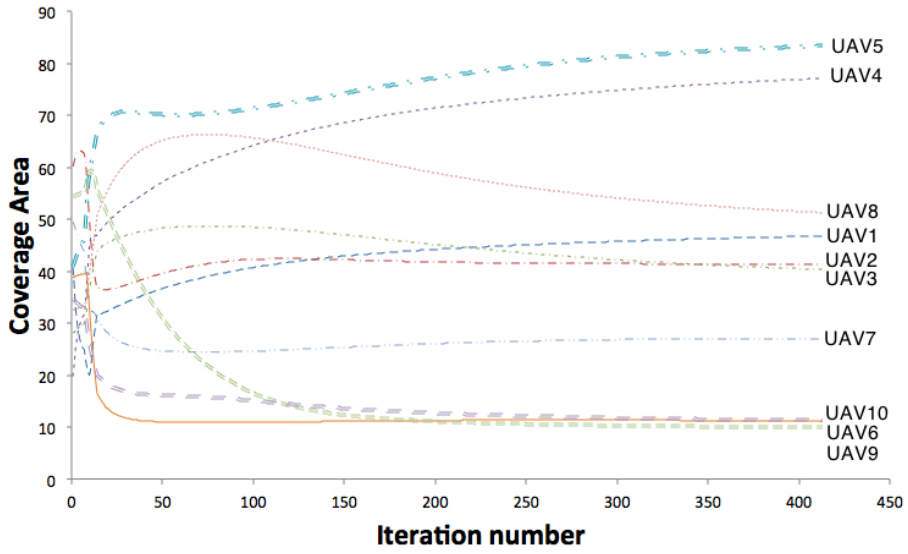

Figure 4.15: Changes in the area of the power cells as the agents update their power cells using the decentralized algorithm.

4.5.3 Simultaneous Task Allocation and Map Partitioning

In coverage scenarios where the preference of the user is known, both the task assignment and map partitioning problems can be solved at the same time. The computational time of the map partitioning is more than the time required for solving the task assignment. Hence, the algorithms for solving both problems can be executed at the same time. While the task assignment problem is executed its transient results are used by the power diagram algorithm as the desired coverage areas for the team of UAVs. Hence, the desired areas of coverage are changing during the early stages of the process. Thus, it requires more time for the map partitioning algorithm to divide the map with suitable areas. However, the total computational time for solving the task assignment and the map partitioning problems (task assignment problem is solved first and then its results are used for solving the map partitioning problem) is higher than the computational time for solving both problems in parallel.

The scenario of the last section is simulated here using parallel computing of the task

assignment problem and the map partitioning problem. Figures 4.16 and 4.17 show the change of the power weights and the change in the area of power cells during the procedure, respectively. As can be seen in Fig. 4.17 the line for the area of each UAV's subregion is non-smooth at the beginning stages, where the map partitioning algorithm is using the transient results of the task assignment problem. However, it does not prevent the method for partitioning the map into proper subregions. The power cells are shown in Figs. 4.18, 4.19, and4.20 through different stages of the solution process. As shown in Fig. 4.15 the number of iterations in this method is approximately the same as the number of iterations in the last section.

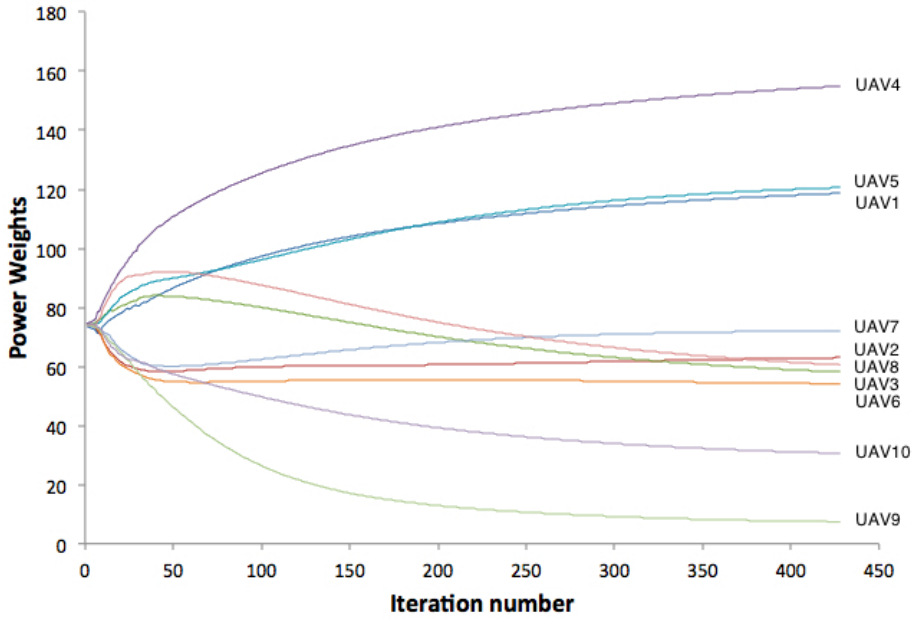

Figure 4.16: Changes in the power weights in partitioning the map in parallel using solving the task assignment problem.

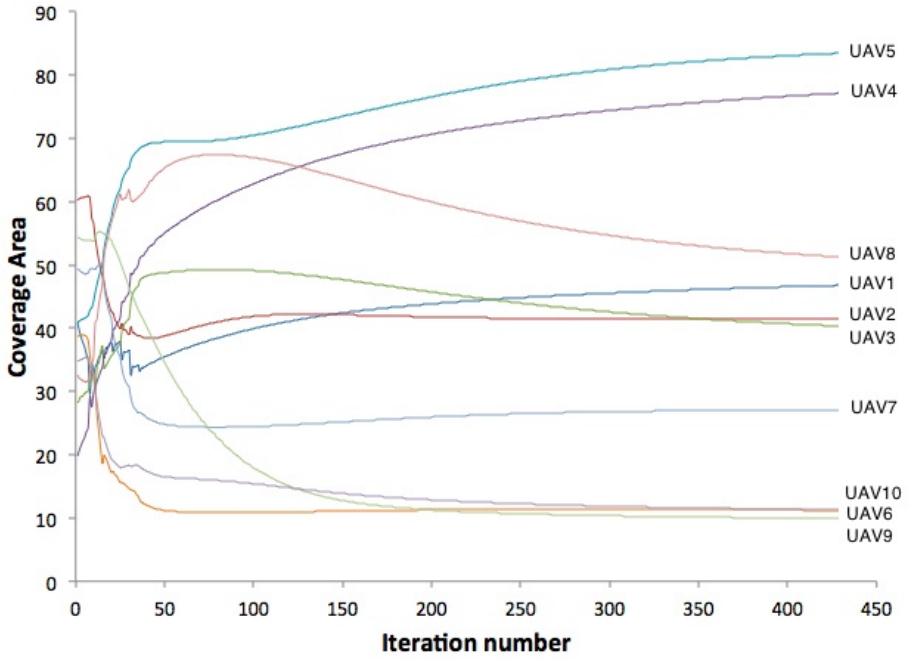

Figure 4.17: Changes in the power weights in partitioning the map in parallel using solving the task assignment problem.

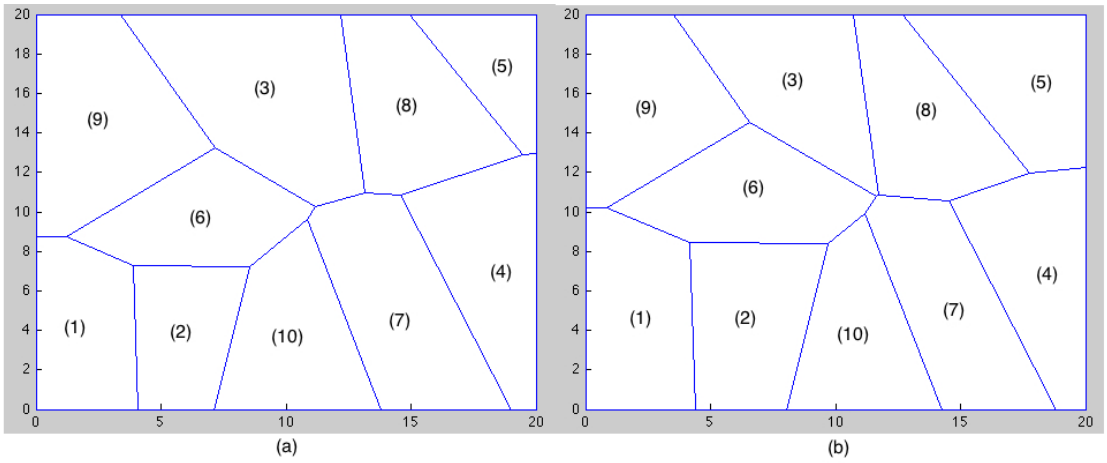

Figure 4.18: The partitioned map in simultaneous task allocation and map partitioning. The power cells are obtained by using the decentralized power diagrams at iterations a) 20 and b) 50. The numbers show the assigned UAVs for covering each subregion.

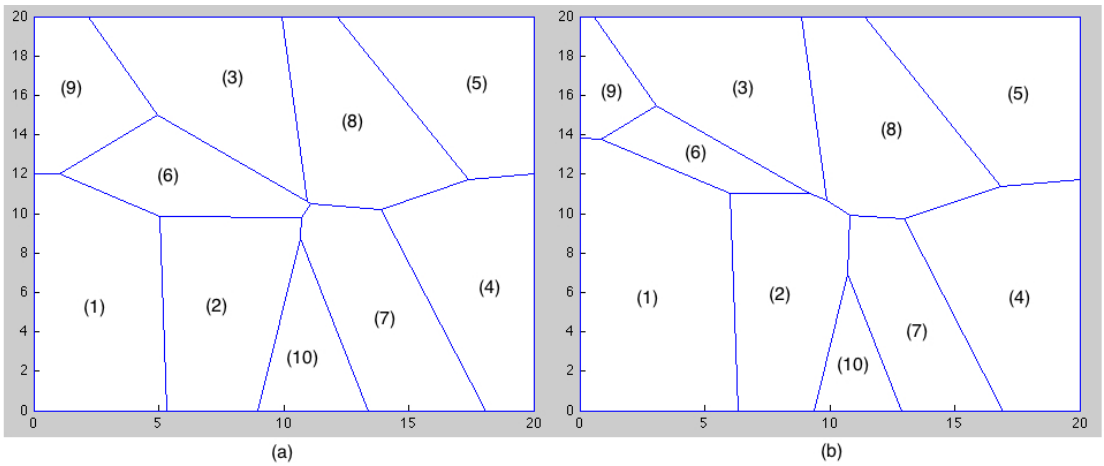

Figure 4.19: The partitioned map in simultaneous task allocation and map partitioning. The power cells are obtained by using the decentralized power diagrams at iterations a) 100 and b) 200. The numbers show the assigned UAVs for covering each subregion.

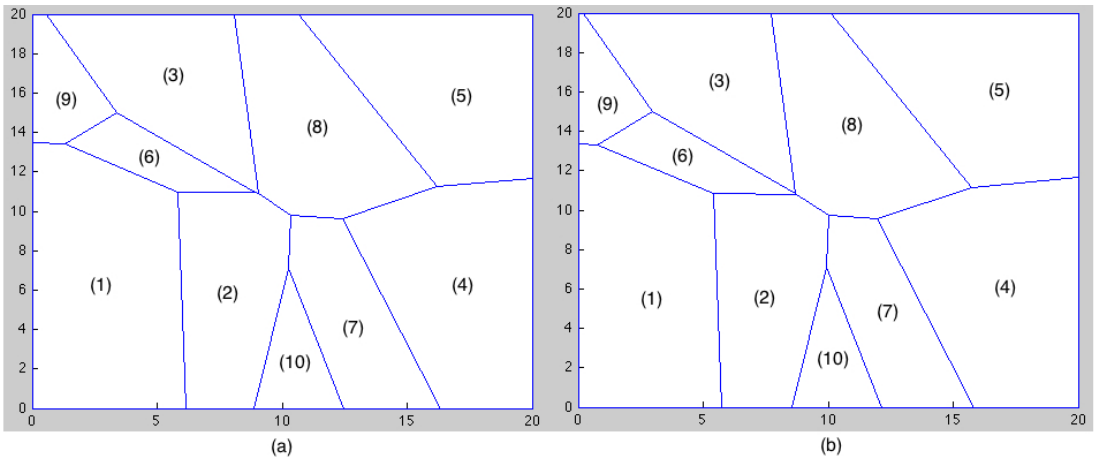

Figure 4.20: The partitioned map in simultaneous task allocation and map partitioning. The power cells are obtained by using the decentralized power diagrams at iterations a) 300 and b) 420. The numbers show the assigned UAVs for covering each subregion.

4.5.4 Minimum Sum of Width

In order to determine the desired partitions which results in minimum number of turns, one needs to find the width of the coverage region. The map for the simulated scenario in this section is a square of $20km * 20km$. It is trivial that a width of a square is equal to the length of its side. Furthermore, as the coverage region is symmetrical the direction of the sweep lines of coverage can be chosen to be parallel to each of the

sides. In Fig. 4.21 the results of this method for the coverage scenario where the only objective of the mission is time ($\omega_t = 1$) are presented.

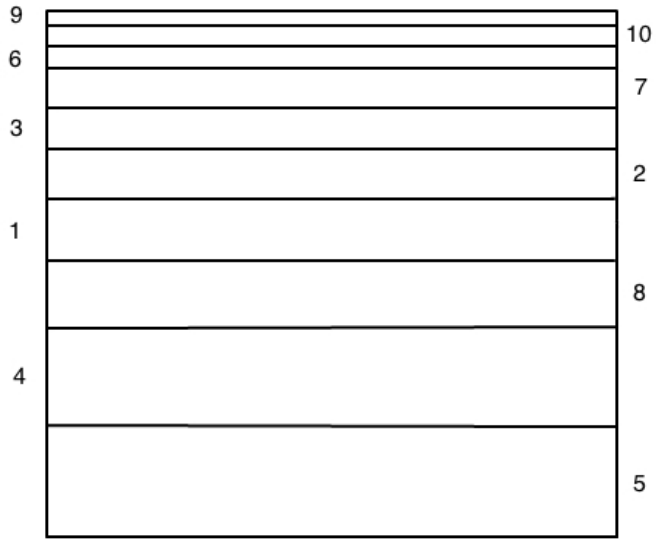

Figure 4.21: The partitioned map with the minimum sum of the width method. The numbers show the assigned UAVs for covering each subregion.

4.5.5 Discussion

The two objective functions in the developed mission scenarios in this thesis are the time cost and the fuel cost. The parameters that affect these objectives are the area of each UAV subregion, the speed of each UAV, and the number of turns that the UAV has to perform in order to complete its search. The first two parameters are considered in formulating the task assignment problem in Chapter 3. Solving the multi-objective task assignment problem would result in determining the optimal speeds and the coverage areas that minimize the mission costs.

The third factor in the mission cost can be optimized by selecting the proper subregions shape and search direction for the team of UAVs. By using the equation (4.10), one can calculate the total number of turns that each partitioning method

requires for completing the coverage mission. The total number of turns that the team of UAVs has to perform for covering the region in Fig. 4.21 with the minimum sum of width method is 159. On the other hand, the UAVs have to perform 547 turns when their subregions are generated by the power diagram method (Fig. 4.8). Since minimizing the number of turns enhances the mission time and its fuel efficiency, partitioning the map with the minimum sum of width method would result in completion of the mission in less amount of time and with less amount of fuel consumed.

Another criterion in comparing the performance of the partitioning methods is their computational time and complexity. Partitioning the map with a minimum sum of width requires much less computation as compared to the power diagram method. In the minimum sum of width method the agents can determine their region of coverage by having the information about the coverage region and the results of the task assignment problem and they do not need to share any information with the other agents.

One should note that although partitioning the map with minimum sum of width method guarantees the minimum total number of turns in the mission, however, the effect of minimizing the number of turns on the fuel and the time cost is not the same as the effect of performing the optimization for finding the optimal tasks. In other words, the optimization performed in chapter 3, has the most effect on minimizing the mission cost. Moreover, one should not consider as a disadvantage of the power diagram method solely because it leads to more number of turns. Specifically, the power diagram has certain advantages over the minimum sum of width method. For example, the number of the nearby agents in certain distance is an important factor in choosing the proper partitioning method. In case of failure in one of the agents having more agents in the nearby regions is an advantage. Furthermore, in search and destroy missions where the agents must attack the target as a team, partitioning using

the power diagrams is more beneficial. In partitioning the map using the minimum sum of width method each subregion has a maximum of two neighboring subregions. This number is much higher in case of the power diagram partitioning method.

4.6 Conclusions

In this chapter, the problem of map partitioning for a team of heterogeneous UAVs is studied. The area of interest is partitioned into subregions using specified areas which are the results of the task assignment optimization problem. After determining the optimal speeds and coverage areas for the UAVs the map should be partitioned according to this optimal solution. Two partitioning methods are studied in this chapter. Partitioning using distance functions are introduced in this chapter and a decentralized power diagram algorithm is presented. In using the power diagrams both the generators positions and their associated power weights are parameters that can be optimized and set to partition the map. A framework for solving the task assignment and the map partitioning in parallel is also developed. During the time that the task assignment is still computing the optimal tasks the agents use the transient results of the task assignment problem. Since performing a turn requires more time and energy than moving in straight direction, it is desired to minimize the number of the turns in the mission. A criterion for achieving the minimum number of the turns using a single UAV in a fixed region is studied and extended to develop a method for partitioning the map among multiple UAVs that minimize the number of the turns.

The methods in this chapter is employed on an example scenario that involves a team of ten heterogeneous UAVs, which must cover an square area of $20km * 20km$ obtaining minimum mission cost for different importance levels of the fuel cost and the time cost. All the methods successfully partition the map into subregions with

pre-specified areas and consequently the mission can be performed while the mission cost is minimized.

Chapter 5

Conclusions and Future Work

The use of teams of unmanned aerial vehicles (UAVs) for performing complex tasks has been an area of significant interest recently. The potential advantages of multi-agent systems such as reduction of the operating time and cost, and robustness to single agent failure are the motivating factors justifying their use in performing a required mission. One of the applications of multi-agent UAV systems is complete coverage of an area of interest. Recent natural disasters such as the Haiti earthquake and the Indonesian Tsunami have highlighted the need for an efficient and timely effective search and rescue strategy.

Use of a team of UAVs is among the best options for addressing these problems since the UAVs can perform their tasks in minimum amount of time and can be utilized in environments that are hazardous to human operators. Collecting and processing data under constrained resources such as under limitations in the fuel, restrictions in time, and constraints in the search region should be considered in developing a strategy for cooperative coverage by multi-agent UAVs. Besides search and rescue missions, the complete coverage strategy has other applications in both civilian and military domains such as cleaning robots, mine detection, and oil spill

detection.

In this thesis, the problem of task assignment for a team of non-homogenous UAVs that are set to perform a coverage mission on a given area of interest that is convex is addressed. It is shown that the UAVs are assigned to act cooperatively to optimize a reward/cost function. This thesis presents a new decomposition strategy for a coverage mission using a team of heterogeneous UAVs. We addressed the area decomposition and the task assignment problems through two stages. In the first stage, we determined the optimal speed and the coverage area for every agent through multi-objective optimization framework. Having the results from the optimization step, the map is then partitioned by using the power diagrams in the second phase of our methodology.

In Chapter 3 we addressed the problem of determining the optimal speed and coverage area for each individual in a team of UAVs that is undertaking the search and coverage mission. For obtaining the optimal results we employed the multi-objective platform to tackle all the different aspects of the mission. The multi-objective evolutionary algorithm is used to obtain the Pareto front when the importance of each objective is unknown. The weighted sum method is also employed to compare with the results of the evolutionary algorithm. Finally the methods in Chapter 3 are employed on an example scenario involving a team of heterogeneous UAVs, which must cover an square area of $20km * 20km$ obtaining the minimum mission cost for different priority levels of fuel cost and time cost.

In Chapter 4 the map partitioning problem is addressed. The required criterion for an applicable method in our scenario is a method that can partition the area of interest into sub regions with specified areas. It is also desired that the shape

of subregions are convex in order to make the use of single-searcher path planning methods applicable to the path planning for our scenarios. Both the Voronoi diagrams and the power diagrams are studied in Chapter 4 and power diagram is chosen due to its extra degree of freedom over the Voronoi diagram. The power diagram is used on two example scenarios involving a team of heterogeneous UAVs, which must cover an square area of $20km * 20km$. The desired areas for each UAV is the result of the multi-objective optimization carried out in Chapter 3. For both cases the map is divided into suitable sub regions and consequently the mission is performed while the mission cost is minimized.

Comparing the results of our proposed strategy with the current strategies in the literature, we should point out that our method is capable of handling different types of UAVs for performing the coverage task. Another advantage of our method is its consideration of multi-objective functions, which allow the mission designer to make decisions based on all the high priority factors in the mission. The algorithm also chooses the best group of UAVs in the team for performing the coverage mission. Our map partitioning method prevents the UAVs from having overlap in their path coverage. Furthermore, we use area decomposition after determining the optimal share of each UAV. Performing the area decomposition before the task assignment leads to forcing the UAVs to choose between the generated regions. Determining the optimal workload for each agent according to the mission objectives helps one to perform the mission with the minimum cost. Thus, the optimality of our proposed strategy makes it very suitable for implementing it in the coverage missions.

Since the ultimate goal of research into multiple UAVs is to develop coordination algorithms for teams of UAVs in real world scenarios a possible direction for further research could be to develop a strategy for the map partitioning algorithm that

works in non-convex regions. Towards this end, the environment of the operation can have multiple no-fly zones and obstacles. The first phase of our method is capable of handling non-convex regions, as it is independent of the shape of the region of interest. The only parameter from the map that influences the results of the multi-objective optimization phase is the total coverage area. Hence, upon the development of a map partitioning algorithm that is capable of partitioning the non-convex areas into area specified sub regions, the proposed decomposition strategy in this thesis can be deployed in the non-convex regions.

Another direction for future research is to implement the proposed task assignment strategy in real-time. To this end, the major criterion is to develop a real time multi-objective optimization strategy. By developing a real-time task assignment strategy this method can be utilized as a real-time fault tolerant strategy. Upon occurrence of a fault, the UAVs need to reset their optimization phase with the new uncovered area in the map. Future work in this direction could be to implement the entire strategy in a decentralized framework to minimize the overall computational complexity and time processing.

Bibliography

[1] M. Abrahamsson, O. Norberg, and K. Noone, "UAVs for atmospheric research in the North of Sweden," 16th ESA Symposium on European Rocket and Balloon Programmes and Related Research, 2003.

[2] D. M. Thome, and T. M. Thome, "Radio-controlled model airplanes: inexpensive tools for low-level aerial photography." Wildlife Society Bulletin 28(2):343-346, 2000.

[3] J. Blazakis, "Border Security and Unmanned Aerial Vehicles," CRS Report for Congress, January 2004.

[4] A. Puri, "A survey of Unmanned Aerial Vehicles (UAV) for Traffic Surveillance," Technical Report 2004.

[5] S. Sasa, Y. Matsuda, M. Nakadate, and K. Ishikawa, Ongoing Research on Disaster Monitoring UAV at JAXAs Aviation Program Group, Proceedings of the SICE Annual Conference, Tokyo, Japan, 2008.

[6] M. A. Goodrich, B. S. Morse, D. Gerhardt, J. L. Cooper, M. Quigley, J. A. Adams, and C. Humphrey, "Supporting wilderness search and rescue using a camera-equipped mini UAV," Journal of Field Robotics, 25: 89-110. 2008.

[7] H. Eisenbeiss, and M. Sauerbier, "Investigation of UAV systems and flight modes for photogrammetric applications," The Photogrammetric Record, 26: 400-421. 2011.

[8] M. Balzer, "Capacity-Constrained Voronoi Diagrams in Continuous Spaces," Proceedings of the Sixth International Symposium on Voronoi Diagrams, pp. 79-88, June 2009.

[9] T. Ohyama, "Division of a Region into Equal Areas Using Additively Weighted Power Diagrams," Proceedings of the 4th International Symposium on Voronoi Diagrams in Science and Engineering, pp. 152-158, July 2007.

[10] M. Pavone, A. Arsie, E. Frazzoli, and F. Bullo, "Equitable partitioning policies for robotic networks," Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2356-2361, May 2009.

[11] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," IEEE Transactions on Robotics and Automation, vol.20, no.2, pp. 243- 255, April 2004.

[12] M. Pavone, A. Arsie, E. Frazzoli, and F. Bullo, "Distributed Algorithms for Environment Partitioning in Mobile Robotic Networks," IEEE Transactions on Automatic Control, vol. 56, no. 8, pp. 1834-1848, Aug. 2011.

[13] P. F. Hokayem, D. Stipanovic, and M. W. Spong, "On persistent coverage control," Proceedings of the 46th IEEE Conference on Decision and Control, pp. 6130-6135, Dec. 2007.

[14] S. Lim, and H. Bang, "Waypoint guidance of cooperative UAVs for intelligence, surveillance, and reconnaissance," Proceedings of the IEEE International Conference on Control and Automation, pp. 291-296, Dec. 2009.

[15] I. I. Hussein, and D. M. Stipanovic, "Effective Coverage Control using Dynamic Sensor Networks with Flocking and Guaranteed Collision Avoidance," Proceedings of the American Control Conference, pp. 3420-3425, July 2007.

[16] Y. Guo, and M. Balakrishnan, "Complete coverage control for nonholonomic mobile robots in dynamic environments," Proceedings of the 2006 IEEE International Conference on Robotics and Automation, pp. 1704-1709, May 2006.

[17] H. Choset, "Coverage for robotics - A survey of recent results," Annals of Mathematics and Articial Intelligence, vol. 31, no. 1-4, pp. 113-126, 2001.

[18] B. Bethke, J. Redding, J. P. How, M. A. Vavrina, and J. Vian, "Agent capability in persistent mission planning using approximate dynamic programming," Proceedings of the American Control Conference (ACC), pp. 1623-1628, June-July 2010.

[19] N. Nigam, and I. Kroo, "Persistent Surveillance Using Multiple Unmanned Air Vehicles," Proceedings of the 2008 IEEE Aerospace Conference, pp. 1-14, March 2008.

[20] N. Boonpinon, and A. Sudsang, "Constrained coverage for heterogeneous multi-robot team," Proceedings of the IEEE International Conference on Robotics and Biomimetics, pp. 799-804, Dec. 2007.

[21] R. J. Meuth, E. W. Saad, D. C. Wunsch, and J. Vian, "Adaptive task allocation for search area coverage," Proceedings of the IEEE International Conference on Technologies for Practical Robot Applications, pp. 67-74, Nov. 2009.

[22] L. Pimenta, V. Kumar, R. C. Mesquita, and G. Pereira, "Sensing and coverage for a network of heterogeneous robots," Proceedings of the 47th IEEE Conference on Decision and Control, pp. 3947-3952, Dec. 2008.

[23] P. B. Sujit, and D. Ghose, "Search using multiple UAVs with flight time constraints," IEEE Transactions on Aerospace and Electronic Systems, vol. 40, no.2, pp. 491- 509, April 2004.

[24] P. B. Sujit, and D. Ghose, "Optimal uncertainty reduction search using the k-shortest path algorithm," Proceedings of the American Control Conference, pp. 3269- 3274 vol. 4, June 2003.

[25] J. Y. Yen, "Finding the K Shortest Loopless Paths in a Network," Management Science, Vol. 17, No. 11, Theory Series, pp. 712-716, Jul, 1971.

[26] P. B. Sujit, and D. Ghose, "Multiple Agent Search of an Unknown Environment using Game Theoretical Models," Proceedings of the American Control Conference, pp. 5564-5569, vol. 6, June - July 2004.

[27] P. B. Sujit, and D. Ghose, "Multiple UAV search using agent based negotiation scheme," Proceedings of the American Control Conference, pp. 2995- 3000 vol. 5, June 2005.

[28] M. L. Baum, and K. M. Passino, "A Search-Theoretic Approach to Cooperative Control for Uninhabited Air Vehicles", Proceedings of the AIAA Guidance, Navigation and Control Conference, and Exhibit, Monterey, CA, USA, August 2002.

[29] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, "Computational Geometry: Algorithms and Applications, 2nd, rev. ed., Berlin, New York, Springer 2000.

[30] A. Suzuki and A. Okabe, "Using Voronoi Diagrams. Facility Location: A Survey of application and Methods" Z. Drezner (ed.), Chapter 6, Springer Series in Operations Research, New York, 1995.

[31] O. Baron, O. Berman, D. Krass, and Q. Wang, "The equitable location problem on the plane, Proceedings of European Journal of Operational Research, 183(2):578-590, 2007.

[32] Q. Jiang, "An improved algorithm for coordination control of multi-agent system based on r-limited voronoi partitions," Proceedings of the IEEE International Conference on Automation Science and Engineering, pp.667-671, Oct. 2006.

[33] J. Cortes, "Coverage Optimization and Spatial Load Balancing by Robotic Sensor Networks," IEEE Transactions on Automatic Control, vol. 55, no. 3, pp. 749-754, March 2010.

[34] H. Sayyaadi, M. Moarref, "A Distributed Algorithm for Proportional Task Allocation in Networks of Mobile Agents," IEEE Transactions on Automatic Control, Vol. 56, no. 2, pp. 405-410, Feb. 2011.

[35] M. Moarref, and H. Sayyaadi, "Facility Location Optimization via Multi-Agent Robotic Systems," Proceedings of the IEEE International Conference on Networking, Sensing and Control, pp. 287-292, April 2008.

[36] D. A. Castanon, and C. Wu, "Distributed algorithms for dynamic reassignment," Proceedings of the 42nd IEEE Conference on Decision and Control, pp. 13- 18 Vol.1, Dec. 2003.

[37] G. Wang, G. Cao, T. La Porta, and W. Zhang, "Sensor relocation in mobile sensor networks," Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies, pp. 2302-2312 vol. 4, March 2005.

[38] Y. Mei, Y. H. Lu, Y. C. Hu, C. S. G. Lee, "Reducing the number of mobile sensors for coverage tasks," Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1426- 1431, Aug. 2005.

[39] Y. Mei, X. Changjiu, S. Das, Y. C. Hu, and Y. H. Lu, "Replacing Failed Sensor Nodes by Mobile Robots," Proceedings of the 26th IEEE International Conference on Distributed Computing Systems Workshops, pp. 87, July 2006.

[40] M. Alighanbari, L. F. Bertuccelli, and J. P. How, "A Robust Approach to the UAV Task Assignment Problem," Proceedings of the 45th IEEE Conference on Decision and Control, 2006, pp. 5935-5940, Dec. 2006.

[41] I. Kamon, E. Rimon, and E. Rivlin, Tangentbug: A range-sensor-based navigation algorithm, The International Journal of Robotic Researches, vol. 17, no. 9, pp. 934-953, 1998.

[42] S. Lloyd, "An optimization approach to relaxation labelling algorithms," Image and Vision Computing, Volume 1, Issue 2, May 1983, Pages 85-91.

[43] C. H. Caicedo-Nuez, and M. Zefran, "A coverage algorithm for a class of non-convex regions," Proceedings of the 47th IEEE Conference on Decision and Control, pp. 4244-4249, Dec. 2008.

[44] A. Breitenmoser, M. Schwager, J. C. Metzger, R. Siegwart, and D. Rus, "Voronoi coverage of non-convex environments with a group of networked robots," Proceedings of the 2010 IEEE International Conference on Robotics and Automation, pp. 4982-4989, May 2010.

[45] M. Khan, and K. Khorasani, "Coordinated Rendezvous for Multiple Unmanned Aerial Vehicles (UAVs) Subject to Actuator Faults," Proceedings of the 2010 conference on Modeling, Identification, and Control, 2010.

[46] M. Khan, "Coordinated Rendezvous and Surveillance for Multiple Unmanned Aerial Vehicles (UAVs) subject to Actuator and Sensor Faults," Master's Thesis, Departement of Electrical and Computer Engineering, Concordia Univresity, 2008.

[47] D. W. Casbeer, R. W. Beard, T. W. McLain, S. M. Li, and R. K. Mehra, "Forest fire monitoring with multiple small UAVs," Proceedings of the 2005 American Control Conference, pp. 3530- 3535, vol. 5, June 2005.

[48] D. W. Casbeer, D. B. Kingston, R. W. Beard, T. W. McLain, S. M. Li, and R. K. Mehra, "Cooperative Forest Fire Surveillance Using a Team of Small Unmanned Air Vehicles, International Journal of Systems Science, Volume 37, Issue 6, May 2006, pages 351-360.

[49] E. P. Anderson, "Extremal Control and Unmanned Air Vehicle Trajectory Generation, Master of Science Thesis, Brigham Young University, April 2002.

[50] K. B. Judd, "Trajectory Planning Strategies for Unmanned Air Vehicles," Masters thesis, Department of Mechanical Engineering, Brigham Young University, Provo, USA, 2001.

[51] L. P. Gewali, S. Ntafos, and I. G. Tollis, "Path planning in the presence of vertical obstacles," IEEE Transactions on Robotics and Automation, vol. 6, no. 3, pp. 331-341, Jun 1990.

[52] S. A. Bortoff, "Path planning for UAVs," Proceedings of the American Control Conference, vol. 1, no. 6, pp. 364-368 vol.1, Sep. 2000.

[53] T. McLain, and R. Beard, "Trajectory Planning for Coordinated Rendezvous of Unmanned Air Vehicles," Proceedings of the AIAA Guidance, Navigation, and Control Conference, Denver, Colorado, August 2000.

[54] P. R. Chandler, S. Rasmussen, and M. Pachter, "UAV Cooperative Path Planning," Proceedings of the AIAA Guidance, Navigation and Control Conference, and Exhibit, August 2000.

[55] A. A. ATA, "Optimal Trajectory Planning of Manipulators: A Review," Journal of Engineering Science and Technology Vol. 2, No. 1 (2007) 32-54.

[56] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". IEEE Transactions on Systems Science and Cybernetics SSC4 4 (2): 100-107, 1968.

[57] E. Dijkstra, "A note on two problems in connexion with graphs," Numerische mathematik, Springer, 1959.

[58] D. Eppstein, "Finding the k-th shortest paths," SIAM Journal of Computation, vol. 28, no. 2, pp. 652-673, 1999.

[59] E. Frazzoli, and F. Bullo, "Decentralized algorithms for vehicle routing in a stochastic time-varying environment," Proceedings of the 43rd IEEE Conference on Decision and Control, pp. 3357- 3363 Vol.4, Dec. 2004.

[60] A. Ahmadzadeh, J. Keller, G. J. Pappas, A. Jadbabaie, and V. Kumar, "An Optimization-based Approach to Time Critical Cooperative Surveillance and Coverage with Unmanned Aerial Vehicles, International Symposium on Experimental Robotics, 2006.

[61] N. Motee, and A. Jadbabaie, "Optimal Control of Spatially Distributed Systems," IEEE Transactions on Automatic Control, vol. 53, no. 7, pp. 1616-1629, Aug. 2008.

[62] A. Ahmadzadeh, G. Buchman, P. Cheng, A. Jadbabaie, J. Keller, V. Kumar, and G. Pappas, "Cooperative control of UAVs for search and coverage," In Proceedings of the conference on unmanned systems, AUVSI, Arlington, pp. 1-14, 2006.

[63] A. Ahmadzadeh, A. Jadbabaie, V. Kumar, and G. Pappas, "Multi-UAV Cooperative Surveillance with Spatio-Temporal Specifications," Proceedings of the 45th IEEE Conference on Decision and Control, pp. 5293-5298, Dec. 2006.

[64] P. DeLima, and D. Pack, "Toward developing an optimal cooperative search algorithm for multiple unmanned aerial vehicles," International Symposium on Collaborative Technologies and Systems, pp. 506-512, May 2008.

[65] P. DeLima, D. Pack, and J. C. Sciortino, "Optimizing a search strategy for multiple mobile agents, Proceedings of the SPIE, Volume 6563, pp. 65630B, 2007.

[66] Y. Mei, Y. H. Lu, Y. C. Hu, and C. S. G. Lee, "Deployment Strategy for Mobile Robots with Energy and Timing Constraints," Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pp. 2816- 2821, April 2005.

[67] Y. Mei, Y. H. Lu, Y. C. Hu, and C. S. G. Lee, "Deployment of mobile robots with energy and timing constraints," IEEE Transactions on Robotics, vol. 22, no. 3, pp. 507- 522, June 2006.

[68] M. F. Tompkins, "Optimization Techniques for Task Allocation and Scheduling in Distributed Multi-Agent Operations," Masters thesis, Massachusetts Institute of Technology, Cambridge, MA, June 2003.

[69] C. M. Fonseca, and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," Evolutionary Computation, 3(1): 1-16. 1995.

[70] Y. Ding, S. Gregov, O. Grodzevich, I. Halevy, Z. Kavazovic, O. Romanko, T. Seeman, R. Shioda, and F. Youbissi, (eds.) "Discussions on normalization and other topics in multi-objective optimization," Fields-MITACS, Fields Industrial Problem Solving Workshop. 2006.

[71] O. Grodzevich, and O. Romanko, "Normalization and other topics in multi-objective optimization," Proceedings of the first fields - MITACS industrial problems workshop. Toronto. pp. 89-101.

[72] D. Kingston, R. W. Beard, R. S. Holt, "Decentralized Perimeter Surveillance Using a Team of UAVs," IEEE Transactions on Robotics, vol. 24, no. 6, pp. 1394-1404 , Dec. 2008.

[73] S. Ponda, J. Redding, C. Han-Lim, J. P. How, M. Vavrina, and J. Vian, "Decentralized planning for complex missions with dynamic communication constraints," Proceedings of the American Control Conference, pp. 3998-4003, June-July 2010.

[74] C. Han-Lim, A. K. Whitten, and J. P. How, "Decentralized task allocation for heterogeneous teams with cooperation constraints," Proceedings of the American Control Conference, pp. 3057-3062, June-July 2010.

[75] A. Suzuki, and Z. Drezner, "The minimum equitable radius location problem with continuous demand, Proceedings of European Journal of Operational Research, 2009, pp.17-30.

[76] M. Pavone, K. Savla, and E. Frazzoli, "Sharing the load," IEEE Robotics and Automation Magazine, vol. 16, no. 2, pp. 52-61, June 2009.

[77] J. Corts, S. Martnez, and F. Bullo, "Spatially-distributed coverage optimization and control with limited-range interactions," ESAIM: Control, Optimization and Calculus of Variations, Vol. 11, pp. 691-719, 2005.

[78] N. Motee, A. Jadbabaie, and G. Pappas, "A duality approach to path planning for multiple robots," Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 935-940, May 2010.

[79] M. R. Doostmohammadian, H. Sayyaadi, and M. Moarref, "A novel consensus protocol using facility location algorithms," Proceedings of the IEEE Conference on Control Applications and Intelligent Control, pp. 914-919, July 2009.

[80] J. S. Kim, and B. K. Kim, Minimum-Time Grid Coverage Trajectory Planning Algorithm for Mobile Robots with Battery Voltage Constraints, In Proceedings of the International Conference on Control, Automation and Systems, Oct. 2010.

[81] S. S. Ge, and C. H. Fua, Complete Multi-Robot Coverage of Unknown Environments with Minimum Repeated Coverage, Proceedings of the 2005 IEEE International Conference on Robotics and Automation, April 2005.

[82] E. M. Arkin, M. A. Bender, E. D. Demaine, S. P. Fekete, J. S. B. Mitchell, and S. Sethia, "Optimal covering tours with turn costs," In Proceedings of the 13TH ACM-SIAM symposium on Discrete algorithms, 2003.

[83] J. S. Oh, Complete coverage navigation of cleaning robots using triangular-cell-based map, IEEE Transactions on Industrial Electronics, June 2004.

[84] Y. Mao, Combined complete coverage path planning for autonomous mobile robot in indoor environment, In Proceedings of the Asian Control Conference, Aug. 2009.

[85] M. Jager, and B. Nebel, "Dynamic Decentralized Area Partitioning for Cooperating Cleaning Robots," Proceedings of the 2002 IEEE international Conference on Robotics and Automation, May 2002.

[86] A. Agarwal, L. M. Hiot, N. T. Nghia, and E. M. Joo, "Parallel Region Coverage Using Multiple UAVs," In Proceedings of the 2006 IEEE Aerospace Conference, March 2006.

[87] J. G. Carlsson, B. Armbruster, and Y. Ye, "Finding Equitable Convex Partitions of Points in a Polygon Efciently," ACM Transactions on Algorithms (TALG), Volume 6 Issue 4, August 2010.

[88] Z. B. Alfred, A. A. Rizzi, and R. L. Hollis, Cooperative Coverage of Rectilinear Environments, In Fourth International Workshop on the Algorithmic Foundations of Robotics, 2000.

[89] H. Choset, and P. Pignon, Coverage path planning: the boustrophedon cellular decomposition, In Proceedings of the International Conference on Field and Service Robotics, December 1997.

[90] H. Choset, E. Acar, A. Rizzi, and J. Luntz, Exact cellular decompositions in terms of critical points of Morse functions, In Proceedings of the IEEE International Conference on Robotics and Automation, 2000.

[91] T. M. Chan, "A Fully Dynamic Algorithm for Planar Width," Journal of Discrete and Computational Geometry. 2003, 30(1): 17-24.

[92] W. Huang, The minimal sum of altitudes decomposition for coverage algorithms, Rensselaer Polytechnic Institute Computer Science Technical Report 00-3, June 2000.

[93] W. Huang, Optimal line-sweep-based decompositions for coverage algorithms, In Proceedings of the IEEE International Conference on Robotics and Automation, 2001.

[94] I. Maza, and A. Ollero, "Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms," Distributed Autonomous Robotic Systems 6, Springer, 2007.

[95] S. Hert , and V. Lumelsky, "Polygon Area Decomposition for Multiple-Robot Workspace Division," International Journal of Computational Geometry and Applications, Vol. 8, pp. 437-466, 1998.

[96] Y. S. Jiao, X. M. Wang, H. Chen, and Y. Li, "Research on the coverage path planning of UAVs for polygon areas," the 5th IEEE Conference on Industrial Electronics and Applications (ICIEA), June 2010.

[97] K. Deb, "Multi-objective Optimization using Evolutionary Algorithms, Wiley-Interscience series in systems and optimization. Wiley, 2002.

[98] K. Miettinen, "Nonlinear Multiobjective Optimizatio, Springer, 1999.

[99] K. Miettinen, "Introduction to Multiobjective Optimization: Noninteractive Approaches, Springer, 2008.

[100] V. Pareto, 1906: Manuale di Economica Politica, Societa Editrice Libraria . Milan, translated into English by A.S. Schwier as Manual of Political Economy , edited by A.S. Schwier and A.N. Page, New York: A.M. Kelley, 1971.

[101] T. W. Athan, and P. Y. Papalambros, "A note on weighted criteria methods for compromise solutions in multi-objective optimization," Engineering Optimization Vol. 27, pp. 155-176, 1996.

[102] W. Chen, A. Sahai, A. Messac, and G. Sundararaj, "Exploration of the effectiveness of physical programming in robust design," Journal of Mechanical Design Vol. 122, pp. 155-163. 2000.

[103] R. T. Marler, and J. S. Arora, "Survey of multi-objective optimization methods for engineering," Structural and multidisciplinary optimization, Springer, 2004.

[104] R. T. Marler, and J. S. Arora, "Review of Multi-Objective Optimization Concepts and Methods for Engineering" Technical Report Number ODL-01.01, University of Iowa, Optimal Design Laboratory, Iowa City, IA. 2003.

[105] W. S. Shin, and A. Ravindran, "Interactive multiple objective optimization: Survey I continuous case " Computers and Operations Research, Vol. 18, no. 1, pp. 97-114. 1991.

[106] T. L. Vincent, and W. J. Grantham, "Optimality in Parametric Systems," New York: John Wiley and Sons, 1981.

[107] K. A. Proos, G. P. Steven, O. M. Querin, and Y. M. Xie, "Multi-criterion evolutionary structural optimization using the weighted and the global criterion methods," AIAA J. 39, 2006-2012, 2001.

[108] W. Chen, M. M. Wiecek, and J. Zhang, "Quality utility - a compromise programming approach to robust design," Journal of Mechanical Design Vol. 121, pp. 179-187, 1999.

[109] J. Koski, "Multi-criterion optimization in structural design," In: E. Atrek, R. H. Gallagher, K. M. Ragsdell, and O. C. Zienkiewicz, (eds.) New Directions in Optimum Structural Design , pp. 483-503. New York: John Wiley and Sons, 1984.

[110] J. Koski, and R. Silvennoinen, "Norm methods and partial weighting in multicriterion optimization of structures," International Journal of Numerical Methods in Engineering, Vol. 24, pp. 1101-1121, 1987.

[111] L. Zadeh, "Optimality and non-scalar-valued performance criteria," IEEE Transactions on Automatic Control, Vol. 8, no. 1, pp. 59- 60, Jan 1963.

[112] S. S. Rao, and T. I. Freiheit, "A modied game theory approach to multi-objective optimization," Journal of Mechanical Design, Vol. 113, pp. 286-291, 1991.

[113] A. Osyczka, "An approach to multicriterion optimization problems for engineering design," Computer Methods in Applied Mechanics and Engineering Vol. 15, pp. 309-333. 1978.

[114] C. Coello, "A comprehensive survey of evolutionary-based multi-objective optimization techniques," Knowledge and Information systems, 1999.

[115] I.Y. Kim, and O.L. de Weck, "Adaptive weighted-sum method for bi-objective optimization: Pareto front generation," Structural and Multidisciplinary Optimization, Springer, 2005.

[116] R. T. Marler, and J. S. Arora, "The weighted sum method for multi-objective optimization: new insights" Structural and multidisciplinary optimization, Springer, 2010.

[117] J. H. Holland, "Adaptation in natural and articial systems," Ann Arbor: University of Michigan Press, 1975.

[118] J. S. Arora, O. A. Elwakeil, A. I. Chahande and C. C. Hsieh, "Global optimization methods for engineering applications: A review," Structural and Multidisciplinary Optimization, Volume 9, Numbers 3-4, pp. 137-159, 1995.

[119] A. Ravindran, K. M. Ragsdell, and G. V. Reklaitis, "Engineering optimization: Methods and applications," Wiley, New York, 1983.

[120] C. A. Floudas, and P. M. Pardalos, Eds., "State of the Art in Global Optimization: Computational Methods and Applications," Norwell, MA: Kluwer, 1996.

[121] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," IEEE Transactions on Evolutionary Computation, vol.6, no.2, pp.182-197, Apr 2002.

[122] J. Bredin, R. T. Maheswaran, C. Imer, T. Basar, D. Kotz, and D. Rus, "A Game Theoretic Formulation of Multi-Agent Resource Allocation," Proceedings of the fourth international conference on Autonomous agents, New York, 2000.

[123] S. S. Fatima, and M. Wooldridge, "Adaptive task resources allocation in multi-agent systems," Proceeding of the fifth international conference on Autonomous agents, ACM, New York, 2001.

[124] P. Faratin, C. Sierra, and N. R. Jennings, "Using similarity criteria to make issue trade-offs in automated negotiations," Artificial Intelligence, Vol. 142, Issue 2, December 2002, Pages 205-237.

[125] K. Leyton-Brown, "Resource Allocation in Competitive Multi-agent Systems," Ph.D. thesis, Stanford University, 2003.

[126] F. Aurenhammer, "Power diagrams: properties, algorithms and applications," SIAM Journal on Computing, 16(1):78-96, 1987.

[127] H. Imai, M. Iri, and K. Murota, "Voronoi diagram in the Laguerre geometry and its applications," SIAM Journal on Computing, 14(1):93-105, 1985.

[128] A. Hatcher, "Algebraic Topology," Cambridge University Press, Cambridge, U.K., 2002.

[129] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, "Spatial Tessellations: Concepts and Applications of Voronoi Diagrams," John Wiley and Sons, New York, NY, 2000.

[130] T. W. McLain, P. R. Chandler, M. Pachter, "A decomposition strategy for optimal coordination of unmanned air vehicles," Proceedings of the American Control Conference, pp. 369-373, Sep 2000.

[131] D. J. Bertsimas, G. V. Ryzin, "A Stochastic and Dynamic Vehicle Routing Problem in the Euclidean Plane," Operations Research , Vol. 39, No. 4 (Jul. - Aug., 1991), pp. 601-615.

[132] H. L. Choi, L. Brunet, and J. P. How, "Consensus-Based Decentralized Auctions for Robust Task Allocation," IEEE Transactions on Robotics, vol. 25, no. 4, pp. 912-926, Aug. 2009.

[133] S. Thompson, and S. Kagami, "Continuous Curvature Trajectory Generation with Obstacle Avoidance for Car-Like Robots," International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, vol. 1, pp. 863-870, Nov. 2005.

[134] A. Matlock, R. Holsapple, C. Schumacher, J. Hansen, and A. Girard, "Cooperative defensive surveillance using Unmanned Aerial Vehicles," Proceedings of the American Control Conference, pp. 2612-2617, June 2009.

[135] J. Yan, A. A. Minai, and M. M. Polycarpou, "Cooperative real-time search and task allocation in UAV teams," Proceedings of the 42nd IEEE Conference on Decision and Control, pp. 7- 12, Dec. 2003.

[136] D. A. Anisi, P. Ogren, H. Xiaoming, and T. Lindskog, "Cooperative surveillance missions with multiple unmanned ground vehicles (UGVs)," Proceedings of the 47th IEEE Conference on Decision and Control, pp. 2444-2449, Dec. 2008.

[137] A. Barili, M. Ceresa, and C. Parisi, "Energy-saving motion control for an autonomous mobile robot," Proceedings of the IEEE International Symposium on Industrial Electronics, vol. 2, pp. 674-676, Jul. 1995.

[138] D. Turra, L. Pollini, and M. Innocenti, "Fast unmanned vehicles task allocation with moving targets," Proceedings of the 43rd IEEE Conference on Decision and Control, pp. 4280- 4285 Vol.4, Dec. 2004.

[139] Z. Tang, and U. Ozguner, "Motion planning for multitarget surveillance with mobile sensor agents," IEEE Transactions on Robotics, vol. 21, no. 5, pp. 898-908, Oct. 2005.

[140] P. Y. Wu, D. Campbell, and T. Merz, "On-board multi-objective mission planning for Unmanned Aerial Vehicles," Proceedings of the 2009 IEEE Aerospace conference, pp. 1-10, March 2009.

[141] C. H. Caicedo-Nunez, and M. Zefran, "Performing coverage on nonconvex domains," Proceedings of the IEEE International Conference on Control Applications, pp. 1019-1024, Sep. 2008.

[142] L. F. Bertuccelli, H. L. Choi, P. Cho, and J. P. How, " Real-Time Multi-UAV Task Assignment in Dynamic and Uncertain Environments," Proceedings of the AIAA Guidance, Navigation, and Control Conference, p.1-16. August 2009.

[143] Y. S. Jiao, X. M. Wang, H. Chen, and Y. Li, "Research on the coverage path planning of UAVs for polygon areas," Proceedings of the 5th IEEE Conference on Industrial Electronics and Applications (ICIEA), pp. 1467-1472, June 2010.

[144] S. Ganeriwal, A. Kansal, and M. B. Srivastava, "Self aware actuation for fault repair in sensor networks," Proceedings of the IEEE International Conference on Robotics and Automation, pp. 5244- 5249, Vol. 5, April-May 2004.

[145] J. J. Enright, K. Savla, E. Frazzoli, and F. Bullo, "Stochastic and Dynamic Routing Problems for Multiple UAVs," AIAA Journal of Guidance, Control, and Dynamics, 34(4):1152-1166, 2009.

[146] D. J. Bertsimas, and G. J. van Ryzin, "Stochastic and dynamic vehicle routing in the Euclidean plane with multiple capacitated vehicles," Operations Research, 41(1):60-76, 1993.

[147] C. Schumacher, P. R. Chandler, S. J. Rasmussen, and D. Walker, "Task allocation for wide area search munitions with variable path length," Proceedings of the 2003 American Control Conference, pp. 3472- 3477, vol.4, June 2003.

[148] D. E. McClure, and R. A. Vitale, "Polygonal Approximation of Plane Convex Bodies," Journal of mathematical analysis and applications, Vol. 51, No. 2, 1975.

[149] U. Ramer, "An iterative procedure for the polygonal approximation of plane curves," Automata, Languages and Programming, Springer, 1990.