2017

Constrained path planning of unmanned vehicles

Yen-Chen Liu
Iowa State University

Constrained path planning of unmanned vehicles

by

Yen-Chen Liu

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Aerospace Engineering

Program of Study Committee:
Ran Dai, Major Professor
Christina Bloebaum
Leifur Leifsson

Iowa State University

Ames, Iowa

2017

DEDICATION

I would like to dedicate this thesis and the completion of this work to my family for their unwavering patience and financial assistance. I would also like to thank my friends for their loving support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would first like to acknowledge my advisor, Dr. Ran Dai, with my profound gratitude for her guidance and support leading me in the right direction throughout my years at Iowa State University. I would also like to acknowledge my committee members, Dr. Christina Bloebaum and Dr. Leifur Leifsson, for their input and time. I would also like to thank the staff of Automation and Optimization Laboratory, including YunQi Bang, Nathaniel Kingry, Matthew Martinez, Benjamin Simon, Chuangchuang Sun and Nicholaus Steffensmeier. Lastly, I must thank the Iowa State University, Department of Aerospace Engineering, for providing invaluable resources.

ABSTRACT

The application of unmanned system performing large-scale tasks, for instance, long-term surveillance/reconnaissance, large area sensing/mapping, and long distance materials handling is a relatively new and exciting topic. However, developing a practical system is still challenging due to complex models and hardware restriction. This manuscript explores various path planning missions from a more realistic perspective, such as point-to-point obstacle avoiding, multi-targets trajectory finding, informative motion planning, and multi-Hamiltonian Path Problem (mHPP) with two types of unmanned vehicles, Unmanned Ariel Vehicles (UAVs) and Unmanned Ground Vehicles (UGVs). These problems are formulated as classical optimization problems with constraints representing the environment and kinematic limitations, and then solved by proposed numerical or heuristic optimization approaches. The selected methods are used to handle nonlinear, discontinuous, and multi-objective formulations of the constrained mission planning problems. The feasibility and effectiveness of the proposed algorithms are inspected by the performance and comparison with other proposed methods in literature. The resulting simulations and experimental tests obtained from all the methods are demonstrated and discussed.

CHAPTER 1. INTRODUCTION

Study of autonomous robots has been a growing interest over the past few years, yet, producing an intelligent multi-tasking unmanned vehicle is still challenging. Different types of robots, such as Unmanned Underwater Vehicles (UUVs), Unmanned Ground Vehicles (UGVs), and Unmanned Ariel Vehicles (UAVs) can be deployed into various unexplored environments which can be abominable or hazardous. The motivation for the development of autonomous systems for unmanned vehicles is to accomplish complex missions far beyond the abilities for human beings to manage. However, the mission planning is often not only dependent upon the environment, but also limited by the characteristics of vehicles, such as the kinematics of UAV or the on-board battery of UGV. The awareness of all possible constraints is significant while constructing path planning missions for unmanned vehicles.

The work of this manuscript introduces four distinct path planning scenarios. The first case describes the most common, point-to-point problem motion planning of a UAV. That is, the vehicle is expected to travel from an initial point to a final point, whether in a known or unknown environment. Many path planning algorithms have been developed to search for a feasible trajectory under certain assumptions. In the past decade, optimal control theory has shown success generating flight paths and guiding UAVs to fly amongst pre-defined obstacles and No-Fly Zone [Jorris and Cobb (2009); Mattei and Blasi (2010)]. Recently, heuristic optimization algorithms have become widely used to explore a design space with absence of knowledge between two points due to its flexibility and efficiency [Buniyamin et al. (2011); Foo et al. (2006); Šišlák et al. (2009)]. An optimized heuristic method, such as the optimized Rapidly-Exploring Random Tree algorithm, named as RRT*, has been developed to search for obstacle avoiding paths with asymptotic optimality property [Karaman and Frazzoli (2011); Karaman (2011)]. However, the RRT* algorithm itself does not include the flight kinematic constraints, therefore,

a refined RRT* algorithm is proposed here to integrate the flight kinematic constraints to generate smooth paths.

The second case builds on the work of the first case, which solved a point-to-point collision-free path, and expands the environment to an obstacle-free environment with numerous target locations. The trajectory is segmented from waypoint to waypoint while still maintaining its continuity and satisfying the kinematic constraints of the UAV. For a multi-waypoint path planning problem, it requires the UAV or robot to perform the specified speed, position, and sequence computed by techniques, such as heuristic algorithms or polynomial methods, in order to reach all waypoints [Davies and Jnifene (2006); Hashim and Lu (2009); McGee and Hedrick (2006)]. Another extension of this work is evaluating decisions of the unknown sequence to pass through all waypoints from a given beginning location. However, it is difficult to decide the sequence unless it is pre-defined or evaluated separately from the dynamic system. New concepts for decision making have been proposed, for instance, hybrid optimal control system designed with switching phases while the vehicle is conducting different situations [Ross and D'Souza (2005); Soler et al. (2010)]. To handle the decision making of visiting sequences, along with the kinematics of the UAV, a set of hybrid decision variables, including both binary and continuous variables, is introduced. The problem is formulated into a form of general/nonconvex QCQP problems, and the flight path and the sequence can be solved simultaneously with an iterative rank minimization (IRM) algorithm.

Another type of autonomous vehicle used in this work is UGV. The third case assigns an UGV for an information gathering mission, where the problem assumed that the quality of information is uniform throughout the environment. Meanwhile, once the information has been collected it ceases to exist and does not regenerate. Due to the even quality of information, the problem becomes equivalent to an area coverage problem while the overlapped portion is eliminated. Recent applications have been done by Smith and Hollinger. Smith presents a linear temporal-logic (LTL) approach of optimal path planning for surveillance [Stephen Smith and Rus (2011)], and a Rapidly-exploring Information Gathering (RIG) method and its variants are proposed by Hollinger to maximize the gathered information with diverse information field to solve obstacle avoiding problem [Hollinger and Sukhatme (2014)]. Challenges of such missions

is that the operation time typically exceeds the limitations of the UGVs, so to continue operation, the UGVs must gather additional energy from other resources. Alternative approaches to extending the operational life of a vehicle by harvesting renewable energy, such as solar energy, have shown to be viable options. Research in the area of solar-powered robotics focuses on mission planning and developing energy awareness. For example, persistent operation for surveillance missions can be achieved by gathering solar energy along an optimally designed path [Vasisht and Mesbahi (2015)]. In addition, energy-aware path planning can be accomplished by creating solar maps using a Gaussian regression method [Plonski et al. (2013)]. A constrained RRT planning strategy is developed to monitor the constraints and maximize the performance.

The last case investigates the possibility of a cooperative multiple UGVs system as multi-robot systems have proven to be effective when performing large-scale tasks. These tasks generally require robots to operate in an environment for prolonged periods of time. When involving only a single robot in such tasks, the robotic system is generally subject to limited mobility and battery capacity. However, even a multi-robot system in these tasks cannot guarantee long duration operation that always meets the task requirements without external power supply. Recently, researchers have focused on extending the operational time of a robotic system by using a cooperative team of UGVs with a charging station. For example, work in Couture [Couture-Beil and Vaughan (2009)] proposed using a multi-robot system that executes a transportation task and periodically recharge from a docking station, which would attempt to incrementally improve its location. Similarly, another work in Wang [Wang et al. (2014)] examined a vehicle routing problem with energy constraints through a network where there are some charging sites within the network. Works in Keshmiri and Wawerla [Keshmiri (2011); Wawerla and Vaughan (2007)] examined practical and scalable control methods to address a multi-robot recharging problem. A novel approach, explored in this work, is to extend the operation time of a cooperative team of UGVs by incorporating a mobile charging station that has the capability of harvesting solar energy during operation. This solar-powered mobile charging station not only provides power for other vehicles, but also aids in completing the tasks the entire UGV team is attempting to accomplish. The variables to be determined in

the cooperative motion planning problem include the trajectories and associated velocities of each robot and the final destination where charging will occur. This problem is similar to a multi-depot multiple traveling salesperson problem (MmTSP) or a multiple Hamiltonian path problem (mHPP). Existing methods used to solve MmTSP or mHPP include ant colony optimization [Ghafurian and Javadian (2011)], genetic algorithm [Carter and Ragsdale (2006)], and approximation algorithm [Rathinam and Raja (2007)], just to name a few. However, none of these methods considers the dynamics of the traveling agents. The problem is examined under a cascaded heuristic optimization (CHO) approach integrating the Genetic Algorithm (GA) to solve the mHPP for objective function and then the Particle Swarm Optimization (PSO) to generate each UGVs velocities and power schedules along the path for strict energy constraints.

This thesis is orgaized as follows: Chapters 2-5 model the problems for the unmanned aerial vehicles with avoidance zones, the unmanned aerial vehicles with decision parameter, the information gathering using solar-powered unmanned ground vehicles, and a multi-robot team with a solar-powered charging station, respectively. The problem formulations and methodologies are also described in relative chapters. Then the computer simulations and experimental results for discussed path planning cases are demonstrated in Chapter 6. Lastly, the conclusion and final remarks are presented in Chapter 7.

CHAPTER 2. PATH PLANNING OF UNMANNED AERIAL VEHICLES WITH AVOIDANCE ZONES

This chapter presents a method for Unmanned Aerial Vehicle (UAV) path planning problems with avoidance zones. The objective is to generate a minimum time flight path with a specific speed while satisfying the initial and terminal constraints and avoiding no-flying zones of different shapes. The approach, based on heuristic search, refines the paths searched from optimized Rapidly-Exploring Random Tree (RRT*) algorithm to generate smoother paths satisfying flight kinematics.

2.1 Problem Formulation

The problem of solving a single UAV passing through hostile environments with avoidance zones is illustrated in Figure. 2.1, where the UAV has assigned starting and ending points, denoted as triangles. The avoidance zones have pre-defined locations and shapes. Depending on the requirements of the flight mission, the performance index can be assigned accordingly, i.e., minimum flight time.

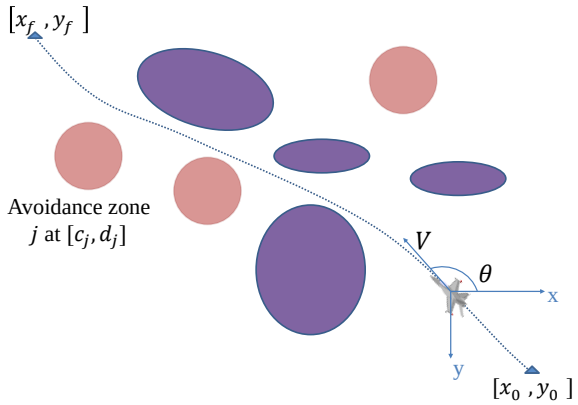


Figure 2.1 Illustration of UAV path planning problem with avoidance zones.

The flight kinematics of a UAV in two-dimensional space is represented by a single control model in the form of

$$\begin{aligned} \dot{x} &= V\cos\theta \\ \dot{y} &= V\sin\theta \\ \dot{\theta} &= u \\ |u| &\le u_{max}, \end{aligned} \tag{2.1}$$

where x and y are the coordinates, V is the specified cruise speed, θ is the heading angle, and u_{max} is the maximum rate of change of the heading angle . The starting and ending points are specified as $[x_0, y_0]$ and $[x_f, y_f]$. The avoidance zones are represented by ellipses formulated as,

$$\left(\frac{x-c_j}{a_j}\right)^2 + \left(\frac{y-d_j}{b_j}\right)^2 \ge 1, \forall\, j = 1, \ldots, m', \tag{2.2}$$

where $[c_j, d_j]$ represents the center of avoidance zone j, $j = 1, \ldots, m'$, (a_j, b_j) are pre-defined parameters, i.e., the semi-major axes of the elliptical zones, and m' is the number of avoidance zones. The above function can create different elliptical shapes (including rotated ones, for they are still quadratic) modeling a range of avoidance zones with specified centers. A special case is a circular-shaped zone by setting $a_j = b_j$. For minimum time of flight, the performance index is $J = \int_{t_0}^{t_f} 1dt$. Consequently, the minimum time path planning problems can be formulated as

$$\begin{aligned} & J = \min_u \textstyle\int_{t_0}^{t_f} 1dt \\ s.t. \quad & x(t_0) = x_0, y(t_0) = y_0, x(t_f) = x_f, y(t_f) = y_f, \\ & \dot{x} = V\cos\theta \\ & \dot{y} = V\sin\theta \\ & \left(\tfrac{x-c_j}{a_j}\right)^2 + \left(\tfrac{y-d_j}{b_j}\right)^2 \ge 1, \forall\, j = 1, \ldots, m', \forall x, y, t_0 \le t \le t_f \\ & \dot{\theta} = u \\ & |u| \le u_{max}. \end{aligned} \tag{2.3}$$

This manuscript propose a two steps heuristic approach to solve the UAV path planning problems posed in (2.3) to improve the computational performance. The first step solves a feasible path with modified RRT* from the pre-defined obstacles, then the second step optimizes the path with refined RRT* for flight kinematics constraints.

2.2 RRT* Algorithm

This section attempts to approach the path planning problem from an incremental sampling and goal-biased exploring aspect. The Rapidly-Exploring Random Tree (RRT) was first introduced by LaValle in 1998, which has a strong capability searching multi-dimensional design spaces with stochastic sampling process [Lavalle (1998)]. However, the original RRT algorithm does not guarantee convergence and local optimal, therefore, an Asymptotically Optimal Rapidly-Exploring Random Tree (RRT*) was developed by Karaman and Frazzoli in 2011 [Karaman and Frazzoli (2011)]. The RRT* has a goal-biased sampling process and a 'rewire' process, which is designed to reduce the cost of trees toward the goal. The comparison of RRT and RRT* is shown in Figure. 2.2. In this manuscript, a modified version of RRT* algorithm is proposed to fast generate a feasible path for the path planning problem with avoidance zones [Karaman (2011)]. A refinement RRT* is used subsequently to satisfy the UAV kinematics and minimize the flight time.

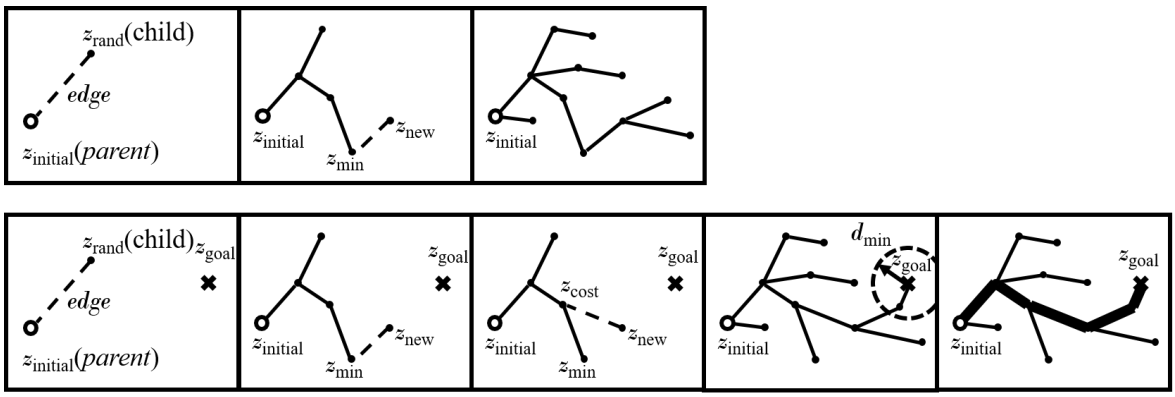


Figure 2.2 Comparison of RRT (up) and RRT* (down).

2.2.1 Modified RRT*

Let S represent the unexplored domain and $S \subset \mathbb{R}^n$. The obstacle region and obstacle-free region are denoted as S_{obs} and S_{free}, respectively, where $S_{\text{free}} = S \setminus S_{\text{obs}}$. Given the initial position, $z_{\text{initial}} = [x_0, y_0]$, and goal position, $z_{\text{goal}} = [x_f, y_f]$, the RRT* algorithm solves the motion planning problem by searching through the S_{free} area to yield a tree, T. Furthermore, an iterative process examines T from z_{initial} to z_{goal} and selects a feasible path, P, that satisfies all the given constraints.

A random position z_{rand} is generated by sampling the obstacle-free zone. It then determines its potential parent z_{min} and tests for collision before extending the tree. If the two points do not contact with obstacles in between, z_{rand} becomes z_{new} and a link is made between the two points. This position z_{new} now joins the tree with remark of its parent. If z_{rand} collides with obstacles on the way to z_{min}, it returns to sample a new random state and continue the algorithm. With sufficient iteration, the expansion of the tree should obtain at least one z_{new} that is close to z_{goal} while satisfying the condition that the distance between z_{new} and z_{goal} is less than d_{min}. The parameter, d_{min}, represents a defined minimum distance within which z_{new} and z_{goal} can be connected. In other words, at least one feasible path exists in T after the final node z_{goal} is added. The path P starts building up from z_{goal} and traces back to z_{initial} via the remarks of related parent from tree. Before discussing the algorithm, it is important to introduce a few necessary functions in the optimized RRT algorithm.

Sampling: The `sample` function returns a state, z_{rand}, that randomly samples from the desired field. The sampling process is assumed to be uniformly distributed among the selected nodes in the field such that $z_{\text{rand}} \in S_{\text{free}}$, $p_1 \in P$, as well as $p_2 \in P$.

Collision Evaluation: By introducing a set of nodes, the `Collision_eval` function forms an edge among two nodes and returns true if the entire edge occurs on the obstacle-free zone [LaValle (2006)].

Cost: The `cost` computation procedure evaluates the summation of distance from z_{rand} to z_{goal} and the distance from z_{rand} to any nearby nodes. When seeking for the minimum total distance during exploring, the edges of T tend to construct toward z_{goal} to accelerate the process of solving for a feasible solution.

Path Connecting: Given a tree with a feasible solution, i.e., $P \subset \mathbb{R}^n$ and $P \subset T$, the `Path_Connect` function backtracks the path from z_{goal} to each corresponding parents until it reaches z_{initial}.

In the RRT* algorithm, the tree begins with z_{initial} and an empty path and the steps are summarized as:

1) $T \leftarrow \{z_{\text{initial}}\}; P \leftarrow \{\}$

2) Iteration: While `isempty`(P) Do

2.1) A random position z_{rand} is generated by sampling the obstacle-free zone, $z_{\text{rand}} \leftarrow$ `Sample`(S_{free}).

2.2) It then determines its potential parent z_{min} by choosing the node with lowest overall cost from the tree, $z_{\text{min}} \leftarrow$ `Cost`(z_{rand}, T).

2.3) If the two points do not contact with obstacles in between, z_{rand} becomes z_{new} and a link is made between two points, denoted by $T \leftarrow T \cup \{(z_{\text{min}}, z_{\text{new}})\}$ for `Collision_eval` $(z_{\text{rand}}, z_{\text{min}})$ being true.

2.4) If `Collision_eval` $(z_{\text{rand}}, z_{\text{min}})$ is false which indicates z_{rand} collides with obstacles on the way to z_{min}, the algorithm returns step 2.3 to sample a new random state and continue the process, denoted by $z_{\text{new}} \leftarrow z_{\text{rand}}$.

2.5) With sufficient iteration, the expansion of the tree should obtain at least one z_{new} that is close to z_{goal} while satisfying the condition that the distance between z_{new} and z_{goal} is less than d_{min}, where d_{min} represents a defined minimum distance within which $T \leftarrow T \cup \{(z_{\text{new}}, z_{\text{goal}})\}$.

3) The path P starts building up from z_{goal} and traces back to z_{initial} via the remarks of related parent from trees, dented by $P \leftarrow$ `Path_Connect`(T).

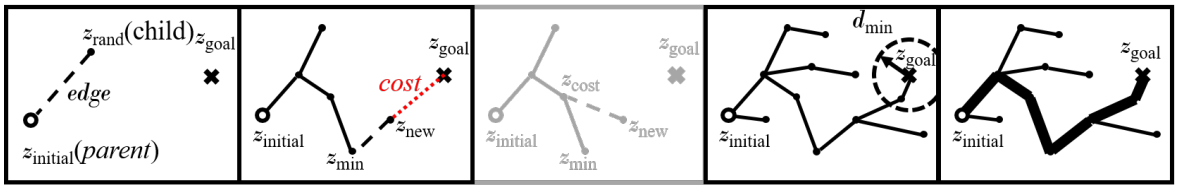


Figure 2.3 Example of modified RRT* steps.

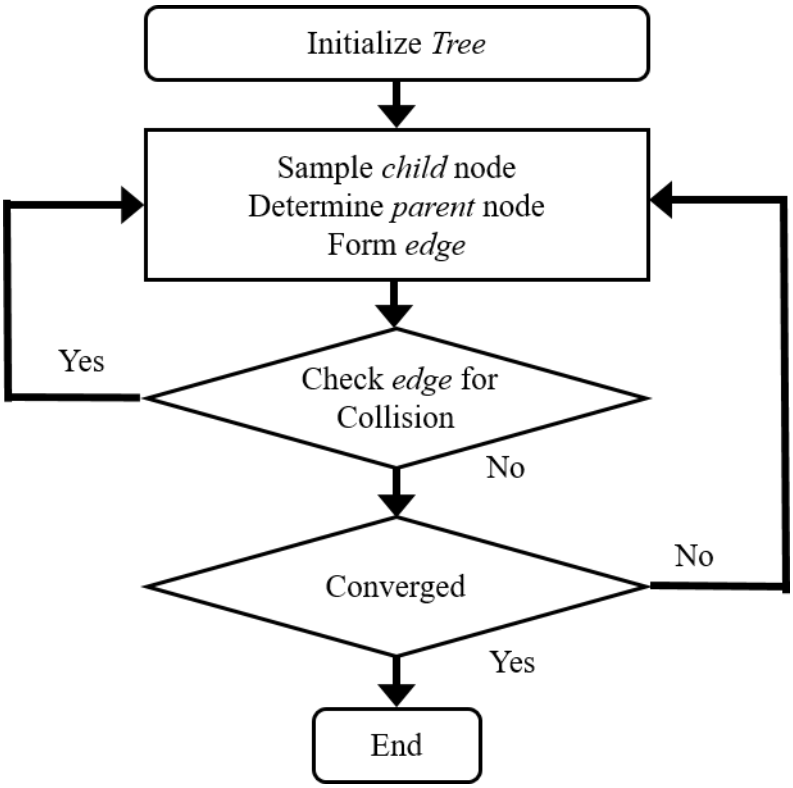


Figure 2.4 Flowchart of modified RRT* steps.

2.2.2 Refined RRT*

The RRT* algorithm stated above does not include the flight kinematics described in (2.1). In addition, there is more margin to shorten the entire path after connecting the feasible sampling trees. Two refinement steps are considered to further improve the performance index and smooth the path P obtained from RRT* by integrating the flight kinematic constraints.

Path Optimization: The `Path_Opt` function randomly samples two points on P and checks for any possible obstacle collision. If `Collision_eval` returns true, the new nodes p_1 and p_2 are added to P to replace any intermediate node. Meanwhile, a new edge between p_1 and p_2 is created as a substitution resulting in shortening the entire path length.

Path Refinement: In order to satisfy the limitation of maximum heading angle change rate, $u_{\max}$, the `Steering_eval` function verifies the heading angle change rate, defined as $\Delta\dot{\phi}$, between two connecting edges added in `Path_Opt` function. The change of heading angle is determined by the three ending points of two connecting edges located on a circular arc, as shown in Figure. 2.5. The heading angle rate is approximated by $\Delta\phi$ divided by the traveling time along the circular arc. The `Steering_eval` function returns true when all of the approximated heading angle change rates are no greater than $u_{\max}$.

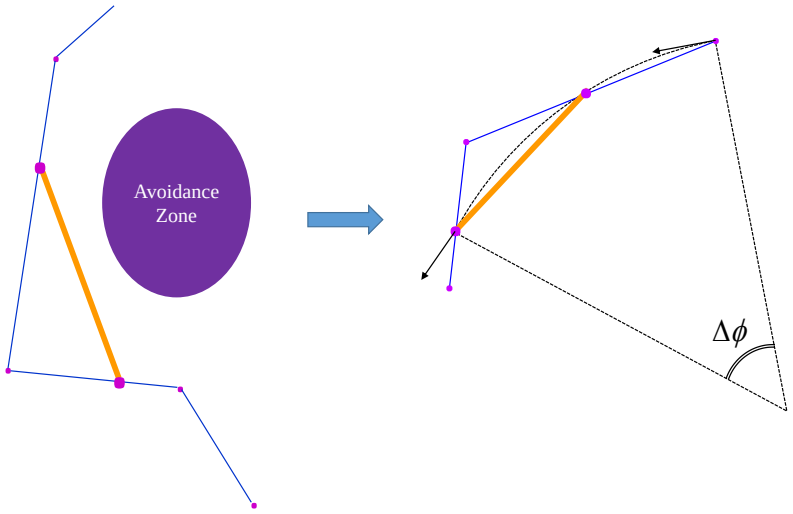


Figure 2.5 Example of refinement steps.

For step 3 in RRT* algorithm, the refinement steps are stated below:

4) Set `Path_Opt`$(P) \leftarrow RRT^*(P)$,

5) Iteration: while `Steering_eval`(P) is not true, $(p_1, p_2) \leftarrow$ `Sample`(P), If `Collision_eval` (p_1, p_2), then $P \leftarrow P \cup \{(p_1, p_2)\}$.

The replaced edges stated in the refinement step reduce the travel time and smooth the sharp turns by eliminating the violation of threshold ϕ. The loop is terminated once `Path_Opt`(P) is no longer empty. The refinement procedure is completely heuristic as well as the RRT* algorithm.

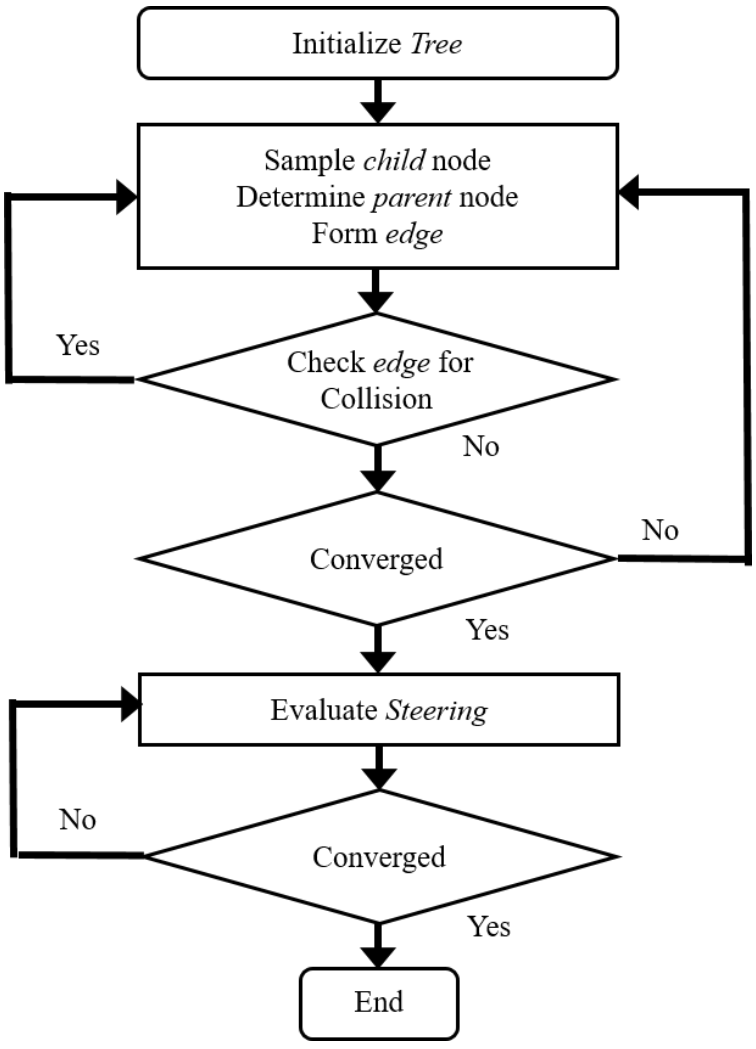


Figure 2.6 Flowchart of refined RRT* steps.

CHAPTER 3. PATH PLANNING OF UNMANNED AERIAL VEHICLES WITH MULTI-WAYPOINT AND DECISION PARAMETER

This chapter is interested in the path planning problem of an Unmanned Aerial Vehicle (UAV) for multiple target locations. The mission has specified the starting point, numerous waypoints, and kinematics of the UAV. An approach is proposed to determine the minimum time flight path and the visiting sequence for the waypoints by introducing a binary integer decision parameter and numerical optimization. Next, the path planning problem is reformulated as a general Quadratically Constrained Quadratic Programming (QCQP) problem and solved by an iterative convex optimization method.

3.1 Problem Formulation

The problem can be described as a single UAV passing through multiple given waypoints, where the UAV has an assigned starting point, but the sequence to reach all waypoints and ending point are to be determined. A set of decision parameters is distributed to each waypoint to evaluate the ending point and the order reaching the rest of the waypoints. The entire flight path is comprised of multiple sections equal to the number of waypoints, demonstrated in Figure 6.14. Depending on the requirements of the flight mission, the performance index can be assigned accordingly, i.e., minimum flight time.

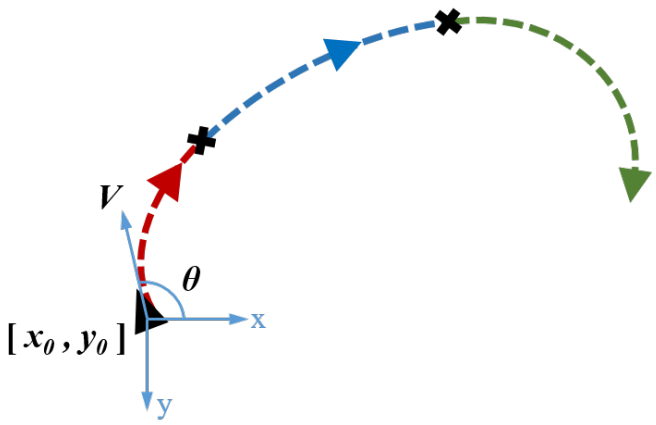


Figure 3.1 Example of path plan through multiple waypoints.

The flight kinematics of a UAV in two-dimensional space is represented by a unicycle model in the form of

$$\begin{aligned} \dot{x} &= V\cos\theta \\ \dot{y} &= V\sin\theta \\ \dot{\theta} &= u \\ |u| &\leq u_{\max}, \end{aligned} \tag{3.1}$$

where x and y are the coordinates, V is the specific cruise speed, θ is the heading angle, and $u_{\max}$ is the maximum heading angle changing rate. The starting point is specified as $[x_0, y_0]$. The decision parameters controlling the sequence are represented by the binary integer system

$$\gamma_{ab} \in \{0,1\}, \forall\ a = 1, ..., k \text{ and } b = 1, ..., l, \tag{3.2}$$

where k is number of waypoints and l is number of flight path sections. The case $\gamma_{ab} = 0$ indicates the decision that point a will be joined by the ending point of section b, and vice versa. The diverse path plans react to different values of decision parameters are illustrated in Figure 3.2. For minimum time flight, the performance index is determined by $J = \int_{t_0}^{t_f} 1dt$.

Consequently, the minimum time path planning problems can be formulated as

$$
\begin{aligned}
& J = \min_{x,y,\theta,u} \int_{t_0}^{t_f} 1 dt \\
s.t. \quad & x(t_0) = x_0, y(t_0) = y_0, \\
& \dot{x} = V\cos\theta \\
& \dot{y} = V\sin\theta \\
& \dot{\theta} = u \\
& |u| \le u_{\max} \\
& \gamma_{ab} \in \{0,1\}, \forall\ a = 1,...,k \text{ and } b = 1,...,l.
\end{aligned}
\tag{3.3}
$$

The above path planning problem formulated by nonlinear equations is difficult to solve. The indirect method requires deriving the necessary conditions for optimality based on Hamiltonian and Euler Lagrange equations. In most cases, the difficulty of guessing the initial adjoint variables and the trigonometric equations involved in the problem formulation make the indirect method infeasible. The direct method, such as collocation and NLP cannot guarantee fast convergence of a local optimal solution, even a feasible solution, when highly nonlinear equations are included in the constraints and/or an initial guess of the solution is randomly selected. Therefore, a novel approaches are proposed below to solve the UAV path planning problems with multi-waypoint and undetermined sequence.

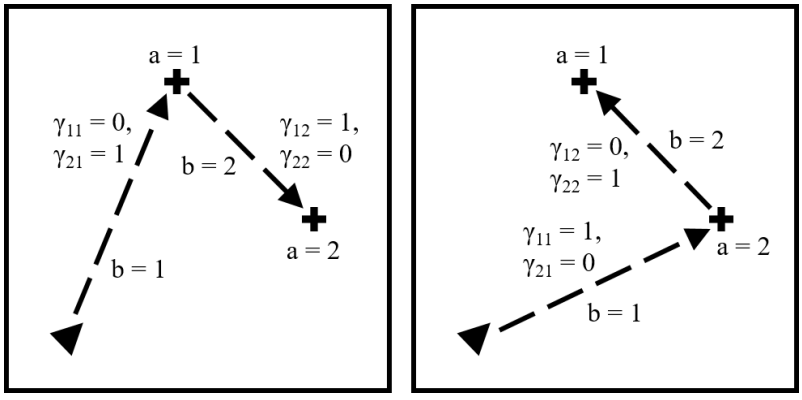


Figure 3.2 Path plans respect to different decision parameters.

3.1.1 Numerical Optimization Approach

The first step in numerical optimization approach is to convert the nonlinear optimization problem formulated in (3.3) into a general QCQP problem, where the objective is a quadratic function and the constraints are quadratic equalities or inequalities. An iterative convex optimization method is then introduced to solve the general QCQP problem. The novelty of QCQP formulation and its associated iterative method is that it does not involve linearization procedures in the formulation and optimization approach, such that errors generated from linearization of a highly nonlinear model are inevitable.

3.1.1.1 Reformulation of the Path Planning Problems as QCQP Problems

The unicycle model described in (3.1) for UAV path planning includes trigonometric functions, which are highly nonlinear and may generate singular matrices in computational operations. The first step toward solving the path planning problem with avoidance zones is to reformulate the above nonlinear optimization problems as general QCQP problems via discretization method. Each continuous flight path can be discretized into a series of segments represented by coordinates $[x_h, y_h]$, $h = 1, ..., H$, at each node, where H is the number of discrete nodes. By discretization, the change rate of the coordinates can be approximately determined by two adjacent nodes,

$$\dot{x} = \frac{x_{h+1}-x_h}{\Delta t} = V\cos\theta,\ h = 1, ..., H, \tag{3.4}$$

$$\dot{y} = \frac{y_{h+1}-y_y}{\Delta t} = V\sin\theta,\ h = 1, ..., H, \tag{3.5}$$

where Δt is the uniform time interval between two adjacent nodes. The above two equations can be synthesized as

$$(x_{h+1} - x_h)^2 + (y_{h+1} - y_h)^2 = V^2(\Delta t)^2,\ h = 1, ..., H. \tag{3.6}$$

Differentiating (3.4)-(3.5) leads to

$$\ddot{x} = \frac{x_{h+2}+x_h-2x_{h+1}}{(\Delta t)^2} = -V\dot{\theta}\sin\theta,\ h = 1, ..., H,$$
$$\ddot{y} = \frac{y_{h+2}+y_h-2y_{h+1}}{(\Delta t)^2} = V\dot{\theta}\cos\theta,\ h = 1, ..., H.$$

Combining with $\dot{\theta} = u$ and $|u| \leq u_{\max}$, the above equations can be synthesized as

$$\dot{\theta}^2 = \left[\frac{x_{h+2} + x_h - 2x_{h+1}}{(V\Delta t)^2}\right]^2 + \left[\frac{y_{h+2} + y_h - 2y_{h+1}}{(V\Delta t)^2}\right]^2 \leq u_{\max}^2.$$

By introducing an additional variable $t' = \Delta t^2$, the above constraint can be reformulated as a quadratic inequality in the form of

$$(x_{h+2} + x_h - 2x_{h+1})^2 + (y_{h+2} + y_h - 2y_{h+1})^2 \leq V^2 u_{\max}^2 t'^2,$$
$$h = 1, ..., H-2.$$

The binary system described in (3.2) can also be further expressed as equality and inequality constraints.

$$\gamma_{ab}(\gamma_{ab} - 1) = 0, \forall\ a = 1, ..., k \text{ and } b = 1, ..., l.$$
$$\textstyle\sum_{p=1}^{k} \gamma_{ap} = k - 1, \forall\ a = 1, ..., k,$$
$$\textstyle\sum_{q=1}^{l} \gamma_{qb} = l - 1, \forall\ b = 1, ..., l,$$
$$\gamma_{ab}^2 \geq \frac{(x_a - x_{f_b})^2 + (y_a - y_{f_b})^2}{M}, \forall\ a = 1, ..., k \text{ and } b = 1, ..., l,$$

where $[x_a, y_a]$ represents the given waypoints, $[x_{f_b}, y_{f_b}]$ are the discretized final point of section b, and $M \in \mathbb{R}$ is a pre-defined constant. The summation of decision parameters for each waypoint, γ_{ap}, equal to $k-1$ implies that one waypoint can only be connected by one section. Meanwhile, the summation of decision parameters for each section, γ_{qb}, equivalent to $l-1$ implies that one section would only send to one waypoint. While the final point of each section is getting close to waypoints, i.e., the distance between $[x_{f_b}, y_{f_b}]$ and $[x_a, y_a]$ is approaching zero, if the connection is allowed and the flight kinematics are satisfied then the value of corresponding γ_{ab} would be 0 and the rest of related decision parameters would return 1. The parameter M is set to be a large number to keep the distance ratio always less than 1, therefore, the decision parameters are restricted to not be greater than 1. Based on the above reformulation, the path planning problems with multiple sections can be generalized as a nonconvex QCQP problem in the form of

$$
\begin{aligned}
& J = \min_{x,y,\theta,u} \int_{t_0}^{t_f} 1dt \\
s.t. \quad & x(t_0) = x_0, y(t_0) = y_0, \\
& (x_{h+1,b} - x_{h,b})^2 + (y_{h+1,b} - y_{h,b})^2 = V^2(\Delta t_b)^2, \\
& h = 1, ..., H, \forall\ b = 1, ..., l, \\
& (x_{h+2,b} + x_{h,b} - 2x_{h+1,b})^2 + (y_{h+2,b} + y_{h,b} - 2y_{h+1,b})^2 \\
& \leq V^2 u_{\max}^2 {t'_b}^2, h = 1, ..., H-2, \forall\ b = 1, ..., l, \\
& \gamma_{ab}(\gamma_{ab} - 1) = 0, \forall\ a = 1, ..., k \text{ and } b = 1, ..., l, \\
& \textstyle\sum_{p=1}^{k} \gamma_{ap} = k - 1, \forall\ a = 1, ..., k, \\
& \textstyle\sum_{q=1}^{l} \gamma_{qb} = l - 1, \forall\ b = 1, ..., l, \\
& \gamma_{ab}^2 \geq \frac{(x_a - x_{f_b})^2 + (y_a - y_{f_b})^2}{M}, \forall\ a = 1, ..., k \text{ and } b = 1, ..., l, \\
& t' = (\Delta t_b)^2, \forall\ b = 1, ..., l.
\end{aligned}
$$

To retain continuity between two adjacent sections, the velocity components of the last point of the previous section and the first point of the current section need to stay the same. Additionally, the heading angle changing rate needs to satisfy the inequality constraint as well, which is approximated by the last two points of the previous section and the first point of the current section. Based on the above reformulation, the path planning problems with avoidance zones can be generalized as a nonconvex QCQP problem in the form of

$$
\begin{aligned}
& J = \min_{\mathbf{x}} \mathbf{x}^T Q_0 \mathbf{x} + {a_0}^T \mathbf{x} \\
s.t. \quad & \mathbf{x}^T Q_j \mathbf{x} + {a_j}^T \mathbf{x} \leq c_j, \forall\ j = 1, ..., m, \\
& l_{\mathbf{x}} \leq \mathbf{x} \leq u_{\mathbf{x}},
\end{aligned}
\tag{3.7}
$$

where $\mathbf{x} \in \mathbb{R}^n$ is the unknown vector to be determined, $Q_j \in \mathbb{S}^{n \times n}$, $j = 0, ..., m$, is an arbitrary symmetric matrix, $c_j \in \mathbb{R}$, $j = 1, ..., m$, and $a_j \in \mathbb{R}^n$, $j = 0, ..., m$. Moreover, $l_{\mathbf{x}} \in \mathbb{R}^n$ and $u_{\mathbf{x}} \in \mathbb{R}^n$ are the lower and upper bounds on $\mathbf{x}$, respectively. Since $Q_j\,(j = 0, ..., m)$ is not necessarily a positive definite matrix, problem in (3.7) is classified as NP-hard.

3.1.1.2 An Iterative Approach for General QCQP Problems

Efforts toward solving general QCQP problems have been focused on finding the bounds on the optimal values by linear or semidefinite relaxation [Vandenberghe and Boyd (1996)]. Although randomization and linearization have been used to find an approximate solution, neither approach guarantees optimality of the approximate solution with a determined convergence rate [Luo et al. (2010)].

The above QCQP problem with inhomogeneous quadratic function can be transformed into homogeneous ones by introducing a new variable $\alpha \in \mathbb{R}$ and a new quadratic constraint $\alpha^2 = 1$ by the following formulation

$$\begin{aligned} J = \min & \begin{bmatrix} \mathbf{x}^T & \alpha \end{bmatrix} \begin{bmatrix} Q_0 & a_0/2 \\ a_0^T/2 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \alpha \end{bmatrix} \\ s.t. \quad & \begin{bmatrix} \mathbf{x}^T & \alpha \end{bmatrix} \begin{bmatrix} Q_j & a_j/2 \\ a_j^T/2 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \alpha \end{bmatrix} + c_j \leq 0, \\ & \forall\, j = 1, ..., m, \\ & l_{\mathbf{x}} \leq \mathbf{x} \leq u_{\mathbf{x}}, \\ & \alpha^2 = 1. \end{aligned} \tag{3.8}$$

Then $\mathbf{x}^*/\alpha^*$ will be a solution of the original problem stated in (3.7), while $(\mathbf{x}^*, \alpha^*)$ is the solution pair of (3.8). In addition, linear constraints in (3.7) can be rewritten in the above quadratic form as well by setting corresponding matrix $Q = \mathbf{0}$. The homogeneous QCQP problem is formulated as

$$\begin{aligned} J = \min & \, \mathbf{x}^T Q_0 \mathbf{x} \\ s.t. \quad & \mathbf{x}^T Q_j \mathbf{x} \leq c_j, \forall\, j = 1, ..., m. \end{aligned} \tag{3.9}$$

Based on this fact, any inhomogeneous QCQP can be transformed into a homogeneous one. Without loss of generality, the following approach to solve nonconvex QCQP problems focuses on homogeneous QCQPs.

After introducing a rank one positive semidefinite matrix, $X = \mathbf{x}\mathbf{x}^T$, the SDP relaxation method is formulated as

$$\begin{aligned} & J = \min_X \langle X, Q_0 \rangle \\ s.t. \quad & \langle X, Q_j \rangle \leq c_j, \forall\, j = 1, ..., m, \\ & X \succeq 0, \end{aligned} \tag{3.10}$$

where '$\langle \cdot \rangle$' denotes the inner product of two matrices, i.e., $\langle A, B \rangle = \mathbf{trace}(A^T B)$. By reformulating the QCQP problem as a relaxed SDP problem, one can obtain a lower bound on its optimal value. However, the relaxation method will not yield an optimal solution of the unknown variables $\mathbf{x}$. The only difference between the SDP relaxation and the equivalent conversion is that the rank-one constraint, $X = \mathbf{x}\mathbf{x}^T$, is excluded in SDP relaxation. In order to obtain the optimal solution of $\mathbf{x}$, past work reconsider the rank-one constraint on matrix $\mathbf{X}$ and propose an alternative approach to gradually approach the rank constraint [Dai and Sun (2015); Sun and Dai (2015)].

A matrix with rank-one has only one nonzero eigenvalue. Therefore, instead of putting a constraint on the rank, we focus on constraining the eigenvalues of $\mathbf{X}$ to enforce the $n-1$ smallest eigenvalues of $\mathbf{X}$ are all zero. It has been proven that, when $\mathbf{X}$ is a nonzero, positive semidefinite matrix, $\mathbf{X}$ is a rank-one matrix if and only if $r\mathbf{I}_{n-1} - \mathbf{V}^T\mathbf{X}\mathbf{V} \succeq 0$, where $\mathbf{V} \in \mathbb{R}^{n\times(n-1)}$ are the eigenvectors corresponding to the $n-1$ smallest eigenvalues of $\mathbf{X}$, $\mathbf{I}_{n-1} \in \mathbb{R}^{(n-1)\times(n-1)}$ is an identity matrix, and r is a significantly small positive number. In other words, a substitute for the rank one constraint on $\mathbf{X}$ is the semidefinite constraint $r\mathbf{I}_{n-1} - \mathbf{V}^T\mathbf{X}\mathbf{V} \succeq 0$, where $r \to 0$. However, until we can find the optimal solution of $\mathbf{X}$, we cannot obtain the exact $\mathbf{V}$ matrix that is dependent on $\mathbf{X}$. In the recent work [Dai and Sun (2015)], an IRM method was proposed to gradually minimize the rank of $\mathbf{X}$. At each step, we try to optimize the original objective function while at the same time minimizing the weighted parameter r such that, when $r = 0$, the rank one constraint on $\mathbf{X}$ will be satisfied. The relaxed problem we are solving at each iteration step, k, of IRM is formulated as a convex optimization problem in the form of

$$
\begin{aligned}
& J = \min_{X_k, r_k} \langle X_k, Q_0 \rangle + w^k r_k \\
s.t. \quad & \langle X_k, Q_j \rangle \leq c_j, \forall\, j = 1, ..., m, \\
& X_k \succeq 0 \\
& r_k I_{n-1} - V_{k-1} X_k V_{k-1} \succeq 0,
\end{aligned}
\tag{3.11}
$$

where $w > 1$ is the weighting factor for r_k. Obviously, the solution at the converged point satisfies the rank one constraint on $\mathbf{X}$ as well as other constraints described in the equivalent QCQP problem. Through the Karush-Kuhn-Tucker conditions [Borwein and Lewis (2010)], we then have attained linear convergence to at least a local optimum of the proposed IRM method. More detailed derivation and proof of linear convergence can be referred to the past work [Dai and Sun (2015); Sun and Dai (2015)].

Different from the NLP, method which can be applied to solve general nonlinear optimization problems, the IRM algorithm above does not require initial guess of the unknown variables. The IRM algorithm is initiated by solving problem (3.10) without considering the rank constraint to obtain V_0 from X_0 for $k = 0$. It then iteratively solves problem (3.11) to obtain X_k and update V_k until $r_k \leq \delta$, where δ is a stopping threshold. Furthermore, except the newly introduced variable r_k, there is no extra introduced unknown variable in the formulation. This simple procedure can be easily implemented for any general QCQP problem. Along with the quadratic formulations of UAV fight kinematics and the binary decision parameters, the IRM method is applied here to solve the multi-waypoint path planning problem.

CHAPTER 4. PATH PLANNING FOR INFORMATION GATHERING USING SOLAR-POWERED UNMANNED GROUND VEHICLES

This chapter focuses on surveillance mission planning and power management of a solar-powered Unmanned Ground Vehicle (UGV). The task of the UGV is to maximize the exploration of an unknown region to gather information from the environment via on-board equipment. Meanwhile, the UGV is required to utilize the ambient solar energy from the environment and operate under specified net energy constraint within a limited traveling time. The modified Rapidly-Exploring Random Tree (RRT) methods is proposed to search for a locally optimized path for the assigned mission.

4.1 Problem Formulation

Consider a solar-powered UGV with known performance characteristics which must survey as much of a two-dimensional area as possible, such as a field or other area of interest, in a predetermined allotted time. Given priori knowledge of the solar energy distribution of the area, the UGV must survey this area in such a way where the UGV's total energy expenditure is less than or equal to what is gained via the environment. This UGV's movement is governed by four differential drive primitives (forward, backward, clockwise rotation, and counterclockwise rotation), which is defined as a Balkcom-Mason curve [Balkcom and Mason (2002)]. The UGV travels a path defined by n waypoints, $n-1$ line segments, and $n-1$ rotations, which can be seen in Figure 4.1.

For this problem, the information in the area is a set of nodes that cover the area in a way inspired by FCC structures of metals. However, to ensure that all the area is given value in the problem, the circles are overlapped more than the actual FCC structure will allow. Figure 4.2 shows an idealized version used in our implementation compared to a true FCC structure.

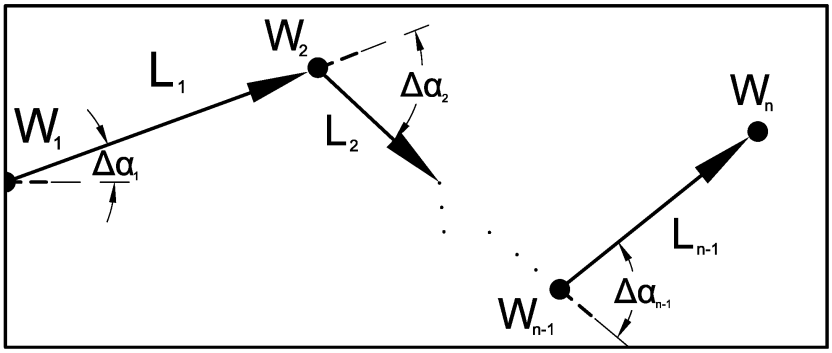


Figure 4.1 Example of Differential-Drive Motion Plan.

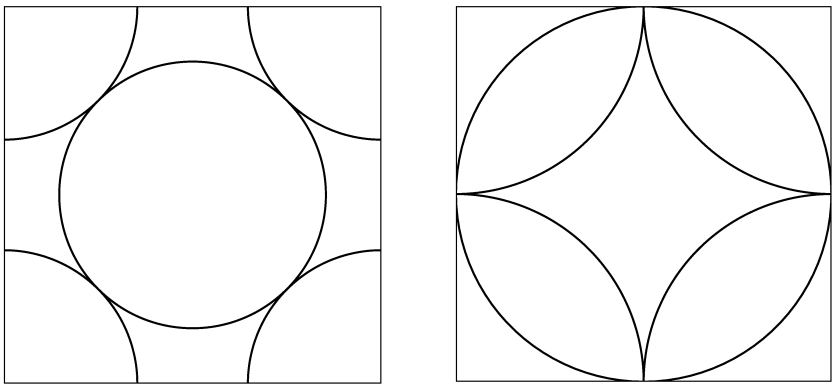


Figure 4.2 Actual FCC Structure (Left) vs. Ideal Structure (Right).

Having the nodes uniformly placed in the area, all with equal weighting of importance, the percent information gathered from the area can be found by finding the nodes that are a within a specified distance, r, defined as the detection radius and for our problem only slightly larger than the platform area of the UGV. This can be mathematically defined as

$$Information_{gathered} = \frac{\sum_{i=1}^{\#ofNodes} \sum_{j=1}^{n-1} \left(\frac{|a_j x_i + b_j y_i + c_j|}{\sqrt{a_j^2 + b_j^2}} \leq r \right)}{\#\ of\ Nodes} \times 100 \tag{4.1}$$

where $ax + by + c = 0$ is the equation of the line in slope-intercept form and (x_i, y_i) is the coordinate of the information node, and n is the number of waypoints in the path. Alternatively, a more traditional approach to the information problem, is determining how much area is covered by the vehicle. In the same way that the information gathered in the nodal method, the same detection radius, r, is used. The detection radius is used to bound the path and determine how much area is covered by the rover along its path. Along each portion of the

path, area that is covered multiple times or that is outside of the area of interest, must be subtracted out. To begin to determine the area coverage, the area the circles centered on each waypoint can be found

$$A_{\mathrm{A}} = n(\pi r^2) \tag{4.2}$$

where n is the number of waypoints and r is the detection radius. Then for every line segment the distance of the path can be used to easily obtain the area between the two waypoints, ensuring that the circles are not included

$$A_{\mathrm{B}} = \sum_{i=2}^{n-1}(2rd_i - \pi r^2) \tag{4.3}$$

where n is the number of waypoints and d is the distance between the two waypoints. Next, the area that is repeated or outside the area can be determined. The overlap from turning can be calculated as

$$A_{\mathrm{C}} = \sum_{i=2}^{n-2}\Big[r^2\frac{\sin(\frac{\pi}{2}-\frac{\alpha_i}{2})}{\sin\frac{\alpha_i}{2}} - \pi r^2\frac{\pi-\alpha_i}{2\pi}\Big] \tag{4.4}$$

Finally, the portion of the initial waypoint that is outside the bounds can be found as

$$A_{\mathrm{D}} = \frac{1}{2}r^2\Big[\frac{\sin\alpha_1}{\sin(\frac{\pi}{2}-\alpha_1)} + \frac{\sin(\frac{\pi}{2}-\alpha_1)}{\sin\alpha_1} - \pi\Big] \tag{4.5}$$

where α is the turning angle. Finally, the total area covered can be found from the summation of each area

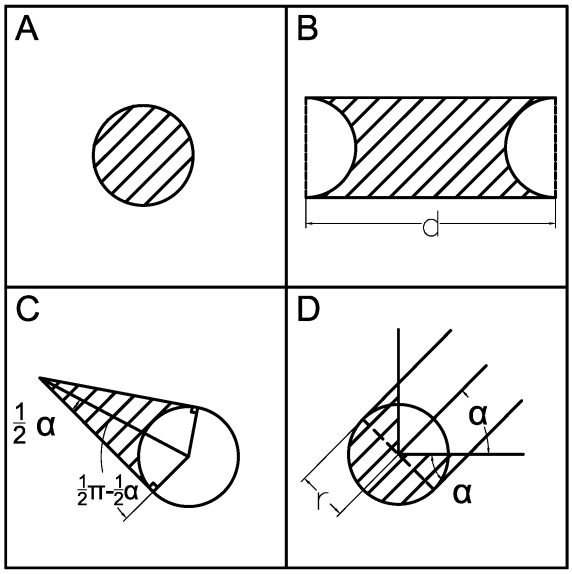
$$A_{\mathrm{Total}} = A_{\mathrm{A}} + A_{\mathrm{B}} - A_{\mathrm{C}} - A_{\mathrm{D}} \tag{4.6}$$

An example of this idea can be seen in Figure 4.3

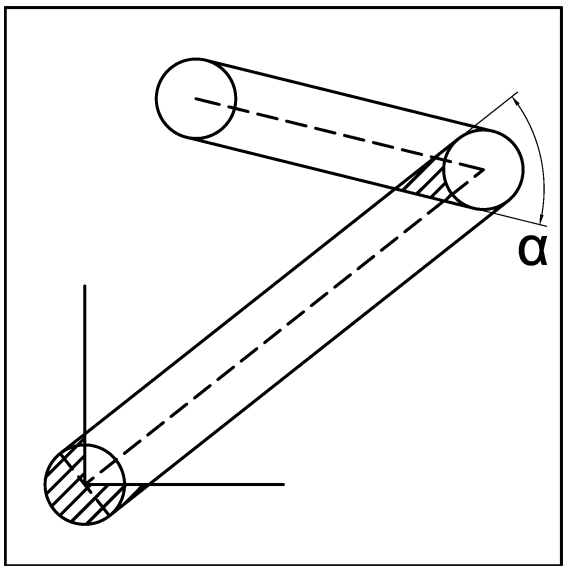
The time required for the UGV to complete its motion plan from the starting point $W_1(x_1, y_1)$ to the ending point $W_n(x_n, y_n)$, is denoted by T_{Total}. T_{Total} is defined as the sum of the time required to complete each rotation and line segment of the motion plan, which is mathematically defined as

$$T_{\mathrm{Total}} = \sum_{i=1}^{n-1}(T^r(i) + T^l(i)) \tag{4.7}$$

where the superscripts 'r' and 'l' denote the rotation and linear segment, respectively, i is the index of segment, and n is the total number of waypoints including the starting and ending



(a)



(b)

Figure 4.3 Explanation of Area Subtraction (Left) and Example of Path with Area Coverage (Right)

points. For each rotational and linear segment, the corresponding average angular speed $\omega(i)$ and linear speed $V(i)$, the T_{Total} can be further written as

$$T_{\text{Total}} = \sum_{i=1}^{n-1} \left(\frac{\alpha(i)}{\omega(i)} + \frac{\beta(i)}{V(i)} \right) \tag{4.8}$$

where $\alpha(i)$ is the change in heading direction from the current to the next prescribed heading and $\beta(i)$ is the length of the line segment i.

The energy gathered by the UGV's solar panel, E_{in}, is dependent on the solar panel's surface area denoted as S_{panel} , and the environment's solar energy density, denoted as $R_{\text{in}}(x, y)$. The environment's solar energy density, $R_{\text{in}}(x, y)$, is a scalar field interpolation constructed from discrete solar energy samples. To calculate the energy gathered along the path, Q, equidistant spaced samples over the line to determine the average energy strength. For rotational segments, the available energy density may be approximated as that at the location of the turn. From this, the energy gathered along each segment is expressed as

$$E^l_{\text{in}}(i) = \sum_{k=1}^{Q} R_{\text{in}}(x_k, y_k) \frac{S_{\text{panel}} \beta(i)}{V(i) Q} \tag{4.9}$$

$$E^r_{\text{in}}(i) = R_{\text{in}}(x(i), y(i)) \frac{S_{\text{panel}} \alpha(i)}{\omega(i)} \tag{4.10}$$

The energy consumed by the UGV to complete each portion of the motion plan is determined by the engine consumption rate $P^l_{\text{e}}(i)$ and $P^r_{\text{e}}(i)$, for the linear and rotational segments respectively, the passive power, which is defined to be the power allocated to the microcontroller, wireless sensor, and idle power (denoted by P_{passive}), and the T_{Total} . From these defined characteristics, the consumed energy can be represented by

$$E^l_{\text{out}}(i) = (P^l_{\text{e}}(i) n_{\text{e}} + P_{\text{passive}}) T^l(i) \tag{4.11}$$

$$E^r_{\text{out}}(i) = (P^r_{\text{e}}(i) n_{\text{e}} + P_{\text{passive}}) T^r(i) \tag{4.12}$$

where n_{e} is the number of engines. Each engine consumption rate under linear and rotational movements is a polynomial function of corresponding linear and angular speed, denoted as $P^l_{\text{e}}(V(i))$ and $P^r_{\text{e}}(\omega(i))$, respectively. These polynomial functions were characterized during experimental testing. The strict net energy constraint imposed over the entire path, net energy

gain during the motion plan is above zero, can be expressed as

$$\sum_{i=1}^{n-1}(E_{\text{in}}^{l}(i) - E_{\text{in}}^{r}(i)) \geq \sum_{i=1}^{n-1}(E_{\text{out}}^{l}(i) + E_{\text{out}}^{r}(i)) \tag{4.13}$$

At any time along the path, the UGV's power for its components may be supplied from either the gained energy from the environment or from the on-board battery denoted as P_{b}. In other words, if the power provided by the UGV's solar panel is less than is required by the motors and systems, the power will be drawn from the on-board battery. Following, any excess energy supplied from the solar panel will be stored in the on-board battery, as long as the batterys energy is within its minimum and maximum values, $P_{\text{b}_{\min}}$ and $P_{\text{b}_{\max}}$. With this, the power balance relationship is formulated as

$$P_{\text{e}}^{l}(i)n_{\text{e}} + P_{\text{passive}} + P_{\text{b}}^{l}(i) - \sum_{k=1}^{Q} R_{\text{in}}(x_k, y_k)\frac{S_{\text{panel}}}{Q} = 0 \tag{4.14}$$

$$P_{\text{e}}^{r}(i)n_{\text{e}} + P_{\text{passive}} + P_{\text{b}}^{r}(i) - R_{\text{in}}(x(i), y(i))S_{\text{panel}} = 0 \tag{4.15}$$

where $P_{\text{b}_{\min}} \leq P_{\text{b}}^{l}(i) \leq P_{\text{b}_{\max}}$ and $P_{\text{b}_{\min}} \leq P_{\text{b}}^{r}(i) \leq P_{\text{b}_{\max}}$.

The motion planning problem for the information gathering UGV with net energy constraint can be summarized as an optimization problem in the form of

$$\begin{aligned}
&\max_{x(i),y(i),\omega(i),V(i)} Node_{gathered} \text{ or } A_{\text{Total}} \\
s.t. \quad &\textstyle\sum_{i=1}^{n-1}(T^{r}(i) + T^{l}(i)) - 60 \leq 0 \\
&\textstyle\sum_{i=1}^{n-1}(E_{\text{in}}^{r}(i) + E_{\text{out}}^{r}(i)) + \sum_{i=1}^{n-1}(E_{\text{in}}^{l}(i) + E_{\text{out}}^{l}(i)) \geq 0 \\
&P_{\text{e}}^{r}(i)n_{\text{e}} + P_{\text{passive}} + P_{\text{b}}^{r}(i) - R_{\text{in}}(x(i), y(i))S_{\text{panel}} = 0 \\
&P_{\text{e}}^{l}(i)n_{\text{e}} + P_{\text{passive}} + P_{\text{b}}^{l}(i) - \textstyle\sum_{k=1}^{Q} R_{\text{in}}(x_k, y_k)\frac{S_{\text{panel}}}{Q} = 0 \\
&P_{\text{b}_{\min}} \leq P_{\text{b}}^{r}(i) \leq P_{\text{b}_{\max}} \\
&P_{\text{b}_{\min}} \leq P_{\text{b}}^{l}(i) \leq P_{\text{b}_{\max}}
\end{aligned} \tag{4.16}$$

where the variables to be optimized are the coordinates of $n-1$ waypoints $(x(i), y(i)), i = 2, ..., n-1$, and the $n1$ linear speeds $V(i)$ and angular speeds $\omega(i)$ along each segment $i, i = 1, ..., n-1$.

4.2 Constrained RRT Algorithm Formulation

This section intends to approach the mission planning problem from a stochastic sampling and exploring aspect. The Rapidly-exploring Random Tree (RRT) algorithm from Lavalle has been modified to serve a better purpose for mission scheduling and path planning of the solar-powered UGV with constrained net energy and traveling time [Lavalle (1998)]. Before discussing the algorithm, it is essential to recognize the mission planning problem and introduce a few necessary procedures implanted in the algorithm.

The environment with solar energy density distribution is denoted as R_{in} where $R_{\text{in}} \subset \mathbb{R}^{\text{n}}$. Let S represents the unexplored domain and $S \subset R_{\text{in}}$. The environment is assumed to be obstacle-free and the rover starts at a given initial position, z_{initial}. The RRT algorithm would solve the constrained mission planning problem by searching through the region S with node sampling in order to construct a Rapidly-exploring Random Tree, $Tree$. Furthermore, each branch from $Tree$ would be examined respectively to yield a feasible path, $Path$, which satisfies all the given requirements.

Sample: The `Sample` function returns a state, z_{new}, that randomly samples from the desired field. The sampling process is assumed to be uniformly distributed while selecting nodes in the field, such that $z_{\text{new}} \in S$ (Alg. 1, line 4). The coverage of every new sample is also restricted by the radius of on-board sensor, r, to not exceed the boundaries of the environment while sampling.

Near: The `Near` function evaluates the new sample and the current tree, then determines a node, z_{near}, where $z_{\text{near}} \in Tree$ that offers the shortest connecting distance of being the potential parent for z_{new}.

EnergyOptimal: Given the solar energy distribution of the environment and the necessary parameters for motion of the rover, the `EnergyOptimal` function determines a favorable PWM associating with speed and duration that maximizes the net energy towards $+\infty$ for either turning or going straight actions.

Coverage: This procedure adopts the calculation of area coverage discussed in section IV with the input of waypoints for the path.

Algorithm 1 outlines the general structure of the RRT algorithm used in this paper. The

tree begins with z_{initial} and a blank path. A coordinate z_{new} is generated by stochastic sampling the design space (Alg. 1, line 4). It then determines its potential parent z_{near} (Alg. 1, line 5) and recruits the available z_{new} for tree extension (Alg. 1, line 6). A set of angular and linear maneuvers would need to be characterized once a new node is added. The turning angle, α, for the first rotational motion at z_{near} is computed by `TurningAngle` function where $\alpha \in [0, \frac{\pi}{2}]$ due to the rover is capable to perform revise driving (Alg. 1, line 7). The `Distance` function calculates the path length, β, from z_{near} to z_{new} for the following straight line movement (Alg. 1, line 8). The estimation of intake solar energy and battery consumption during operations are measured by `EnergyOptimal` function from E to evaluate the required velocity, length of time, and outcome net energy when the rover reaches z_{new} (Alg. 1, line 9-10). While every time a z_{new} is added, a new branch, I, is expanded from z_{initial} to z_{new} and assessed with traveling time and total energy requirements (Alg. 1, line 15-16). After sufficient iterations, the branches that passed the thresholds competing with each other and producing one best branch as *Path* which provides the maximum area coverage of all branches (Alg. 1, line 17-20).

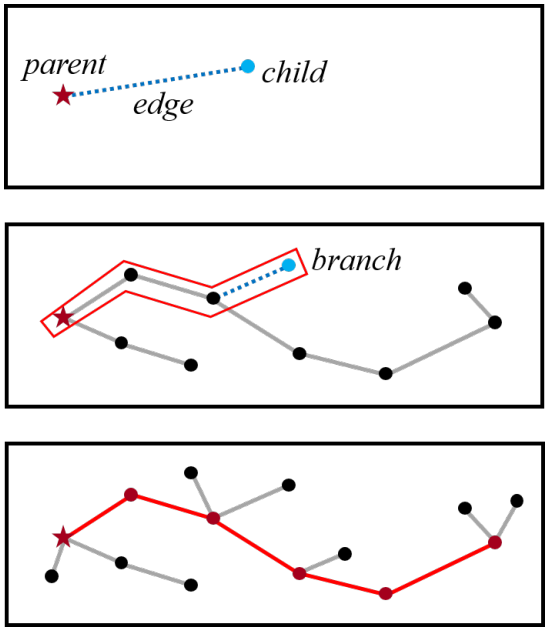


Figure 4.4 Example of constrained RRT steps.

Algorithm 1: The Constrained RRT Algorithm

INPUT: $z_{\text{initial}} \leftarrow (x_0, y_0), r \leftarrow$ Detection Radius, $PWM_{\text{turn}} \leftarrow [turn_{\min}, turn_{\max}]$, $PWM_{\text{line}} \leftarrow [line_{\min}, line_{\max}]$, $P_{\text{passive}} \leftarrow$ Power Passive, $S_{\text{panel}} \leftarrow$ Solar Panel Size, $R_{\text{in}} \leftarrow$ Solar Energy Insolation Map, $K \leftarrow$ Limitation of Traveling Time

Initialize: $Tree \leftarrow \{z_{\text{initial}}\}, turn \leftarrow \{\}, line \leftarrow \{\}, time \leftarrow \{\}, energy \leftarrow \{\}, I \leftarrow \{\}$, $A_{\text{Total}} = 0, Path \leftarrow \{\}$

for $i = 1, ..., n$ **do**

- $z_{\text{new}} \leftarrow \texttt{Sample}(S, r)$
- $z_{\text{near}} \leftarrow \texttt{Near}(z_{\text{new}}, Tree)$
- $Tree \leftarrow Tree \cup \{z_{\text{new}}\}$
- $\alpha \leftarrow \texttt{TurningAngle}(z_{\text{new}}, z_{\text{near}}, Tree)$
- $\beta \leftarrow \texttt{Distance}(z_{\text{new}}, z_{\text{near}})$
- $(turn_{\text{PWM}}, turn_{\text{velocity}}, turn_{\text{time}}, turn_{\text{energy}}) \leftarrow \texttt{EnergyOptimal}(\alpha, P_{\text{passive}}, PWM_{\text{turn}}, R_{\text{in}})$
- $(line_{\text{PWM}}, line_{\text{velocity}}, line_{\text{time}}, line_{\text{energy}}) \leftarrow \texttt{EnergyOptimal}(\beta, P_{\text{passive}}, PWM_{\text{line}}, R_{\text{in}})$
- $turn \leftarrow turn(z_{\text{near}}) \cup \{(turn_{\text{PWM}}, turn_{\text{velocity}})\}$
- $line \leftarrow line(z_{\text{near}}) \cup \{(line_{\text{PWM}}, line_{\text{velocity}})\}$
- $time \leftarrow time(z_{\text{near}}) + turn_{\text{time}} + line_{\text{time}}$
- $energy \leftarrow energy(z_{\text{near}}) + turn_{\text{energy}} + line_{\text{energy}}$
- $I \leftarrow I(z_{\text{near}}) \cup \{z_{\text{new}}\}$
- **if** $time \leq K$ and $energy \geq 0$ **then**
  - $A'_{\text{Total}} \leftarrow \texttt{Coverage}(I)$
  - **if** $A'_{\text{Total}} > A_{\text{Total}}$ **then**
    - $Path \leftarrow I$
    - $A_{\text{Total}} \leftarrow A'_{\text{Total}}$

return $Path$

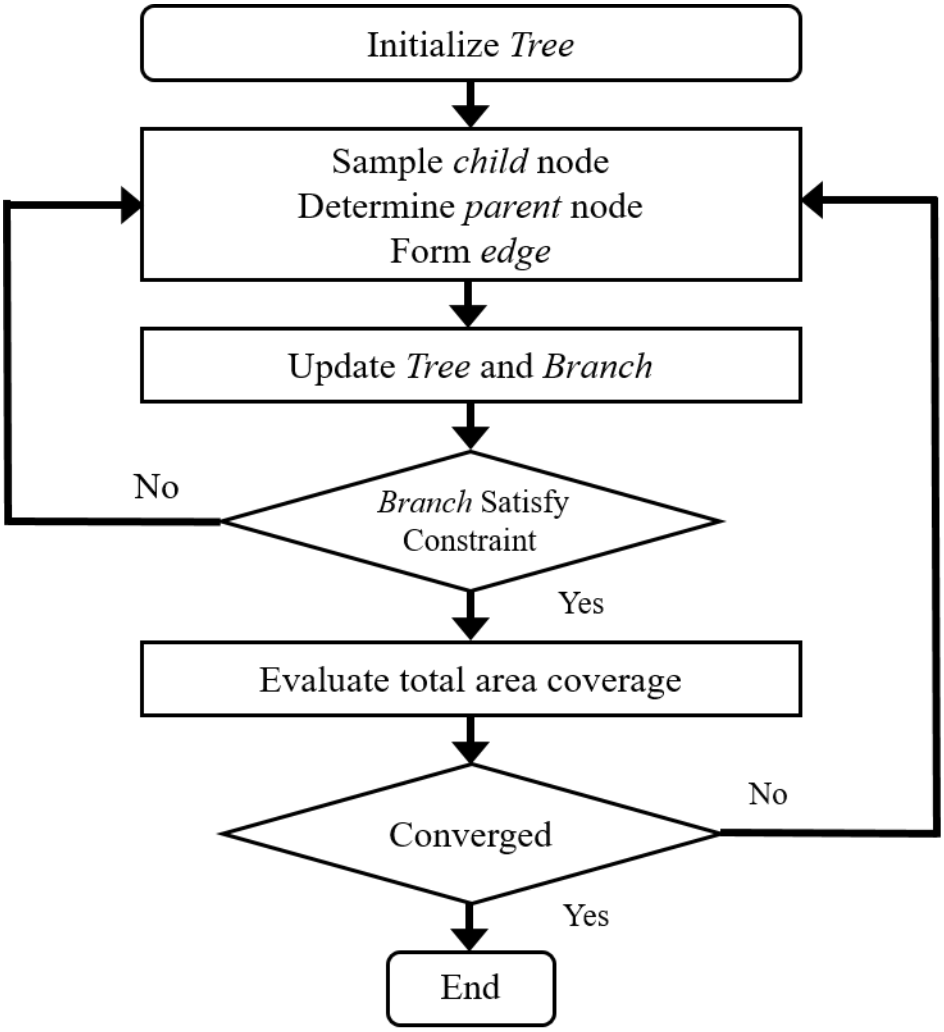


Figure 4.5 Flowchart of constrained RRT steps.

CHAPTER 5. PATH PLANNING OF A MULTI-ROBOT TEAM WITH A SOLAR-POWERED CHARGING STATION

This chapter presents a mission planning problem for a cooperative team of unmanned ground vehicles (UGVs), which includes multiple rovers and a solar-powered mobile charging station. The team is required to start at an initial point and visit a series of objective points before arriving at the final point selected from the set of objective points, where the UGVs will be charged from the solar-powered mobile charging station. This mission is represented as a multi-Hamiltonian Path Problem (mHPP). In order to effectively coordinate the team, an understanding of the mission environment is first obtained by generating a scalar field representation of the solar insolation of the environment from a visual-spectrum image. Then, a cascaded heuristic optimization (CHO) algorithm, using modified genetic algorithm and particle swarm optimization, is used to generate a time-optimized mission plan for the team of UGVs, which guides each UGV to its assigned objective points and then rendezvous at the final charging location while guaranteeing compliance with the net energy gain constraint.

5.1 Problem Formulation

Consider a cooperative team of m UGVs, of known performance characteristics, where one of the UGVs is equipped with a solar-panel and has the capability of harvesting solar energy and charging the other UGVs. This vehicle is referred as a solar-powered, mobile charging station with index $z = 1$. All other vehicles, indexed by $z = 2, ..., m$, are classified simply as worker UGVs. The mission assigns a number of objective points where each objective point can only be visited by one UGV, except the beginning and final points. The team is also required to select one of those objective points and rendezvous at this final destination to charge all worker UGVs. The entire mission is required to be completed with minimum time. In order

to enable persistent operation in the environment, each vehicle's net energy gain is required to be greater than or equal to zero once the mission is complete.

With the involvement of a solar-powered mobile charging station, the UGV team has more flexibility of accessing external power supply without distance constraint. On the other hand, the worker UGVs in smaller size has more mobility when traveling to limited spaces without the top-mounted large solar panel. Furthermore, the charging station can also act as a worker UGV to visit objective points during energy harvesting. Thus, the cooperation of heterogeneous UGV team is capable of performing more challenging tasks with less dependence on power supply.

The optimal mission planning problem for the above described UGV team is similar to the mHPP with additional dynamical and mission constraints. The mHPP can be described by a complete graph G which consists of a set of objective points, $O = O_{\{1,...,n\}}$, where $n \geq m + 2$, and a set of edges, E, with edge cost representing the time required for a UGV traversing the corresponding edge. Let D be the depot point 1 such that $O = D \cup O'$ and O_k be the final location k of all m UGVs, and $c_{ij,z}$ be the binary variable representing if a UGV with index z joins the two points i and j. Then, the mHPP can be formulated as

$$\min G(O, E) \tag{5.1}$$

$$\text{s.t.} \quad \textstyle\sum_{i \in O'} c_{1i,z} = m \tag{5.2}$$

$$\textstyle\sum_{i \in O'} c_{ik,z} = m, k \in O' \tag{5.3}$$

$$\textstyle\sum_{i \in O} c_{ij,z} = 1, j \in O', i \neq j, j \neq k, \forall z = 1, ..., m \tag{5.4}$$

$$\textstyle\sum_{j \in O} c_{ij,z} = 1, i \in O', i \neq j, i \neq k, \forall z = 1, ..., m \tag{5.5}$$

$$c_{1i,z} + c_{ik,z} \leq 1, \forall i \in O', \forall z = 1, ..., m \tag{5.6}$$

$$c_{ij,z} \in \{0, 1\}, \forall i, j \in O, \forall z = 1, ..., m. \tag{5.7}$$

Constraints (5.2) and (5.3) guarantee that all m vehicles start at point 1 and end at point k. Constraints (5.4) and (5.5) impose that one objective point can only be visited once, except for the depot and final locations. Constraint (5.6) ensures that each UGV must visit at least one objective point which is not the depot and final location. The binary constraint (5.7) $c_{ij,z} = 1$ represents the edge is visited by UGV z, and $c_{ij,z} = 0$ otherwise. An example of this problem for three UGVs and nine objective points is illustrated in Fig. 5.1.

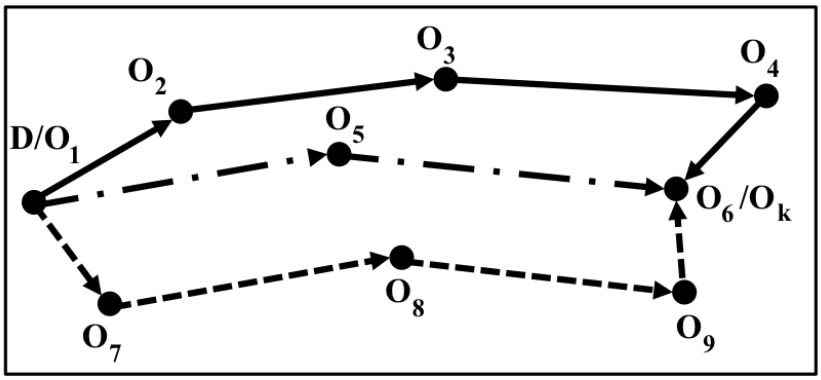


Figure 5.1 An example for a mHPP with three UGVs and nine objective points where path of UGV 1 is solid, UGV2 is dash-dot, and UGV 3 is dash.

The above mHPP does not consider vehicle dynamics and power constraints. In the multi-robot team, each UGV's movements are governed by the four differential-drive primitives of a Balkcom-Mason curve [Balkcom and Mason (2002)], as shown in Fig. 5.2. In this way, the dynamical model for each UGV can be described below. The time spent by UGV z to visit the n_z assigned waypoints is denoted as $T_{travel,z}$, which includes the time consumed by the $n_z + 1$ turns, denoted by T_z^r, and n_z line segments, denoted by T_z^l,

$$T_{travel,z} = \sum_{i=1}^{n_z+1} T_z^r(i) + \sum_{i=1}^{n_z} T_z^l(i), z = 1, ..., m. \tag{5.8}$$

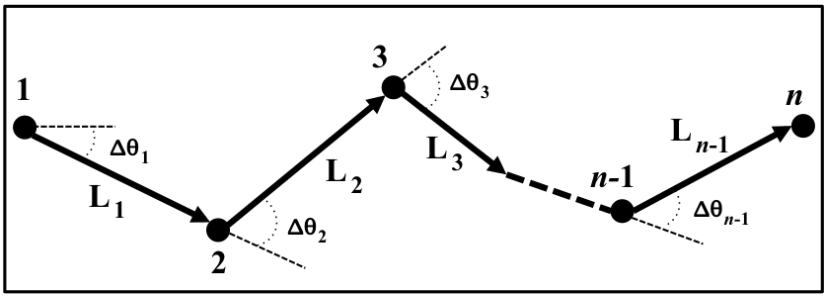


Figure 5.2 An example of Balkcom-Mason curve path.

Along each turn, $i = 1, ..., n_z + 1$, and line segment, $i = 1, ..., n_z$, the UGV z will move at an angular and linear speed, denoted by $\omega_z(i)$ and $V_z(i)$, respectively. These speeds will be constant for worker UGVs. However, the mobile charging station has more flexibility to determine its speeds from the motion planning strategy. Thus, $T_{travel,z}$, can be expressed as

$$T_{travel,z} = \sum_{i=1}^{n_z+1} (\frac{\Delta\theta_z(i)}{\omega_z(i)}) + \sum_{i=1}^{n_z} (\frac{L_z(i)}{V_z(i)}),\ z = 1, ..., m, \tag{5.9}$$

where $\Delta\theta_z(i)$ is the required change in angle from the UGV's present heading to the new heading before moving straight to the next waypoint, and $L_z(i)$ is the length of line-segment i.

For each movement, the UGVs must allocate sufficient power to their engines to maintain the consumption rates for the duration of each line-segment and turn, denoted by $P^l_{e,z}(i)$ and $P^r_{e,z}(i)$, respectively, in addition to a constant power drain from the vehicles' microcontroller and other components, collectively denoted as the passive power draw, P_a. Each movement for UGV, $z = 1, ..., m$, is summarized as

$$E^l_{out,z}(i) = (P^l_{e,z}(i) n_{e,z} + P_{a,z}) T^l_z(i) \tag{5.10}$$

$$E^r_{out,z}(i) = (P^r_{e,z}(i) n_{e,z} + P_{a,z}) T^r_z(i) \tag{5.11}$$

where $n_{e,z}$ is the number of engines for the UGV with index z. The energy recovered throughout the mission is unique for the workers and charging station. For the worker UGVs, it can be expressed as,

$$E_{in,z} = C_{rate} T_{charge,z},\ z = 2, ..., m, \tag{5.12}$$

where C_{rate} is the charging rate, which is hardware dependent, and T_{charge} is the required amount of time to charge the battery. The energy recovered by the charging station with index $z = 1$ is determined by the solar panel's surface area, S_a, and the available solar energy R_{in}. For a line segment, the available energy is calculated by sampling over the line segment at Q equidistantly spaced points to determine the average energy density along the line, $R_{in}(x_q, y_q)$, where $q = 1, \ldots, Q$. While for a turn, the available energy density is simply approximated as the location of the turn, $R_{in}(x(i), y(i))$, $i = 1, \ldots, n_1 + 1$. Under these assumptions, the energy harvesting amount for the mobile charging station along the turn and line-segment is expressed as

$$E^l_{in,1}(i) = \sum_{q=1}^{Q} R_{in}(x_q, y_q) S_a L_1(i)/(V_1(i)Q) \tag{5.13}$$

$$E^r_{in,1}(i) = R_{in}(x(i), y(i)) S_a \Delta\theta_1(i)/\omega_1(i). \tag{5.14}$$

In order to achieve persistent operation, the strict net energy constraint imposed over the entire mission is expressed as

$$\begin{aligned}
\Delta E_{total,1} = &\sum_{i=1}^{n_1+1} (E^r_{in,1}(i) - E^r_{out,1}(i)) \\
&+ \sum_{i=1}^{n_1} (E^l_{in,1}(i) - E^l_{out,1}(i)) \geq 0. \\
\Delta E_{total,z} = &- \sum_{i=1}^{n_z+1} (E^r_{out,z}(i)) \\
&- \sum_{i=1}^{n_z} (E^l_{out,z}(i)) + E_{in,z} \geq 0, \, \forall \, z = 2, ..., m.
\end{aligned}$$

At any instance during the operating period, each UGV must be capable of maintaining operation by obtaining power from its on-board battery, $P_{b,z}$ or external sources. The worker UGVs, $z = 2, ..., m$, can only obtain energy from the charging station with the power balance relationship expressed as

$$P^l_{e,z}(i) n_{e,z} + P_{a,z} + P^l_{b,z}(i) = 0, \; i = 1, \ldots, n_z, \tag{5.15}$$

$$P^r_{e,z}(i) n_{e,z} + P_{a,z} + P^r_{b,z}(i) = 0, \; i = 1, \ldots, n_z + 1. \tag{5.16}$$

For the solar-powered mobile charging station, power is first drawn from the solar panel. If the power provided from the solar panel is less than the power required to maintain the UGV's operation, the on-board battery will supplement the rest of the power. On the other hand, if there is excess power provided by the solar panel, the remaining will be stored in the on-board battery. This relationship for both linear and rotational segments of the charging station with $z = 1$ is formulated as

$$P^l_{e,1}(i)n_e + P_{a,1} + P^l_{b,1}(i) = \sum_{q=1}^{Q} R_{in}(x_q, y_q)S_a/Q,$$
$$i = 1, \ldots, n_1 \tag{5.17}$$
$$P^r_{e,1}(i)n_e + P_{a,1} + P^r_{b,1}(i) = R_{in}(x(i), y(i))S_a,$$
$$i = 1, \ldots, n_1 + 1. \tag{5.18}$$

Lastly, the battery of each UGV must stay within its minimum and maximum values, denoted by $P_{b_{min},z}$ and $P_{b_{max},z}$, respectively. Hence, for any $z = 1, \ldots m$, it satisfies the following constraints such that

$$P_{b_{min},z} \leq P^l_{b,z}(i) \leq P_{b_{max},z},\ i = 1, ..., n_z$$
$$P_{b_{min},z} \leq P^r_{b,z}(i) \leq P_{b_{max},z},\ i = 1, ..., n_z + 1.$$

Accordingly, the multi-robot mission planning problem can be summarized as a parameter optimization problem in form of

$$\min_{\mathbf{X}} \sum_{z=1}^{m} \Big(\sum_{i=1}^{n_z+1} T^r(i) + \sum_{i=1}^{n_z} T^l(i) \Big) + T_{charge} \tag{5.19}$$
$$\text{s.t} \quad c_{ij,z} \in \text{constraints in } (5.2) - (5.7)$$
$$\Delta E_{total,z} \geq 0, z = 1, \ldots, m$$
$$P^l_{e,1}(i)n_{e,1} + P_{a,1} + P^l_{b,1}(i) - \sum_{q=1}^{Q} R_{in}(a_q, b_q)A_s/Q = 0$$
$$P^r_{e,1}(i)n_{e,1} + P_{a,1} + P^r_{b,1}(i) - R_{in}(x(i), y(i))S_a = 0$$
$$P^l_{e,z}(i)n_{e,z} + P_{a,z} + P^l_{b,z}(i) = 0, z = 2, \ldots, m$$
$$P^r_{e,z}(i)n_{e,z} + P_{a,z} + P^r_{b,z}(i) = 0, z = 2, \ldots, m$$
$$P_{b,z_{min}} \leq P^l_{b,z}(i) \leq P_{b,z_{max}}, z = 1, \ldots, m$$

where the optimization variable set $\mathbf{X}$ include the binary set $\{c_{ij,z}\}$ which determines objective points assigned to each UGV, the visiting sequence of assigned objective points, and the final rendezvous location for charging. The remaining design variables in $\mathbf{X}$ are the corresponding linear and rotational velocities $V_{i,z}$, $i = 1, \ldots, n_z$ and $\omega_{i,z}$, $i = 1, \ldots, n_z + 1$ for each UGV $z = 1, ..., m$. The mission planning problem in (5.19) including both continuous and binary design

variables and nonlinear constraints are complicated, requiring a specially designed algorithm to search for an optimized solution.

5.2 Cascaded Heuristic Optimization Method

5.2.1 Genetic Algorithm

GA, developed in the 1970's, is an adaptive heuristic algorithm based in the larger class of evolutionary algorithms [Holland (1975)]. GAs employ a population of N individuals, named chromosomes, which are encoded to represent a possible solution to a problem usually in binary or decimal form. Extending the genetic analogy, each chromosome is composed of i genes, or variables. For each generation, or iteration, of the GA, a new set of N individuals are generated from the previous generation. This process is completed in three steps that simulates the evolutionary and survival of the fittest process: (1) selection, (2) cross-over, and (3) mutation. As generations continue, GAs intelligently explore a design space using a random search, exploiting historical information to direct the search towards regions of space with higher performance. Due to the great number of binary design variables involved in the mission planning problem, GA is elected to determine the binary set $\{c_{ij,z}\}$ first by encoding them as a single chromosome.

5.2.2 Particle Swarm Optimization

PSO is a population based stochastic heuristic optimization method which was developed in 1995 [Eberhart and Kennedy (1995)]. PSO simulates the social behavior of animals, such as birds or fish, by creating N groups of potential solutions, or particles, each being composed as a particular solution set to the optimization problem in form of

$$P^n = [x_1^n, x_2^n, \ldots, x_d^n], \forall\, n = 1, \ldots, N. \tag{5.20}$$

At each iteration, the N particles are perturbed by a distance, D, that is determined based on the particle's best known solution, P_{BEST} and the population's best performing solution, G_{BEST}. This movement, D, can be adjusted by the user-defined weights, C_1 and C_2, that determine the importance of moving towards P_{BEST} and G_{BEST}. In order to effectively search the design space, the movement D are also multiplied by a pseudorandom number R between

0 and 1. PSO used in this paper also takes into account the previous iteration's movement $D_i^j(t-1)$ multiplied by an inertia factor I, $0 \leq I \leq 1$, which is a common method to reduce the chances of being trapped in a local minimum, such that

$$\begin{aligned} D_i^n &= ID_i^n(t-1) + (C_1R(P_{BEST_i}^n - P_i^n)) \\ &\quad +(C_2R(G_{BEST_i} - P_i^n)). \end{aligned} \tag{5.21}$$

After each perturbation, the $n = 1, \ldots, N$ particles' fitness is re-evaluated against the problem's performance index. If appropriate, the particle's new solution replaces the previous P_{BEST} and/or G_{BEST} values. As iterations continue, the G_{BEST} will approach an optimal solution set. Due to the high computational performance of handling nonlinear constraints, PSO is elected to search for continuous design variables in the energy-constrained mission planning problem.

5.2.3 Cascaded Heuristic Optimization Method

The mission planning problem requires that all objective points, $O_j(x_j, y_j)$, $j = 1, ..., k$, are visited by only one UGV, except for the starting location and the final charging location. Accordingly, a CHO algorithm is proposed to solve the energy-constrained mHPP formulated in §II. The CHO will first allocate the objective points among UGVs, the visiting sequence of assigned point for each UGV, and the final charging location, through the use of a GA. Then, we will design linear and angular velocities, $V_{i,z}$, $i = 1, ..., n_z$ and $\omega_{i,z}$, $i = 1, ..., n_z + 1$ for $z = 1, ..., m$ along the path segments with a modified PSO.

Our first step is to determine the binary set $\{c_{ij,z}\}$, which represents objective points that each UGV will visit, the visiting sequence, and the final charging location. Due to the involvement of a great number of binary variables, GA is elected to solve for the optimal path orders for each vehicle simultaneously, while still being able to incorporate the non-linearity of the solar insolation map for the purpose of energy harvesting. For a mHPP with homogeneous UGVs, the objective of the GA is to minimize the time to complete the mission. Then, the motion planning problem without a charging station would simply be searching the overall minimum distance the UGV team must travel. However, when involving a solar-powered UGV to harvest as much as energy as possible while working with other UGVs, the objective becomes a hybrid

function composed of distance and collected energy per distance. We elect to apply a weighted objective function to incorporate the performance index of the solar-powered charging station, expressed as

$$\min \sum_{z=2}^{m} d_{total,z} - w \sum_{i=1}^{n-1} \frac{E^l_{in,1}(i) + E^r_{in,1}(i)}{d_{total,1}}, \tag{5.22}$$

where $E^l_{in,1}(i)$ and $E^r_{in,1}(i)$ are defined in (5.13) and (5.14), respectively, and w is a user specified weighting factor.

To solve this problem using a GA, each portion of the problem must be represented in every chromosome. To accurately and efficiently represent each portion, a three-part chromosome is utilized containing the charging location, the objective points, and how many and which objective points are assigned to each vehicle, as shown in Fig. 5.3. The three-part chromosome used in this paper is an extension of the work completed by Carter [Carter and Ragsdale (2006)] that verifies a two-part chromosome outperformed the two chromosome representation for a MmTSP. By representing the problem in this way, all information required to obtain a fitness value for the particular chromosome is contained in a single array.

The GA developed in this paper uses the tournament selection approach, which selects a subset of the population and then selects the fittest member of the population. Instead of using a single point crossover for the objective point crossover, we implemented an ordered crossover [Goldberg (1989)], which is one of the fastest crossover operators, requiring limited overhead operations. The working mechanism of ordered crossover is to select n consecutive

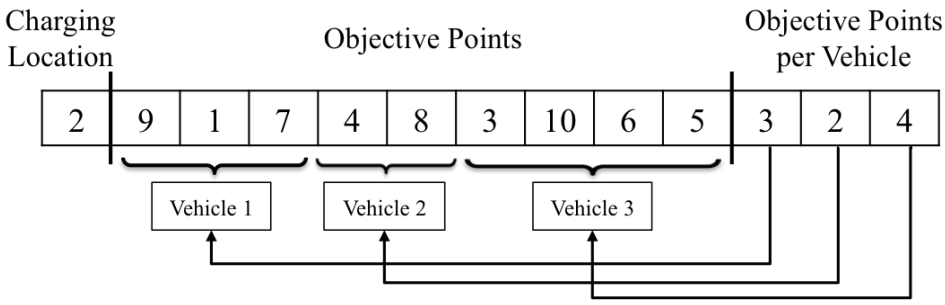


Figure 5.3 Example of the three-part chromosome representation for a 10-point mHPP with a three-UGV team.

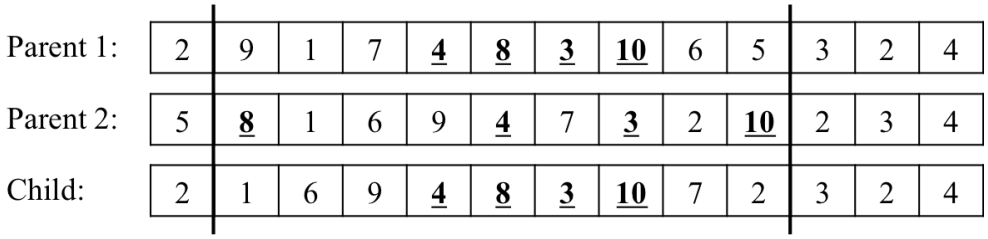


Figure 5.4 Example of Order 1 cross over with the three-part chromosome.

genes within one of the parents, which are represented by bolded and underlined numbers in Fig. 5.4, to be dropped down to the child. Then, the remaining values are placed in the child in the order that they appear in the other parent. For the charging location and the number of objective points assigned to the vehicle, these values from the parent that has the highest fitness are selected to be passed on to the child. However, a weighting is applied so that there is a chance that the values from the other parent are used instead. This offers some variability and random chances to help from being trapped in a local minimum. Lastly, for a small subset of the population, mutation is completed by randomly swapping two objective points, which also prevents becoming trapped in local minimums.

Next, a modified PSO is used to determine the linear and rotational speed along each segment of the optimized path. A set of k candidate solutions are pseudorandomly initialized as

$$P^k = [V^k(1), \ldots, V^k(n_z), \omega^k(1), \ldots, \omega^k(n_z + 1)]. \tag{5.23}$$

Throughout the optimization process the particles are constrained by the physical limitations of the vehicles, $V_{min} < V^k < V_{max}$ and $\omega_{min} < \omega^k < \omega_{max}$, as well as the constraints stated in (5.19). The objective function for this optimization process is the one specified in (5.19). As the particles are perturbed through each iteration, the particles will converge to an optimal solution that meets all constraints applied to the system. In the event that the initialization does not provide a particle that satisfies the net energy constraints, the particles will be reinitialized. A summary of the proposed CHO method is shown in Fig. 5.5.

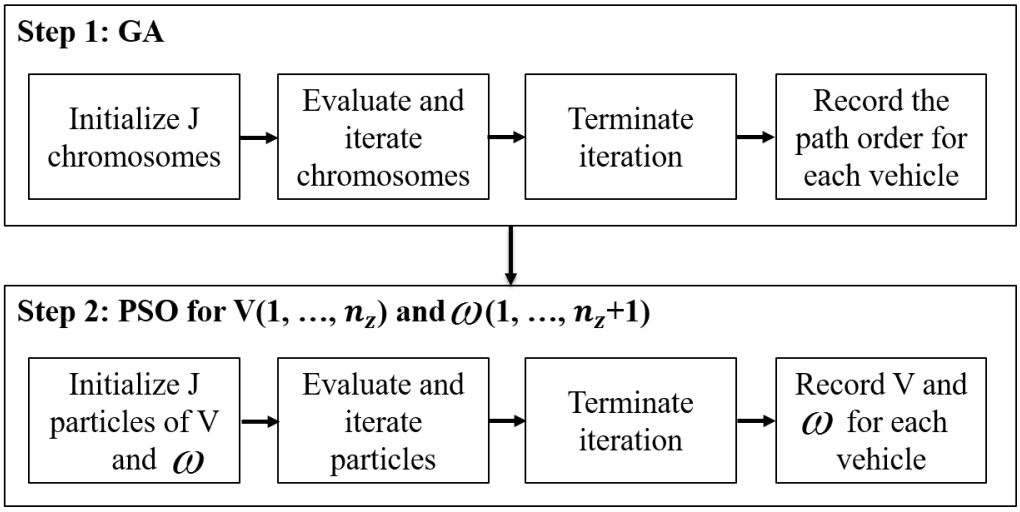


Figure 5.5 Flowchart of the proposed CHO method

CHAPTER 6. SIMULATION AND EXPERIMENT RESULTS

This chapter contains the virtual simulations and experimental results for constrained path planning of unmanned vehicles problems discussed in previous chapters. Section 6.1 presents simulated results of refined RRT* in Chapter 2 for single and multiple avoidance zones cases and a benchmark comparison to address the performance of heuristics method. The simulations for multi-waypoint and decision parameter methodology posted in Chapter 3 are shown in Section 6.2. In Section 6.3 and 6.4, both computer simulations and experimental data are included for information gathering and multi-rover with charging station problems from Chapter 4 and 5, respectively.

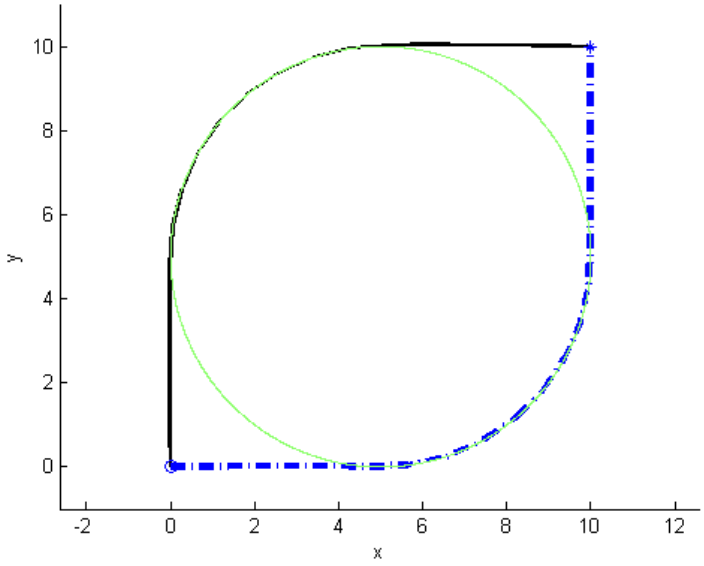
6.1 Path Planning Of Unmanned Aerial Vehicles with Avoidance Zones

In this section, a group of simulation cases are presented to demonstrate the advantages and limitations of the proposed approach and compared with an Iterative Rank Minimization (IRM) approach [Sun and Dai (2015)]. In the first case, one avoidance zone is considered and the path planning problem for this simple case has an analytical solution as a reference to verify the accuracy of proposed approaches. Other cases consider a group of cluttered avoidance zones, which makes the path planning problem more difficult. In both types of cases, the parameters in the IRM method are set as $[x_0, y_0] = [0, 0]^T, [x_f, y_f] = [10, 10]^T, V = 1, u_{max} = 0.2, w = 1.5$, and $\epsilon = 1e - 4$ as the threshold for the convergence of r. All cases use 25 discrete nodes to represent the planned path. The corresponding parameters in the refined RRT* method are set as $d_{min} = 0.3$ and $\theta = 150°$. All of the simulation cases are run on a desktop computer with a 3.50 GHz processor and of 16 GB RAM.

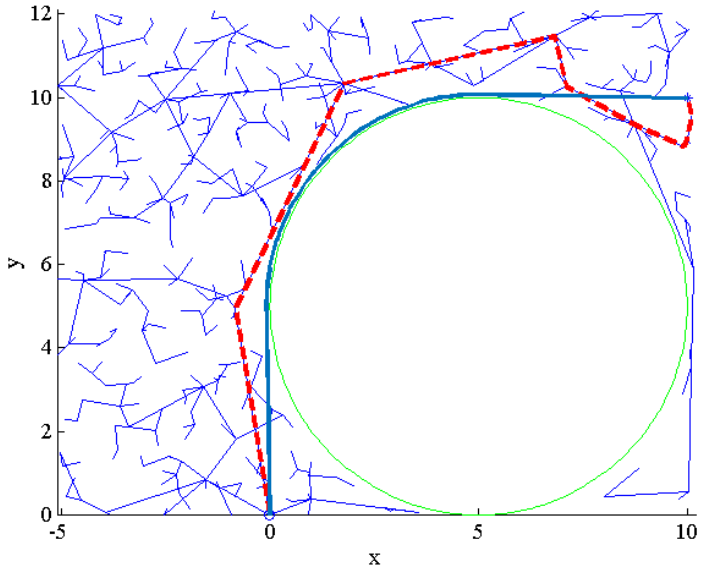
6.1.1 Case with One Avoidance Zone

The single avoidance zone considered here is a circle centered at $[5, 5]$ with radius of 5. The analytical solution for this case yields a minimum flight time of $f^*_{benchmark} = 17.85$. The optimal solution from IRM is $f^*_{IRM} = 17.84$. The relative error of the solution from IRM is only 0.066% compared to the benchmark, which verifies the high accuracy of IRM. In addition, results obtained from the NLP solver in MATLAB is provided. Groups of initial guesses are randomly generated and a feasible path which yields a flight time of $f^*_{NLP} = 17.84$ is demonstrated in Figure 6.1(a). Therefore, the comparative results indicate that IRM and NLP both work in solving this type of benchmark problems when converging to a local minimizer. The paths obtained from IRM and NLP are demonstrated in Figure 6.1(a). It takes 10 iterations for the IRM with threshold $\epsilon = 1e - 4$ converging to the optimal solution. The overall computation time for IRM is 202.90 seconds.

The planned paths from RRT* and refined RRT* are shown in Figure 6.1(b). The original path from RRT* includes sharp turns at connections of sampling trees. Through refinement steps, the refined path with flight time of $f^*_{refined} = 18.00$ is very close to the planned path generated from IRM. It takes 0.95 seconds to generate the coarse path from RRT*. The refinement steps take additional 25.15 seconds to refine the coarse path. Although the refined RRT* does not yield the same high accuracy compared to IRM, it has significant reduction on computation time in this simple example.



(a) Planned paths generated from IRM (dash) and NLP (solid) for case one



(b) Planned paths generated from modified RRT* (dash) and refined RRT* (solid) for case one

Figure 6.1 Comparative results of planned path for case one.

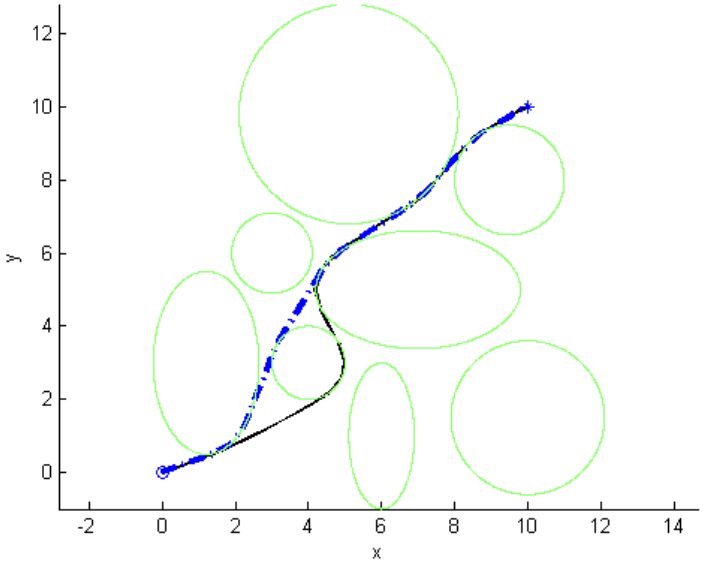
Moreover, as the dynamics equations are discretized, we present the results from different N in Table 6.1. In this example, as we have $N = 20$ for it is already accurate enough.

Table 6.1 Comparative results for different N.

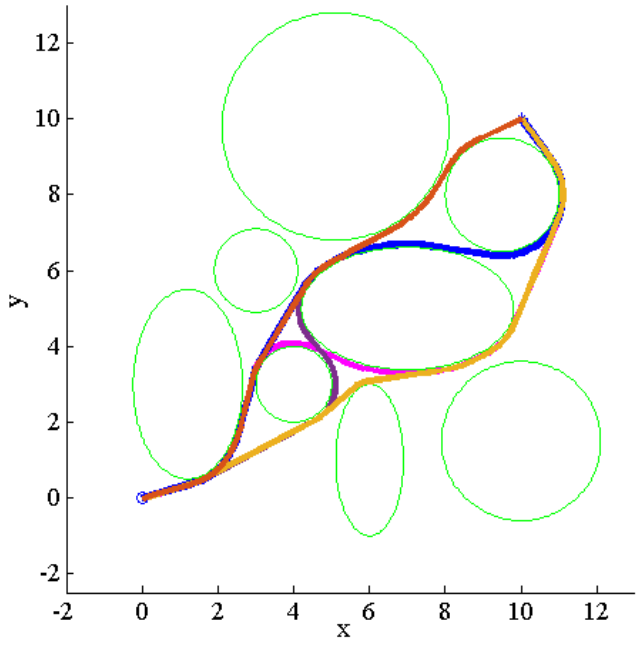
N	10	15	20	25
$f_{\text{benchmark}}$	17.85	17.85	17.85	17.85
f_{IRM}	17.80	17.86	17.84	17.85
IRM Rel. Error/%	-0.27	0.063	-0.066	0.0029

6.1.2 Cases with Multiple Avoidance Zones

A group of cluttered avoidance zones in the shape of circles and ellipses are considered in the second case. The optimal solution from IRM is $f^*_{IRM} = 14.75$, while the one from NLP is $f^*_{NLP} = 16.02$. Figure 6.2(a) shows the planned paths from IRM and NLP for this case. The IRM converges to the optimal solution within 15 iterations and the overall computational time is 1065.77 seconds. Due to the multiple avoidance zones, it is more difficult to find a feasible path when using a random initial guess with the NLP method. Different from case one where the paths from refined RRT* lead to a semi-uniform result with significantly small differences, paths generated for case two fall into several branches, as shown in Figure 6.2(b). Among 350 simulation runs, only 5.43% simulations lead to a shortest path with $f^*_{refined} = 14.93$, which is consistent with the one obtained from IRM. However, most of the simulations, with the probability of 79.71%, generate flight paths yielding $f^*_{refined}$ around 17.96 with ± 0.001 difference. The remaining 14.85% simulations generate results with even larger $f^*_{refined}$. Although the random process is assumed to sample the distribution uniformly, it is observed that the RRT is biased to generate more samples in larger spaces Choset (2007). Thus the sampling trees are not likely to distribute vertices in narrow spaces, leading to the low opportunity of producing the shortest outcome in case two. It takes an average of 10.88 seconds to find the coarse path from RRT* and the average computation time to obtain the final path considering refinement steps is 193.29 seconds. Compared with the simulation time of IRM, the refined RRT* requires much less computation time without guarantee of optimality. For the convenience for comparison, we listed the result from the three methodologies in Table 6.2.



(a) Planned paths generated from IRM (dash) and NLP (solid) for case two



(b) Planned paths generated from refined RRT* for case two

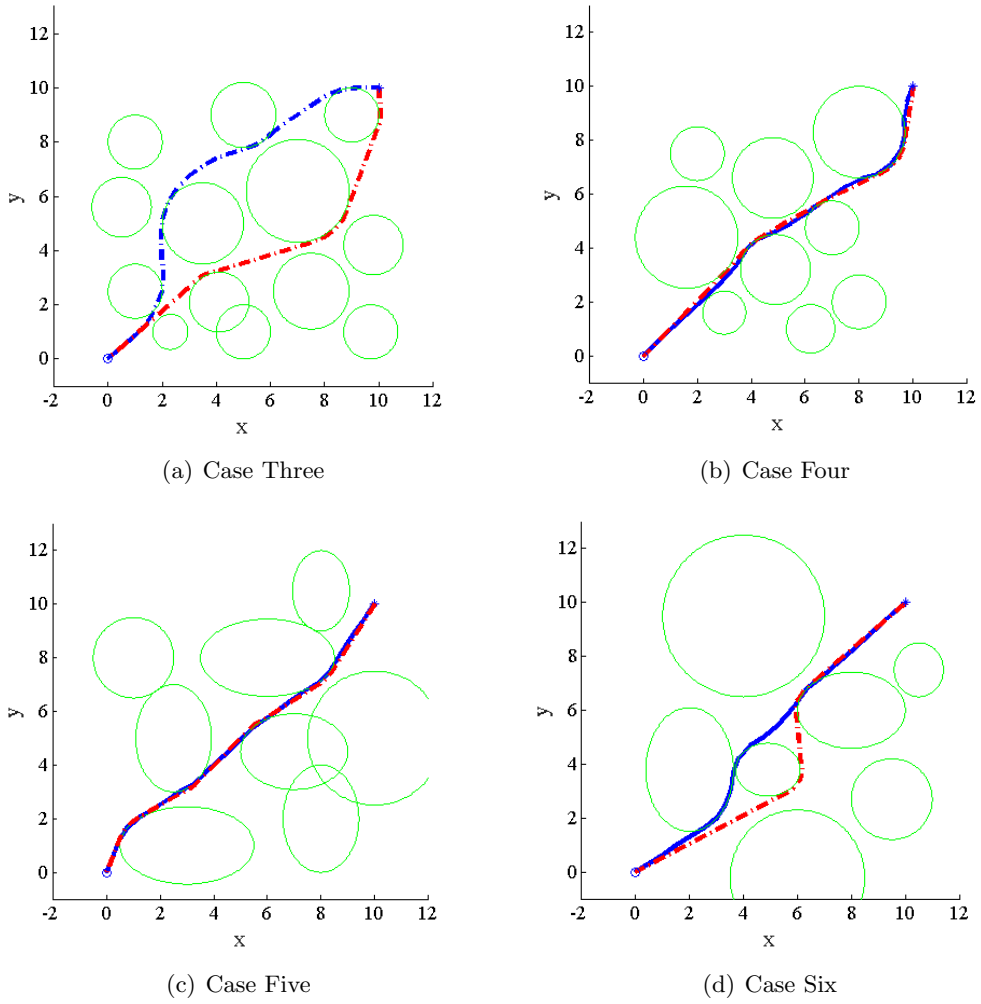
Figure 6.2 Comparative results of planned path for case two.

Table 6.2 Comparison for three algorithms.

Algorithm	Parameter	time/sec	f^*	Remark
NLP	SNOPT	20.76	16.02	Least time
IRM	$N = 25$,$w = 1.5$,$\epsilon = 1e-4$	1065.77	14.75	(Equally best f^*)
Modified RRT*	$d_{min} = 0.3$	10.88	-	
Refined RRT*	$\theta = 150^\circ$	193.29	14.93	

To further verify advantages of the proposed methods, results of four additional cases are demonstrated in Figure 6.3, where overlap between avoidance zones are considered. For each case, the refined RRT* execute 15 simulation runs and the path yielding best performance is provided in the corresponding plot to compare with those obtained from IRM. Among the four cases, two of them generate very similar results and the remaining two lead to different paths where flying time resorting from the IRM is slighter shorter than the corresponding results from the best result of the refined RRT*.

All example cases indicate that the numerical optimization approach using the IRM method achieves high performance with guaranteed convergence. However, at each iteration of IRM, the additional semidefinite constraint, $r_k I_{n-1} - V_{k-1} X_k V_{k-1} \succeq 0$ in Sun and Dai (2015), introduces $(n-1) \times (n-1)$ linear constraints, which results in significant increases of computational time compared to the semidefinite relaxation formulated in Sun and Dai (2015). The heuristic search based on refined RRT*, on the other hand, has faster convergence without guarantee of optimality. In real applications, according to the mission definition, one of the approaches can be selected to meet specific mission planning requirement.



(a) Case Three

(b) Case Four

(c) Case Five

(d) Case Six

Figure 6.3 Planned paths generated from IRM (solid) and best result of refined RRT* (dash) for cases three to six

Table 6.3 Comparison for four cases.

	f^*_{IRM}	$f^*_{refined}$ (15 simulations)	time of IRM(sec)	time of Refined RRT*(sec)
Case 3	15.73	15.92	1481.27	605.98
Case 4	14.86	15.29	1879.19	184.39
Case 5	14.43	14.52	424.83	212.24
Case 6	14.52	15.42	1124.73	188.90

6.2 Path Planning Of Unmanned Aerial Vehicles with Multi-Waypoint and Decision Parameter

In this section, several simulation cases are presented to demonstrate the capability of proposed method. In the first case, a simple multi-waypoint path planning problem with only two stops are considered. This example verifies the proposed approach is practical and successful. The second case considers a much more complex situation while the coordinates of four waypoints are provided substantially increasing the difficulty of the path planning problem. This example is also compares the performance with the combination of the Traveling Salesman Problem (TSP) toolbox and Nonlinear Programming (NLP) solver in MATLAB. In both cases, the parameters in IRM method are set as $[x_0, y_0] = [0, 0]^T$, $V = 1$, $u_{\max} = 0.5$, and $M = 125$. All sections are discretized into 5 nodes as the total number of nodes is $5n + 1$ where n is the number of sections. All simulation cases are run on a desktop computer with a 3.50 GHz processor and 16 GB RAM.

6.2.1 Two Waypoints Path Planning

The two given waypoints the UAV needs to pass by are $[0.4, 3]$ and $[2.9, 2.9]$. The optimal flight time from decision parameter method is $f^*_{IRM} = 6.011$ with overall computation time of 419.271 seconds. With two waypoints, the flight path is assembled by two continuous sections each containing 5 nodes, with 11 nodes in total included the starting point. The feasible trajectory generated from decision parameter method is shown in Figure 6.4. The result flight path validated the decision parameter method is able to produce a fairly smooth curve even with very small amount of discretized nodes representing each flight path section, yet, still keep the continuity of entire route.

6.2.2 Complex Waypoints Path Planning

The sequence of multiple waypoints in a simple situation can be determined without much computation. The complex multi-waypoint situation considered an initial point and four waypoints given as $[-1.089, -1.677]$, $[3, 0]$, $[3.109, 2.517]$ and $[-2.113, 4.532]$ where the sequence is

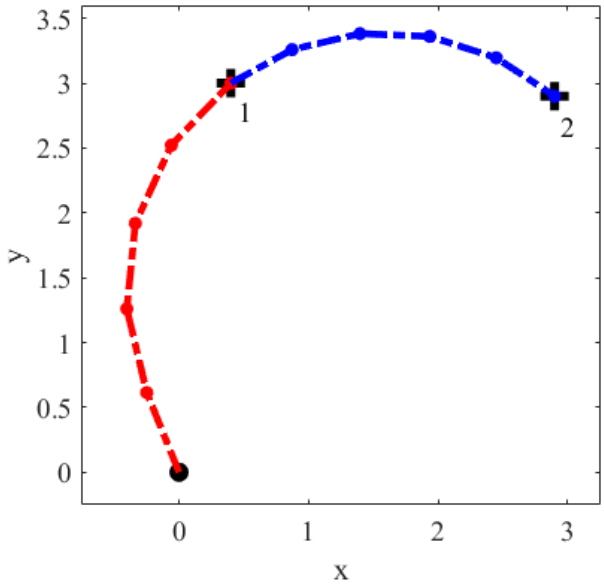


Figure 6.4 Simulation results of two waypoints situation.

no longer obvious. The proposed method was examined for the robustness and compared to a combined method that evaluates the sequence and kinematics of UAV individually.

The same condition was solved by a combined method of TSP toolbox and NLP solver in MATLAB. The combined method first analyzes the shortest closed tour from the given waypoints as a traditional TSP. The relatively longer link among the two that connect to the starting point is eliminated, and the sequence can be formed. Then, the NLP solver computes the flight path for each section regarding to the sequence from TSP and overall constraints.

Figures 6.5-6.6 show the paths generate from two approaches. The flight time performance from the decision parameter method is $f^*_{IRM} = 18.848$ while the one from combined method is $f^*_{combined} = 20.691$. The two method applied different sequence along with the trajectory of UAV. As utilizing shortest distance calculation of TSP, the combined method began the trip from the waypoint closest to the origin and produced a complete flight path. On the other hand, the decision parameter method used the route that is eliminated by TSP and generated a path that outperforms the combined method.

However, the trajectory from decision parameter method for complex condition is not as smooth as the simple condition, especially at the longer distance section. This can be improved

by increasing the number of discretized nodes to gain more flexibility, and potentially achieve better solution than using less discretized nodes. The computation time gradually increases when the number of waypoints and the number of discretized nodes increase.

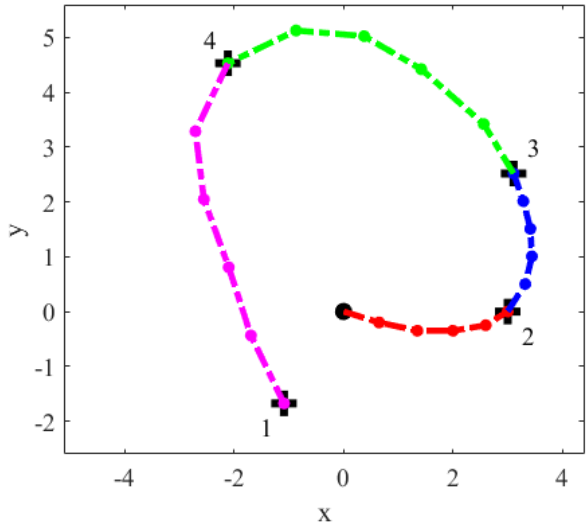


Figure 6.5 Simulation results of four waypoints situation with decision parameter method.

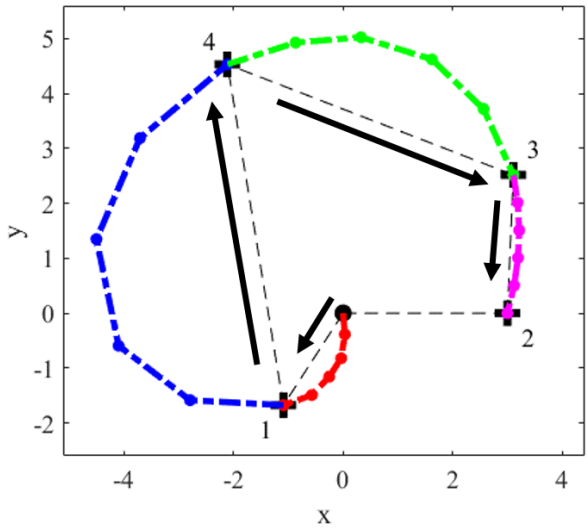


Figure 6.6 Simulation results of two waypoints situation with combined method.

6.3 Path Planning For Information Gathering Using Solar-Powered Unmanned Ground Vehicles

This section presents one real-world test environment, which was described in Kingry and Liu (2017). The area contains three high pressure sodium lights randomly placed in the 3.8 $\times$ 1.31 m^2 area, shown in Figure 6.7. The solar energy density distribution is mapped, and the modified RRT algorithm is used to generate motion plans and compared with a modified PSO algorithm. The resulting motion plans were then executed by the demonstration vehicle (Figure 6.8), described previously, five times each.

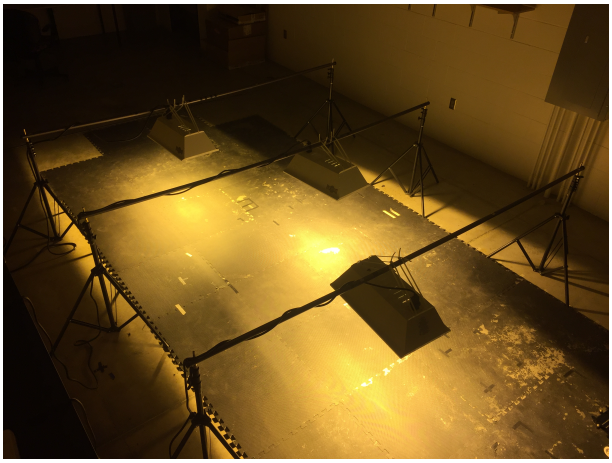


Figure 6.7 Indoor Testing Environment.

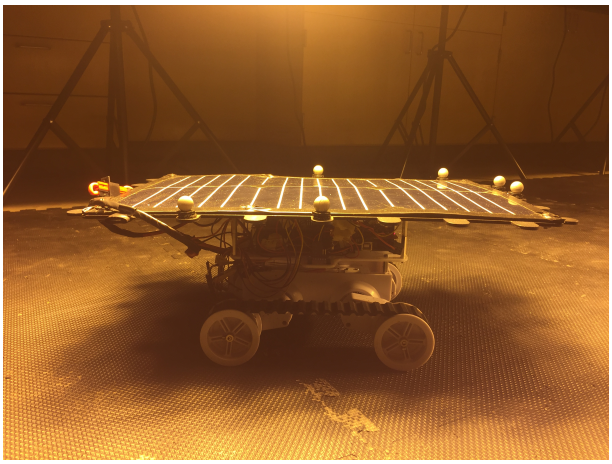


Figure 6.8 Demonstration Vehicle with 18W Solar Panel.

The optimized path, predicted variable history, and the recorded experimental data for the RRT is shown in Figures 6.9,6.10 and for the PSO shown in Figures 6.11, 6.12, where the predicted power harvested, P_{IN} is represented as (—), the power consumed, P_{OUT}, as (—), and the change in battery energy, $\Delta Battery$, as (—). The corresponding recorded experimental data is denoted as P_{IN} (- - -), P_{OUT} (- - -), and $\Delta Battery$ (- - -).

The anticipated travel time for the RRT motion plan was 57.07 seconds with a net energy increase of 10.26 J that covered 49.57% of the area, while the average experimental travel time was 56.23 seconds with an average net energy increase of 6.14 J. In comparison, the anticipated travel time for the PSO motion plan was 58.61 seconds with a net energy increase of 2.65 J that covered 62.5% of the area, while the average experimental travel time was 58.40 seconds with an average net energy increase of 4.516 J.

From the experimental data trials for both motion plans, the runtime match very closely with the predicted, only having errors of 1.47% and 0.358%, respectively for the RRT and PSO implementations. The experimental net energy of both motion plans are either slightly higher or lower than the predicted. These deviations from the predictions may be contributed by the vehicle's small deviations from the planned path. These deviations are due to the high-pressure sodium lights emitting near infrared light and conflicting with the VICON motion capture system used to determine the UGV's orientation and position in the testing environment, which result in an average 1.06% error in distance traveled and 0.80% error in angle turned. Measurement error in the voltage/current sensors, at worst $\pm 0.79\%$, could also contribute to the small experimental deviations observed.

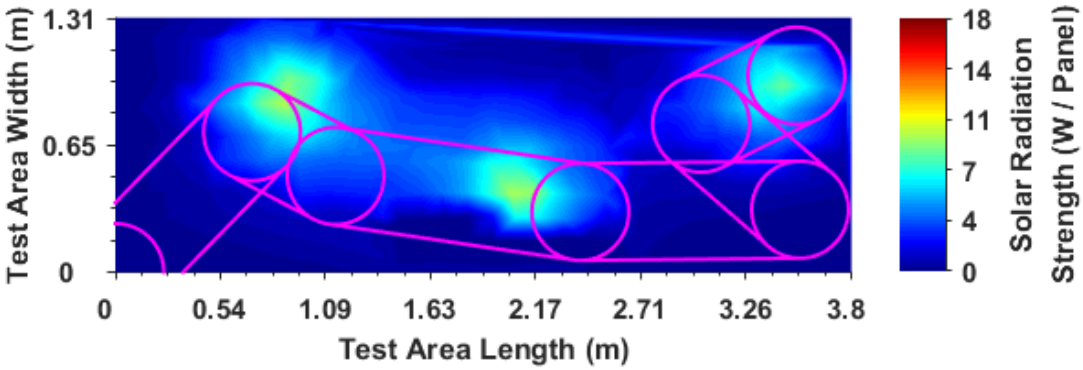


Figure 6.9 Constrained RRT Optimized Path Plan.

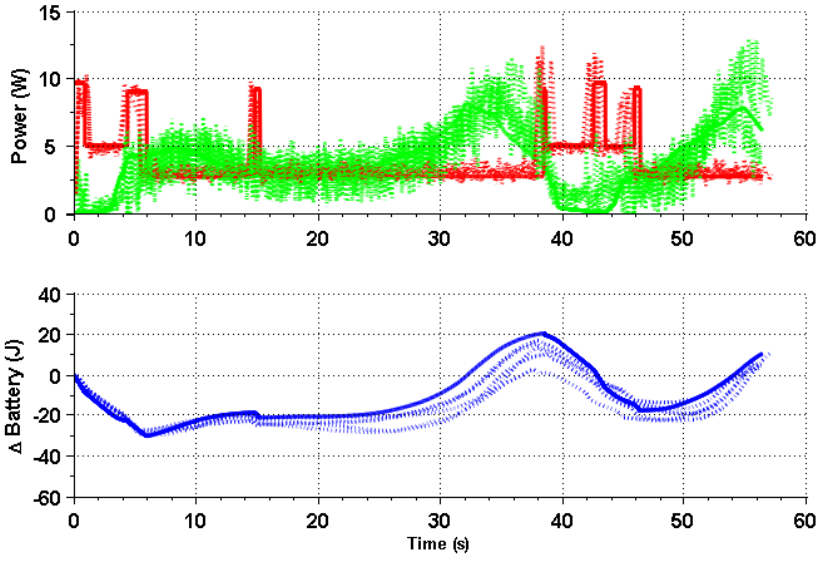


Figure 6.10 Constrained RRT Optimized Energy Schedule.

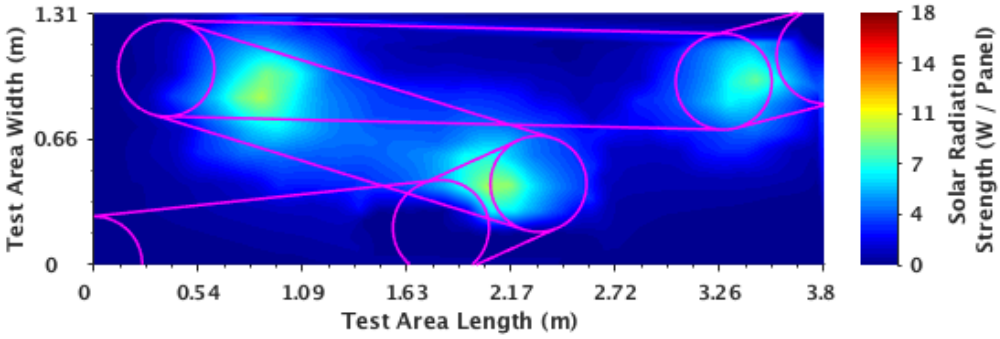


Figure 6.11 PSO Optimized Path Plan.

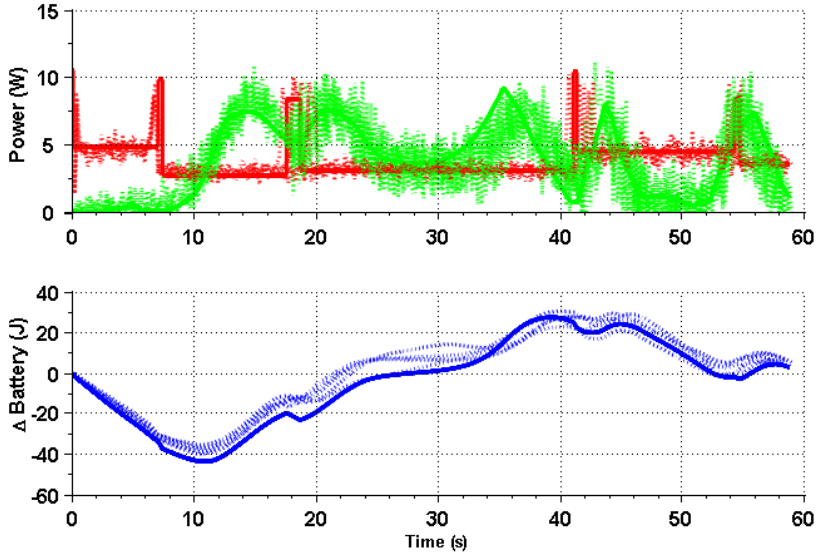


Figure 6.12 PSO Optimized Energy Schedule.

6.4 Path Planning Of A Multi-Robot Team with A Solar-Powered Charging Station

6.4.1 Results from Simulation and Experimental Testbed

In this section, results from simulation and experimental test trials are presented. All simulation and experiments presented are conducted in the indoor testing environment described in Nathaniel Kingry and Dai (2018). Due to the limited testing area, the cooperative mission requires the multi-robot team (6.13) to visit eight objective points within the area. The CHO method described in Chapter 5 is implemented using a 75% crossover rate and a 1% mutation rate for the GA settings. Parameters in PSO are set as $C_1 = C_2 = 2$ and $I = 1$. The planned paths for all UGVs are shown in Figure 6.14 and the time history of the associated power schedules for the UGVs with $z = 1, 2, 3$ are shown in Figures 6.15-6.17, respectively. The predicted harvesting power, P_{IN}, is represented as (—), the power consumed, P_{OUT}, as (—), and the change in battery energy, $\Delta Battery$, as (—). The associated experimental data is represented as (- - -) for each variable (P_{IN}, P_{OUT}, and $\Delta Battery$).

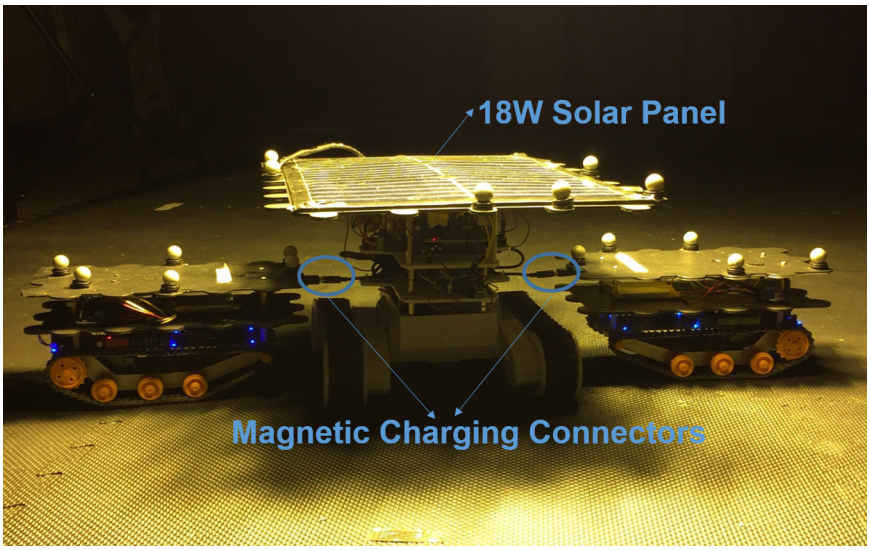


Figure 6.13 A multi-robot team including a solar-powered mobile charging station and two worker UGVs.

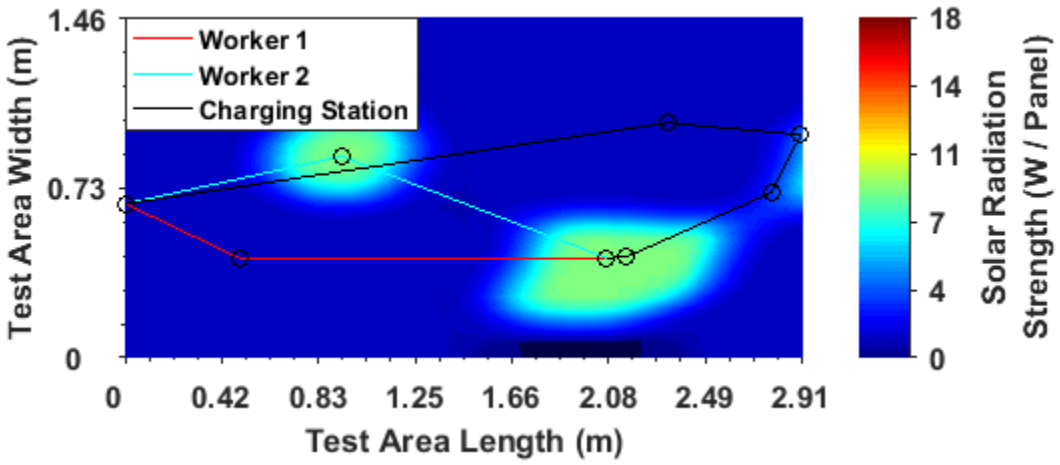


Figure 6.14 Optimized path plan.

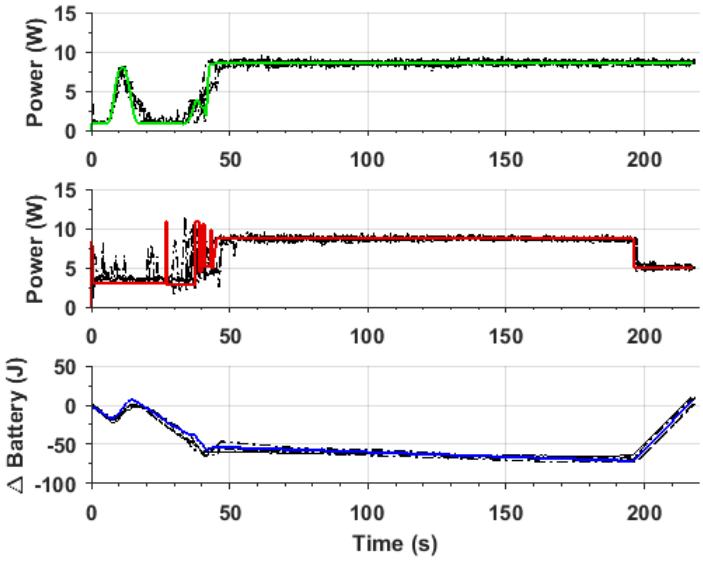


Figure 6.15 Energy schedule of the solar-powered mobile charging station, UGV 1.

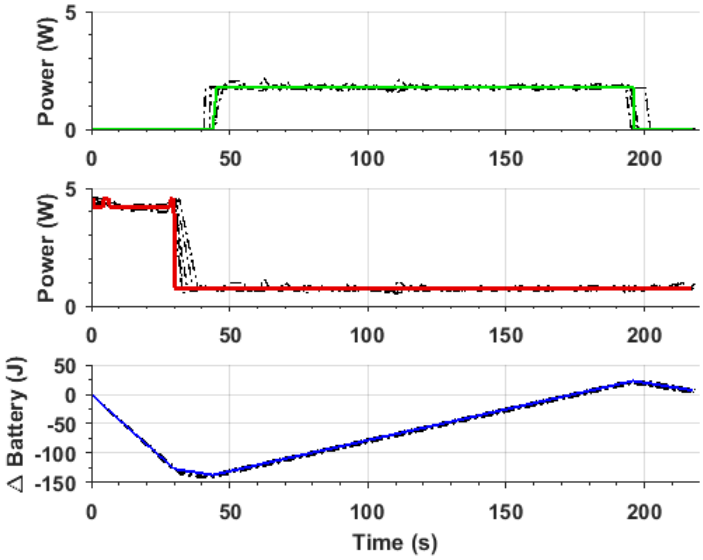


Figure 6.16 Energy schedules of worker UGV 2.

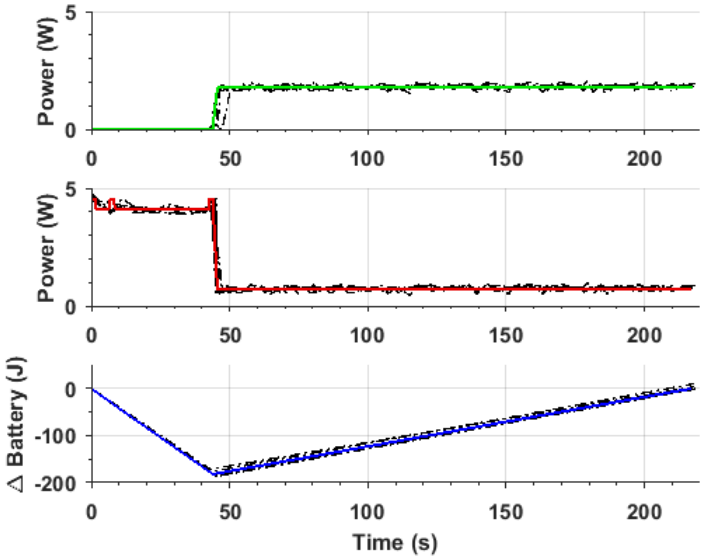


Figure 6.17 Energy schedules of worker UGV 3.

Table 6.4 Optimized Mission Characteristics

Vehicle	Travel Time (s)	$\Delta Battery$ (J)	Charging Time (s)
UGV $z = 1$ (Charging station)	43.54	5.236	-
UGV $z = 2$	29.88	6.956	152.13
UGV $z = 3$	43.85	0.105	173.46

Table 6.5 Statistical Analysis of CHO Method

	Average	Min	Max	Standard Deviation
Computation Time (s)	20.32	15.21	26.82	2.39
Mission Time (s)	217.70	217.32	219.67	0.64
UGV 1 $\Delta Battery$ (J)	11.69	5.24	21.88	5.07
UGV 2 $\Delta Battery$ (J)	6.41	0.11	6.96	1.88
UGV 3 $\Delta Battery$ (J)	0.55	0.11	5.66	1.58

The optimized plan with predicted mission characteristics is shown in Table 6.4. The predicted total mission time from the optimized plan is 217.32 seconds. The average total mission time from the five experimental trials is 217.81 seconds and the $\Delta Battery$ of the five trials for each UGVs with index $z = 1, 2, 3$ are 6.619 J, 3.653 J, and 1.942 J, respectively. A video of the experimental test is provided.

To verify robustness of the proposed method, statistical analysis has been performed that generates 50 virtual simulation trials to examine the computational performance. The results are shown in Table 6.5. Among the 50 trials, 46 trials find the same optimized solution and the other 4 cases generate solutions with a few seconds additional time cost.

6.4.2 Analysis of CHO and Errors from Experimental Testbed

Figure 6.18 demonstrated one simulation from CHO. The shown test run converged to the optimal in eight generations and the top three fitness solutions of each generation are listed. Compared to the virtual simulation results, the small variations observed in the experimental test data P_{in}, P_{out}, and $\Delta Battery$ can largely be attributed to the interaction between the solar lights and the Vicon system, which cause an average of 1.51% error in distance traveled and 1.20% error in turn angle. These errors were determined by following a similar process described in our previous work [Kaplan et al. (2016)]. However, instead of operating a single vehicle in the area, the error was determined under operating conditions. Other sources of error include measurement noise in the UGVs' voltage/current sensors and communication latency. In order to reduce communication latency, it was selected that the charging station data was sampled twice as often as other rovers, since the charging station experiences faster changes in the power gained due to the solar insolation of the environment.

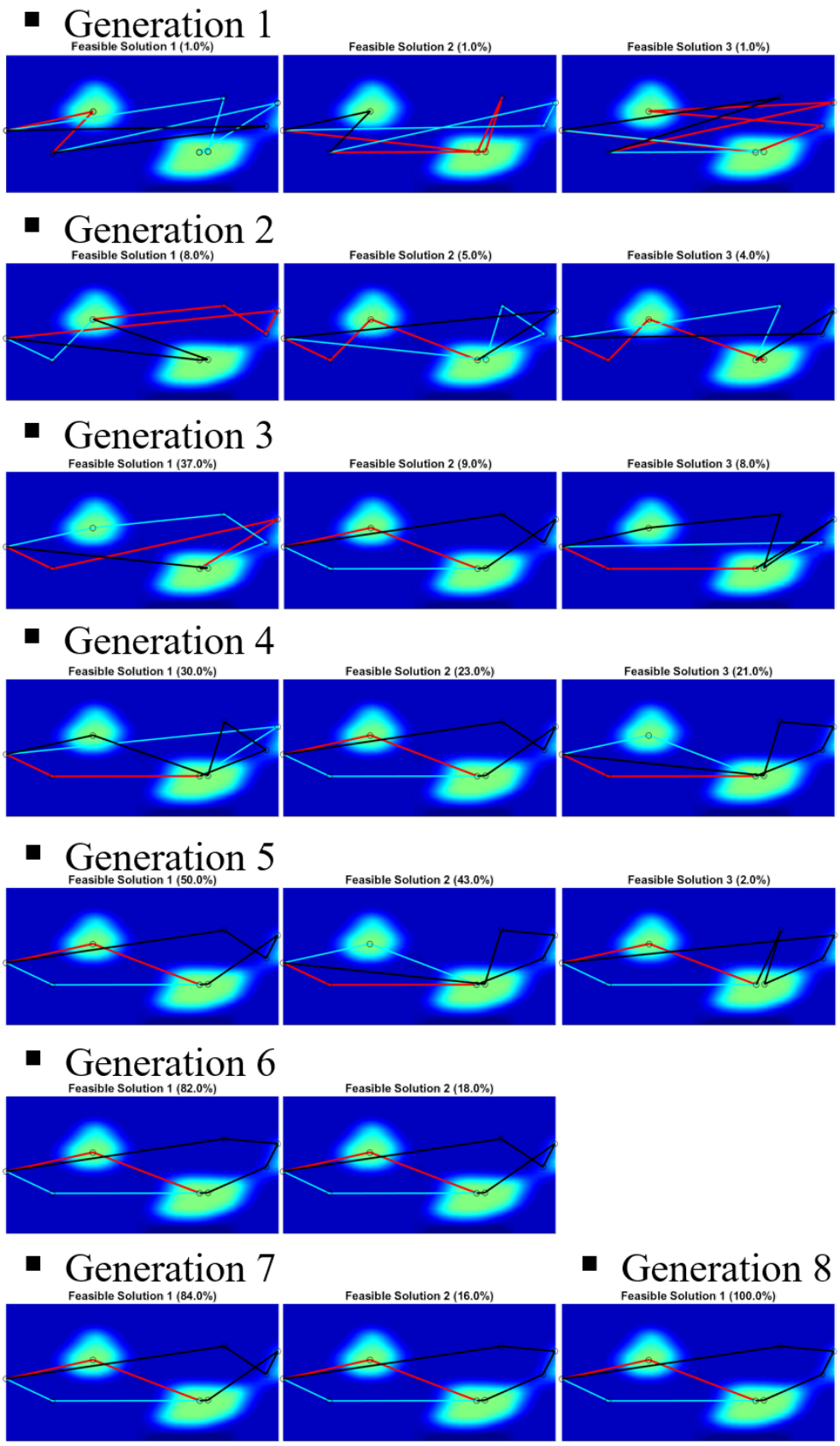


Figure 6.18 Example of a test simulation of CHO.

CHAPTER 7. CONCLUSION

This manuscript has studied four different constrained path planning problems and their potential for future real-world implementation. (1) The sampling-based heuristic search method refines the trajectory obtained from the optimized Rapidly-Exploring Random Tree algorithm to satisfy the kinematic constraints of UAV and reduce the cost of flight time for the point-to-point obstacle avoiding problems. (2) Extended multi-waypoint flight path is segmented and converted into Quadratically Constrained Quadratic Programming problem along with undetermined sequence and flight kinematics which is gradually examined by an iterative rank minimization method in the collision-free zone. (3) The constrained Rapidly-Exploring Random Tree method is modified from the original RRT to build a set of paths which awared of the surrounding energy resources and power consumption of the solar-powered UGV for evaluating optimal path with maximum area coverage. (4) A computationally efficient cascaded heuristic optimization method is proposed to search for the optimal mission plan for the described multi-Hamiltonian Path Problem with nonlinear dynamics and constraints of a cooperative team of UGVs including a solar-powered mobile charging station to prolong overall operation times. All scenarios have been verified by computer simulations or experimental results.

Future efforts will generalize proposed ideas to address uncertainty and robustness issue, and conduct real-world testing with proper process and equipment. In addition, improving the accuracy of algorithms for large-scale models and reducing computation time for real-time executions.

BIBLIOGRAPHY

Balkcom, D. J. and Mason, M. T. (2002). Time optimal trajectories for bounded velocity differential drive vehicles. *The International Journal of Robotics Research*, 21(3):199–217.

Borwein, J. M. and Lewis, A. S. (2010). *Convex analysis and nonlinear optimization: theory and examples*. Springer Science & Business Media.

Buniyamin, N., Wan Ngah, W., Sariff, N., and Mohamad, Z. (2011). A simple local path planning algorithm for autonomous mobile robots. *International journal of systems applications, Engineering & development*, 5(2):151–159.

Carter, A. E. and Ragsdale, C. T. (2006). A new approach to solving the multiple traveling salesperson problem using genetic algorithms. In *European Journal of Operational Research*, volume 175.1, pages 246–257.

Choset, H. (2007). Robotic motion planning: Rrts. Visited on 2015-09-27.

Couture-Beil, A. and Vaughan, R. T. (2009). Adaptive mobile charging stations for multi-robot systems. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*.

Dai, R. and Sun, C. (2015). Path planning of spatial rigid motion with constrained attitude. *Journal of Guidance, Control, and Dynamics*, 38(8):1356–1365.

Davies, T. and Jnifene, A. (2006). Multiple waypoint path planning for a mobile robot using genetic algorithms. In *2006 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*, pages 21–26. IEEE.

Eberhart, R. C. and Kennedy, J. (1995). A new optimizer using particle swarm theory. In *Proceedings of the sixth international symposium on micro machine and human science*, volume 1, pages 39–43. New York, NY.

Foo, J., Knutzon, J., Oliver, J., and Winer, E. (2006). Three-dimensional path planning of unmanned aerial vehicles using particle swarm optimization. In *11th AIAA/ISSMO multi-disciplinary analysis and optimization conference, Portsmouth, Virginia*.

Ghafurian, S. and Javadian, N. (2011). An ant colony algorithm for solving fixed destination multi-depot multiple traveling salesmen problems. In *Applied Soft Computing*, volume 11.1, pages 1256–1262.

Goldberg, D. (1989). Genetic algorithms in search, optimization and machine learning.

Hashim, M. S. M. and Lu, T.-F. (2009). Multiple waypoints trajectory planning with specific position, orientation, velocity and time using geometric approach for a car-like robot. In *Australasian Conference on Robotics and Automation (ACRA)*, pages 1–7.

Holland, J. (1975). Adaptation in natural and artificial systems.

Hollinger, G. and Sukhatme, G. (2014). Sampling-based robotic information gathering algorithms. *The International Journal of Robotics Research*, (10.1177/0278364914533443):1–17.

Jorris, T. R. and Cobb, R. G. (2009). Three-dimensional trajectory optimization satisfying waypoint and no-fly zone constraints. *Journal of Guidance, Control, and Dynamics*, 32(2):551–572.

Kaplan, A., Kingry, N., Top, J. V. D., Patel, K., Dai, R., and Grymin, D. (2016). Motion planning for persistent traveling solar-powered unmanned ground vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Karaman, S. and Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7):846–894.

Karaman, S. e. a. (2011). Anytime motion planning using the rrt*. In *2011 IEEE International Conference on Robotics and Automation*, pages 1478–1483. IEEE.

Keshmiri, S. (2011). Multi-robot, multi-rendezvous recharging paradigm: an opportunistic control strategy. In *Robotic and Sensors Environments (ROSE)*.

Kingry, N. and Liu, Y.-C. (2017). Mission planning for information gathering using solar-powered unmanned ground vehicles. In *55th AIAA Aerospace Sciences Meeting, AIAA SciTech Forum*.

Lavalle, S. M. (1998). Rapidly-exploring random trees: A new tool for path planning. Technical report.

LaValle, S. M. (2006). *Planning Algorithms*. Cambridge; New York: Cambridge University Press.

Luo, Z.-Q., Ma, W.-K., So, A. M.-C., Ye, Y., and Zhang, S. (2010). Nonconvex quadratic optimization, semidefinite relaxation, and applications. *IEEE Signal Process. Mag.(Special Issue on Convex Optimization for Signal Processing)*.

Mattei, M. and Blasi, L. (2010). Smooth flight trajectory planning in the presence of no-fly zones and obstacles. *Journal of guidance, control, and dynamics*, 33(2):454–462.

McGee, T. G. and Hedrick, J. K. (2006). Path planning and control for multiple point surveillance by an unmanned aircraft in wind. In *2006 American Control Conference*, pages 6–pp. IEEE.

Nathaniel Kingry, Yen-Chen Liu, M. M. B. S. Y. B. and Dai, R. (2018). Mission planning for a multi-robot team with a solar-powered charging station. In *IEEE International Conference on Intelligent Robots and Systems (IROS), submitted*.

Plonski, P. A., Tokekar, P., and Isler, V. (2013). Energy-efficient path planning for solar-powered mobile robots*. *Journal of Field Robotics*, 30(4):583–601.

Rathinam, S. and Raja, S. (2007). 5/3-approximation algorithm for a multiple depot, terminal hamiltonian path problem. In *Institute of Transportation Studies*.

Ross, I. M. and D'Souza, C. N. (2005). Hybrid optimal control framework for mission planning. *Journal of Guidance, Control, and Dynamics*, 28(4):686–697.

Šišlák, D., Volf, P., and Pechoucek, M. (2009). Flight trajectory path planning. In *Proc. of Intl. Scheduling and Planning Applications Workshop*, pages 76–83.

Soler, M., Olivares, A., and Staffetti, E. (2010). Hybrid optimal control approach to commercial aircraft trajectory planning. *Journal of Guidance, Control, and Dynamics*, 33(3):985–991.

Stephen Smith, Jana Tumova, C. B. and Rus, D. (2011). Optimal path planning for surveillance with temporal-logic constraints. *The International Journal of Robotics Research*, (10.1177/0278364911417911):1695–1708.

Sun, C. and Dai, R. (2015). An iterative approach to rank minimization problems. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 3317–3323. IEEE.

Vandenberghe, L. and Boyd, S. (1996). Semidefinite programming. *SIAM review*, 38(1):49–95.

Vasisht, S. and Mesbahi, M. (2015). Trajectory design and coverage control for solar-powered uavs. AIAA Guidance, Navigation, and Control Conference.

Wang, T., Cassandras, C. G., and Pourazarm, S. (2014). Energy-aware vehicle routing in networks with charging nodes. In *19th IFAC World Congress*, Cape Town, South Africa. The International Federation of Automatic Control.

Wawerla, J. and Vaughan, R. T. (2007). Near-optimal mobile robot recharging with the rate-maximizing forager. In *European Conference on Artificial Life*. ECAL 2007.