Optimal Control of Two-Wheeled Mobile Robots for Patrolling Operations

Walaaeldin Ahmed Ghadiry

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy at

Concordia University

Montréal, Québec, Canada

August 2015

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis proposal prepared

By: **Walaaeldin Ahmed Ghadiry**

Entitled: **Optimal Control of Two-Wheeled Mobile Robots for Patrolling Operations**

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

Doctor of Philosophy

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

Signed by the final examining committee:

Dr. Fariborz Haghighat, Chair
Dr. Mo Jamshidi, External Examiner
Dr. R. Sedaghati, External to Program
Dr. S. Hashtrudi Zad, Examiner
Dr. Wei-Ping Zhu, Examiner
Dr. Amir G. Aghdam, Thesis Supervisor
Dr. Youmin Zhang, Thesis Co-supervisor

Approved by
Chair of the Department of Electrical and Computer Engineering

Dean of the Faculty of Engineering and Computer Science

ABSTRACT

Optimal Control of Two-Wheeled Mobile Robots for Patrolling Operations

Walaaeldin Ahmed Ghadiry,

Concordia University, 2015

This work studies the use of the two-wheeled mobile robots in patrolling operations, and provides the most distance-efficient as well as time-efficient trajectories to patrol a given area. Novel formulations in the context of constrained optimization are introduced which can be solved using existing software. The main concept of the problem is directly related to the well-known Traveling Salesman Problem (TSP) and its variants, where a salesman starts from a base city and visits a number of other cities with minimum travel distance while satisfying the constraint that each city has to be visited only once. Finally, the salesman returns back to the starting base city after completing the mission. Two different patrolling configurations that are related to the TSP and its variants, namely the Single Depot multiple Traveling Salesman Problem (mTSP) and the Multidepot multiple Traveling Salesman Problem (MmTSP) are investigated. Novel algorithms are introduced for the trajectory planning of multiple two-wheeled mobile robots, either with two differential motors (which can turn on the spot) or with Dubins-like vehicles. The output trajectories for both types of wheeled robots are investigated by using a model predictive control scheme to ensure their kinematic feasibility for the best monitoring performance. The proposed formulations and algorithms are verified by a series of simulations using efficient programming and optimization software as well as experimental tests in the lab environment.

To my parents,

to my wife and my kids

to my brother and sister

for their love and sacrifice

ACKNOWLEDGEMENTS

First of all, I would like to express my profound gratitude to my supervisors, Dr. Amir G. Aghdam and Dr. Youmin M. Zhang for their invaluable support, encouragement, immense knowledge, experience and supervision without which this thesis would never have been materialized. I am very grateful for the research freedom I was given and more so for the necessary corrections along my road of inquiry. I feel fortunate to work closely with two exceptional and brilliant supervisors that have different expertise, style and interest .

Thanks to all my committee members for their helpful comments during all my PhD program, their comments helped me really to achieve the final version of my thesis.

In particular, I would like to thank Dr. Jalal Habibi for his support and motivation and with whom I co-authored my papers. I would like to acknowledge my colleagues at Concordia University, who provided me with some fruitful collaborations, in particular Narges Roofigari, Mohammad ALi Askri Hemmat and Yiqun Dong who helped me during my experimental work.

My deepest gratitude and appreciation go to my parents, my mother and my father, who have always encouraged me to move forward in my academic pursuits and who were beside me by their hearts day by day to complete this work. I am very grateful for their prayers and support in my entire life. My deepest gratitude extends to my brother and sister for their supports and goodwill.

Last but not least, I am so blessed to have an outstanding wife, Ghada. I would like to thank her for constant support, care, patience and faith in me. Without whom my life would never have been so glamorous and exciting.

TABLE OF CONTENTS

List of Figures

List of Tables

List of Abbreviations

WMR	**W**heeled **M**obile **R**obot
TWMR	**T**wo- **W**heeled **M**obile **R**obot
TSP	**T**ravelling **S**alesman **P**roblem
mTSP	Singledepot **m**ultiple **T**ravelling **S**alesmen **P**roblem
MmTSP	**M**ultidepot **m**ultiple **T**ravelling **S**alesmen **P**roblem
MTMmTSP	**M**inimum **T**ime **M**ultidepot **m**ultiple **T**ravelling **S**alesmen **P**roblem
TSPTW	**T**ravelling **S**alesman **P**roblem with **T**ime **W**indows
MPC	**M**odel **P**redictive **C**ontrol
RHC	**R**eceding **H**orizon **C**ontrol
MIP	**M**ixed **I**nteger **P**rogramming
QP	**Q**uadratic **P**rogramming
SEC	**S**ub-tour **E**limination **C**onstraint
ETSP	**E**uclidean **T**ravelling **S**alesman **P**roblem
DTSP	**D**ubins' **T**ravelling **S**alesman **P**roblem
AA	**A**lternating **A**lgorithm
2PA	**T**wo **P**oint **A**lgorithm
3PA	**T**hree **P**oint **A**lgorithm
LAA	**L**ooking **A**head **A**lgorithm
PA	**P**ulleys **A**lgorithm
QGV	**Q**uanser **G**round **V**ehicles
UGV	**U**nmanned **G**round **V**ehicles
UAV	**U**nmanned **A**erial **V**ehicles
UUV	**U**nmanned **U**nderwater **V**ehicles
DOF	**D**egrees **O**f **F**reedom
NAV	**N**etwork **A**utonomous **V**ehicle

Chapter 1

Introduction

1.1 Literature Review

Mobile robots are increasingly ubiquitous today, and are used in a variety of different applications, including exploration, search and rescue, materials handling and entertainment. While legged robots are able to step over obstacles, they are more complex to design and control due to the greater number of degrees of freedom. Particularly wheeled mobile robots (WMRs) are more energy efficient, have a simpler mechanical structure and simpler dynamics compared to legged robots [1]. This type of robots is often developed in applications concerning dangerous environments where human safety can be at risk. Among various types of WMRs, the two-wheeled mobile robots (TWMRs) with a third caster wheel have several advantages. For instance, in addition to their simple kinematics and dynamics, they have high maneuverability because of their ability to turn on the spot, although they are nonholonomic [2].

Coordinated teams of autonomous agents can effectively complete tasks requiring repetitive execution, such as monitoring oil spills [3], detecting forest fires [4], border surveillance [5], and environmental monitoring [6]. For example, monitoring a certain environment is typically carried out by assigning some viewpoints, the number of which is greater than robots to be used in the patrolling problem. The surveillance of an area of interest requires the robots to travel across the environment continuously while minimizing a prescribed cost function such as the travel distance [7, 8, 9]. The patrolling problem has many real life applications, among which are border patrol for reducing illegal immigration, fighting the threat of terrorism, maritime surveillance for reducing illegal fishing, forest fires monitoring, infrastructure protection, surveillance using unmanned aerial vehicles (UAV) and indoor patrolling including nuclear power stations, private roads and campuses [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34]. It is considered as one of the important real applications of the three main basic research areas of the navigation of the WMRs which are following a path, tracking a reference trajectory and point stabilization [35, 36].

The first criterion to be optimized in the patrolling operation after assigning the viewpoints is either the travel time or the travel distance via these viewpoints. If the travel distance is considered to be optimized, this will be strongly related to the conceptually relevant Travelling Salesman Problem (TSP) and two of its variants: the single depot multiple travelling salesmen problem (mTSP) and the multidepot multiple Travelling Salesman Problem (MmTSP). In the TSP, a salesman is required to start from a base city or a depot (the node from which the salesman starts his mission) and visits a number of cities with minimum travel distance, subject to the constraint that each city has to be visited only once and that the salesman should eventually return back to the starting base city after completing the mission [37]. However, in this problem, only one salesman (a robot in this case) is used to visit a given number of destinations (viewpoints), but if there exists a larger number of robots needed to visit these viewpoints such that each viewpoint is only visited by one of the given robots this leads to a variant problem of the TSP which is defined as the multiple Traveling Salesman Problem.

If all the robots start their trajectories from only one depot, this is the mTSP, while if the robots start their trajectories from more than one depot, this is the MmTSP. Compared to the TSP, the mTSP better represents real life routing and scheduling scenarios due to its generality in terms of the number of the salesmen [38]. Applications of the mTSP include crew scheduling [39], school bus routing problem [40], and, in particular, mission planning which arises in the context of autonomous mobile robots. The mission plan consists of the optimal path determination for each robot to accomplish the goals of the mission in the shortest time. In the mission planning, a variant of the mTSP is used where there are m robots and n viewpoints which must be visited by some robots and a base city to which all robots must eventually return [38]. Some applications of the mTSP in mission planning are reported in [41] and [42]. In [43], Yu *et al* model the planning of autonomous robots in cooperative robotics as a variant of the mTSP. Similarly, the routing problems that arise in the planning of UAV applications are investigated by Ryan *et al* [44] in the context of the patrolling problem. The MmTSP is considered as a generalization of the mTSP to the case where more than one depot exists and there are a number of salesmen at each depot [45, 46, 47, 48].

Minimizing the time needed to patrol an area can be another scenario of the patrolling operation. The problem turns out to be a new variant of the TSP, namely, minimum-time multidepot multiple Traveling Salesmen Problem (MTMmTSP).

Several articles investigate minimum distance trajectory in the patrolling problem or intend to minimize the waiting time of salesmen in TSP with time windows (TSPTW), which is closely related to the underlying patrolling problem. In the TSPTW problem, each customer has a service time and a time window between the ready time and due date. Each customer must be visited before its due date; otherwise, the tour is said to be infeasible. If, on the other hand, a vehicle arrives before the above-mentioned time window, it must wait. The TSPTW can be modeled as a routing or a scheduling problem. In routing tasks, it is desired to find a route to visit a number of viewpoints, starting and ending at the same depot, with the constraint that each viewpoint must be visited in a time window. In scheduling jobs on a single machine, on the other hand, setup times are sequence dependent, and each job has a release and due date. In this case, the objective function is to minimize the tour-completion time, or the so-called makespan [49, 50, 51, 52].

In all the previous discussed patrolling trajectories, the wheeled robots are assumed to be agile with no slipping effect and can change directions quickly relative to the inter-activity travel times, the time between two activities can then be described approximately in terms of the distances between the activities. The resulting sequencing problem becomes an Euclidean Travelling Salesman Problem (ETSP). Exact algorithms, heuristics as well as constant factor approximation algorithms with polynomial time requirements are available for the ETSP, however, when vehicles have significant kinematic constraints such as limited turning radius, and the inability to move in a reverse direction, the paths obtained from ETSP solutions are hard to approximate with flyable trajectories. Thus, the ETSP solution provides poor estimates of actual travel time and vehicle location [53].

A classical model for two-dimensional motion of vehicles with kinematic constraints is Dubins' model [54, 55, 56, 57, 58] ; we refer to these models as Dubins' vehicles. The solution of TSP problems with Dubins' vehicles (DTSP) was recently considered in many articles [59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70]. This means that the distance between any two pairs of viewpoints depends on the incoming and outgoing directions of the trajectory through the node pair. Thus, the distances cannot be precomputed considering only the location of the nodes. Extensions of the TSP formulation for Dubins' vehicles are possible by creating multiple nodes for each physical waypoint representing possible discrete travel orientations, but these extensions result in significantly larger TSPs, making the real-time solution of the path planning problem impractical. Alternative approaches for DTSP proposed in [53, 60, 61] is to use a hierarchical approach: First, determine the sequence of the viewpoints by solving for the exact optimal ETSP then find a sub-optimal path through the sequence of points that satisfies the nonholonomic kinematic constraints based on DTSP.

After obtaining the desired optimal trajectories for the TWMRs or for the Dubins' vehicles to be tracked in the patrolling problem, a controller is needed to feasibly track these trajectories. Due to the nonholonomic features of the TWMRs, the design of a feedback controller for such robots became a challenging task. According to Brockett's result [71], nonholonomic systems cannot be stabilized by a continuously differentiable, time-invariant state feedback control law. Trajectory tracking in particular is of potential interest in various applications [72]. Here the term trajectory refers to the path that a robot should traverse as a function of time. A trajectory planner generates the appropriate trajectory for arriving at a particular location, patrolling in a prespecified area, etc., and at the same time avoiding collisions with different kinds of obstacles. To obtain a feasible trajectory, i.e., a trajectory that a robot is able to track, the planner also needs to consider various physical and dynamic limitations of the robot such as its velocity and acceleration limits. A trajectory can be generated in real-time on the basis of current sensor readings or generated in advance on the basis of operating environment map [73].

Motion control of WMRs has been and still is the subject of numerous research studies. Many nonlinear techniques have been proposed in the literature such as dynamic feedback linearization [74], sliding mode control [75], backstepping techniques [76], Lyapunov Techniques [77] etc., to name only a few. Model predictive control (MPC) also referred to as receding horizon control (RHC) has been widely adopted in process control industry for decades because control objectives and operating constraints can be integrated explicitly in the optimization problem that is solved at each instant. Many successful MPC applications have been reported in the last three decades. Although the method is traditionally applied to plants with sufficiently slow dynamics to permit computations between samples, with the advancement of faster computers, it has become possible to implement MPC on systems governed by faster dynamics including WMRs as in [78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99] and there seem to be a promising future for the application of MPC to WMRs .

1.2 Motivation

As introduced earlier, several articles tackled the mTSP and the MmTSP problems by using prespecified starting depots as in [46, 47]. Three main assumptions are typically used: (i) a set of starting depots for the robots is prespecified; (ii) a set of customer nodes that contains the rest of all the other nodes which are to be visited by the robots throughout the process is prespecified; (iii) The number of robots at each starting depot is also prespecified.

Although the patrolling problem has been widely studied in the literature, the existing results always depend on the previous assumptions. This gives motivation to seek for the distance-efficient and time-efficient patrolling trajectories without any prior assumptions on the depots and the initial number of robots at each starting depot.

As discussed earlier, the MPC has a promising future for the control of WMRs, this gives motivation to apply this type of control on the TWMRs to feasibly track the optimally-obtained patrolling trajectories. This can be applied for the optimal ETSP and also the sub-optimal DTSP trajectories.

1.3 Thesis Contributions

The main contributions of this thesis arise from the investigation of patrolling operations. This is achieved by introducing distance-efficient as well as time-efficient formulations and algorithms either for TWMRs or Dubins' vehicles. The formulations and algorithms introduced in this research are verified by a series of simulations using well-known programming and optimization software as well as practical implementations in the lab.

The contributions of the thesis are as follows:

- Introducing a framework for distance-efficient trajectory optimization of non-prespecified mTSP-based patrolling problems.
- Introducing a generalized formulation for distance-efficient trajectory optimization of non-prespecified MmTSP-based patrolling problems.
- Introducing a minimum-distance formulation for trajectory optimization of unknown MmTSP-based patrolling problems.
- Introducing a minimum-distance formulation for trajectory optimization of non-TSP-based patrolling problems for unknown number of staring depots and robots.
- Introducing a minimum-time formulation for trajectory optimization of non-prespecified MmTSP-based patrolling problems.
- Introducing an algorithm for obtaining the minimum-time optimal trajectory using minimum-distance optimal trajectory for non-prespecified MmTSP-based patrolling problems.
- Introducing a minimum-time formulation for trajectory optimization of unknown MmTSP-based patrolling problems.

- Introducing an algorithm for obtaining the minimum-time optimal trajectory using minimum-distance optimal trajectory for unknown MmTSP-based patrolling problems.
- Introducing an algorithm for obtaining feasible sub-optimal Dubins' trajectory (DTSP) based on the optimally-obtained (ETSP) trajectory.
- Driving upper bounds for some existing sub-optimal algorithms in the literature used to obtain the DTSP from the ETSP and introducing enhancements to them.

1.4 Thesis Outline

The thesis is organized as follows:

Chapter 2 presents the minimum-distance patrolling problem when multiple robots perform the patrolling operation start from a single depot (mTSP). Two new formulations are presented on the basis of the non-prespecified starting depot with a detailed comparison between their simulation results and the commonly-used formulation results. The computational time of the two presented formulations is compared to each other at the end of the chapter.

Chapter 3 presents the minimum-distance patrolling problem when multiple robots performing the patrolling operation start from multiple depots (MmTSP). First, a generalized formulation is presented on the basis of non-prespecifying the starting depots as well as the initial number of robots at each starting depot with a comparison between its simulation results and the commonly-used formulation results. Second, two new generalized formulations are presented for the same problem when the optimal number of starting depots as well as the optimal number of robots needed for the patrolling operation are unknown. One of these two formulations is MmTSP based, while the other is not.

Chapter 4 presents the minimum-time patrolling problem when multiple robots performing the patrolling operation start from multiple depots (MTMmTSP). First, a new formulation is presented on the basis of the non-prespecified starting depots and robots at each starting depot. Second, a new algorithm is introduced to obtain the minimum-distance trajectory among the possible minimum-time trajectories of the previous problem. Third, a new formulation and a new algorithm are presented analogous to the previous two problems but for the case where the number of depots and robots are unknown.

Chapter 5 introduces a new algorithm for softening the optimally-obtained sharp-turning patrolling trajectories to result in sub-optimal trajectories that can be continuously tracked by Dubins' vehicles. The new algorithm is compared theoretically to other works to elaborate its efficacy. Some upper-bounds and enhancements are introduced to the existing works in the literature.

Chapter 6 introduces the experimental results. This is done by using MPC controller in the trajectory tracking of the linearized model of the TWMRs to track the optimally-obtained trajectories in three experiments. The first two are for tracking the exact sharp-turning patrolling trajectories optimally-obtained in Chapters 2-4 using TWMRs dynamics. The third is for testing the new algorithm introduced in Chapter 5 for softening the optimally-obtained sharp-turning patrolling trajectories into sub-optimal trajectories that can be tracked by Dubins' vehicles.

Finally, Chapter 7 is devoted to the conclusions and future work.

Chapter 2

A Framework for Distance-Efficient Trajectory Optimization in Patrolling Problem with Non-prespecified Starting Depot

In this chapter, two new formulations are presented for trajectory optimization in the patrolling problem. It is assumed that the starting depot is not prespecified; an assumption that distinguishes the present work from the existing literature [38, 39, 40, 41, 42, 43, 44]. A number of viewpoints are assigned to be visited in a certain sequence to minimize the total travel distance. The problem turns out to be a variant of the well-known Traveling Salesmen Problem (TSP), namely the Single depot multiple Traveling Salesmen Problem (mTSP). The only information known *a priori* is the total number of robots in addition to the number of viewpoints. It is assumed that the starting depot is among the nodes (viewpoints) to be visited, and that the robots have the same nonlinear dynamics as the TWMRs. The latter assumption makes it possible to turn on the spot, which means that the robots can move on sharp-edged paths. Furthermore, there are no physical constraints that can affect the motion trajectories. The efficiency of the motion-planning strategy based on the proposed new formulations for the patrolling problem with non-prespecified starting depot is evaluated and compared to the conventional case, where the starting depot is prespecified. Simulations confirm that under the proposed method robots travel a shorter distance and complete the patrolling mission more rapidly.

The rest of this chapter is organized as follows. The problem statement of the Single Depot multiple traveling Salesmen Problem is introduced in Section 2.1. Section 2.2 presents the proposed frameworks and the new formulations. The simulation results are provided in Section 2.3 in a comparable fashion to show the efficacy of the proposed formulations.

2.1 Problem Statement

Consider a complete undirected graph $G(V, E)$, where $V = \{v_1, \ldots, v_n\}$ denotes a set of n viewpoints through which m robots, $m < n$, will perform a patrolling operation for monitoring an area, starting from a prespecified depot. E denotes the set of all edges connecting any two nodes, and is used to represent the motion trajectories to be tracked by the robots. Let $[c_{ij}]$ represent the cost matrix corresponding to the path lengths of all edges between viewpoints v_i and v_j. The distance between two connected nodes is adopted as the edge weight, which implies $c_{ij} = c_{ji},\ \forall (i, j) \in E$. This choice of edge weight also satisfies the triangular inequality, i.e., the shortest path between viewpoints constitute a suitable choice [7]. More precisely, for any three nodes (i, j, k), $c_{ij} + c_{jk} \geq c_{ik},\ \forall i, j, k \in V$ [38].

The above formulation is for the traditional patrolling problem, where the starting depot is prespecified. In the new formulations proposed here, only the set of n viewpoints $V = \{v_1, \ldots, v_n\}$ is assumed to be given. In other words, there is no prespecified depot. The minimum distance trajectories are required to be computed such that each robot follows an appropriate trajectory connecting a subset of the viewpoints starting from the depot (which is not prespecified) and returning back to it. There are two cases of interest: (i) when a return trip is allowed, i.e., the robot can visit only one viewpoint and return to its previous viewpoint, and (ii) when a return trip is not allowed. The restriction in the second case applies to applications such as pickup and delivery, where visiting only one node is not allowed. It will be shown later the minimum distance trajectories in the new formulation highly depend on the starting depot's position. This will have a significant impact on the travel distance, and hence, on the operation life time.

2.2 Proposed Frameworks

Problem 1: Single Depot multiple Traveling Salesmen Problem (mTSP) with non-prespecified starting depot, and with return trip allowed. For the aforementioned graph $G(V, E)$ and cost matrix $[c_{ij}]$, the Single Depot multiple Traveling Salesmen Problem represented in the literature can be formulated in an optimization framework as follows: Let a link connecting two arbitrary nodes v_i, v_j on the trajectory from a prespecified starting depot be represented by a binary variable x_{ij}, which is equal to 1 if the trajectory is optimal and 0 otherwise. The cost function to be minimized can be represented as the overall sum of the product of each element in $[c_{ij}]$ and its corresponding variable x_{ij}, and can be formulated as:

$$\min_{x_{ij}} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} c_{ij} x_{ij} \tag{2.1}$$

The number of departures from and arrivals to the starting depot, which is defined *a priori* here as node #1, is denoted by m as described below:

$$\sum_{j=2}^{n} x_{1j} = m \tag{2.2}$$

$$\sum_{j=2}^{n} x_{j1} = m \tag{2.3}$$

In addition, the number of departures from and arrivals to any other node not including the starting depot is one, i.e.:

$$\sum_{\substack{i=1 \\ i \neq j}}^{n} x_{ij} = 1, \qquad j \in V - \{1\} \tag{2.4}$$

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} x_{ij} = 1, \qquad i \in V - \{1\} \tag{2.5}$$

On the other hand, one of the most important constraints in the Traveling Salesman Problem (TSP) is to avoid any sub-tour, which is a closed trajectory that does not include the starting depot as the starting and ending point. This could be presented in the optimization framework as an inequality that contains two variables corresponding to each link between any two nodes v_i, v_j. Thus, the differences between the variables corresponding to a sub-tour not including the starting depot will always contradict the inequality, preventing such sub-tours from being formed. The aforementioned sub-tour elimination inequality is:

$$\begin{aligned} u_i - u_j + (n - m)\, x_{ij} \leq n - m - 1, & \\ i \in V, \quad j \in V - \{1\}, \quad i \neq j \end{aligned} \tag{2.6}$$

where u_i and u_j are the two artificial variables that prevent the formation of such sub-tours among nodes not including the starting depot.

Remark 2.1. *A similar formulation is presented in [46] but with a set of constraints on the maximum and minimum number of nodes that a robot has to visit. The equations given above do not include such restrictions.*

The new formulations presented in this work represent a novel optimization framework. It is assumed that for a given set of viewpoints, only the total number of robots is given without pre-defining a certain starting depot. The first formulation has been represented in [100], where the optimization framework is repeated at each possible depot and the final optimal solution is the minimum among all the optimal solutions at each possible depot. This formulation will be referred to as the *greedy algorithm* in this chapter. However, in the second formulation, the optimization framework is solved only once, seeking for the optimal depot that results in trajectories with minimum overall travel distance.

Starting with the first proposed greedy algorithm, denote the index of the starting depot by k, and note that this depot can be any of the viewpoints. Note also that the optimal trajectories depend on k, with an optimal solution $D^*(k)$. The problem is solved by considering every k as a candidate for the optimal index and solving a mixed-integer linear programming problem, and then comparing the values obtained for different cases. The above optimal solution can be described by a minimum cost function as follows:

$$\min_k D^*(k) = \min_k \min_{x_{ij}} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j\neq i}}^{n} c_{ij} x_{ij}, \quad k \in V \tag{2.7}$$

The number of departures from and arrivals to the starting depot which depends on k is denoted by m, as expressed below:

$$\sum_{\substack{j=1 \\ j\neq k}}^{n} x_{kj} = m \tag{2.8}$$

$$\sum_{\substack{j=1 \\ j \neq k}}^{n} x_{jk} = m \tag{2.9}$$

In addition, the number of departures from and arrivals to any other node not including the starting depot is equal to one, i.e.:

$$x_{kj} + \sum_{\substack{i=1 \\ i \neq k}}^{n} x_{ij} = 1, \qquad j \in V, j \neq k \tag{2.10}$$

$$x_{ik} + \sum_{\substack{j=1 \\ j \neq k}}^{n} x_{ij} = 1, \qquad i \in V, i \neq k \tag{2.11}$$

The sub-tour elimination constraint (SEC) can be presented by:

$$\begin{aligned} u_i - u_j + (n-m)\, x_{ij} \leq n - m - 1, & \\ & i, j \in V,\ i \neq j,\ j \neq k \end{aligned} \tag{2.12}$$

Note that the binary variable x_{ij} is equal to one if the edge (i, j) is optimal, and is zero otherwise. Equations 2.7 - 2.12 therefore compose the new greedy algorithm for the return trip allowed case.

The second proposed formulation uses an improved optimization framework that is solved only once to obtain the same optimal result as the first proposed formulation. In this case, the binary variables are denoted by $x_{ijk}, i, j, k \in V$. If a link between v_i and v_j and originating from depot v_k constitutes an optimal trajectory, then x_{ijk} is equal to one; otherwise, it is zero. A set of n auxiliary binary variables ω_k is introduced, where each variable corresponds to one of the possible choices of depots among the n viewpoints in such a way that ω_k is equal to one if v_k is an optimal starting depot, and zero otherwise.

The cost function to be minimized can be reformulated as:

$$\min_{x_{ijk}} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \sum_{k=1}^{n} c_{ij} x_{ijk} \tag{2.13}$$

The number of departures from and arrivals to the starting depot (which is not prespecified) is denoted by m, as given below:

$$\sum_{\substack{j=1 \\ j \neq k}}^{n} x_{kjk} = m\, \omega_k, \qquad k \in V \tag{2.14}$$

$$\sum_{\substack{j=1 \\ j \neq k}}^{n} x_{jkk} = m\, \omega_k, \qquad k \in V \tag{2.15}$$

Moreover, a new constraint on the sum of the new auxiliary variables is given by:

$$\sum_{k=1}^{n} \omega_k = 1 \tag{2.16}$$

which implies that there is only one optimal starting depot. In addition, the number of departures from and arrivals to any node other than the starting depot is one, i.e.:

$$\sum_{\substack{i=1 \\ i \neq j,k}}^{n} x_{ijk} = \omega_k, \qquad j, k \in V, j \neq k \tag{2.17}$$

$$\sum_{\substack{i=1 \\ i \neq j,k}}^{n} x_{jik} = \omega_k, \qquad j, k \in V, j \neq k \tag{2.18}$$

The SEC can be presented by:

$$\begin{aligned} u_i - u_j + (n-m)\, x_{ijk} \leq n - m - 1 + (1 - \omega_k), \\ i, j, k \in V,\ i \neq j,\ j \neq k \end{aligned} \tag{2.19}$$

The above inequality is different from inequality (2.6) as a new expression is added in (2.19). In this new expression, where v_k is the optimal depot if $\omega_k = 1$, for which the set of inequalities representing an optimal tour will be valid. Equations 2.13 - 2.19 therefore compose the new formulation for the non-prespecified depot with the return trip allowed case.

Problem 2: Single Depot multiple Travelling Salesmen Problem (mTSP) with non-prespecified starting depot, and with return trip not allowed. Typically in applications such as pickup and delivery, the robot is not allowed to return to a viewpoint before visiting at least two other nodes. The inequalities describing that the return trip is not allowed in the first and second new proposed formulations are given by:

$$x_{kj} + x_{jk} \leq 1, \qquad\qquad j, k \in V,\ j \neq k \tag{2.20}$$

and

$$\sum_{\substack{k=1 \\ k \neq j}}^{n} x_{kjk} + \sum_{\substack{k=1 \\ k \neq j}}^{n} x_{jkk} \leq 1, \qquad j, k \in V \tag{2.21}$$

respectively. Problems 1 and 2 are both expressed in mixed integer programming framework with linear constraints. Thus, they are convex and always have a feasible optimal solution, which can be obtained using solvers such as MOSEK optimization software [101] and Gurobi Optimizer 6.0 [102]

2.3 Simulation Results

Consider the patrolling problem for a 20m by 20m field, where a set of 10 nodes (viewpoints) are to be visited by 3 robots ($n = 10$ and $m = 3$). It is desired to find the minimum-distance trajectories using the formulation for the prespecified set of starting depots as well as the proposed formulations for the non-prespecified starting depots. MATLAB was employed with MOSEK optimization software [101] to obtain all the results using Intel Core i7-3537U @ 2.00GHz processor with 8 GB RAM.

Scenario 1. In this scenario, node 1 is assumed to be the prespecified starting depot. It is also assumed that the return trip is allowed. The optimal result in this case is demonstrated in Fig. 2.1 and the total travel distance is about 101m.

Scenario 2. In this scenario, node 6 is assumed to be the prespecified starting depot. In fact, it can be verified that this is the worst-case scenario for the given node configuration as far as the minimum travel distance is concerned. Again, it is assumed that the return trip is allowed. The optimal result in this case is depicted in Fig. 2.2, and the total travel distance is about 109m.

Scenario 3. In this scenario, it is assumed that unlike the previous two scenarios, the starting depot is not prespecified, and that, like the previous scenarios, the return trip is allowed. Using the proposed formulation, the optimal trajectory depicted in Fig. 2.3 is obtained. The total travel distance in this case is about 78m.

Table 2.1 provides a comparison between the optimal travel distance in the above three scenarios. As shown in this table, the reduced travel distance in only one patrolling tour with a non-prespecified starting depot could range between 23m (23.08% of the total travel distance) and 30m (28.34% of the total travel distance), compared to scenarios 1 and 2, respectively. Such a significant reduction in travel distance can also lead to major improvement in the operation time of the robots due to the increase in the life time of the batteries (note that typically the patrolling operation can be repeated for a long period of time). Table 2.1 also shows that the computation time required for the new formulation is longer than that for the conventional method. However, it is important to note that the computation is

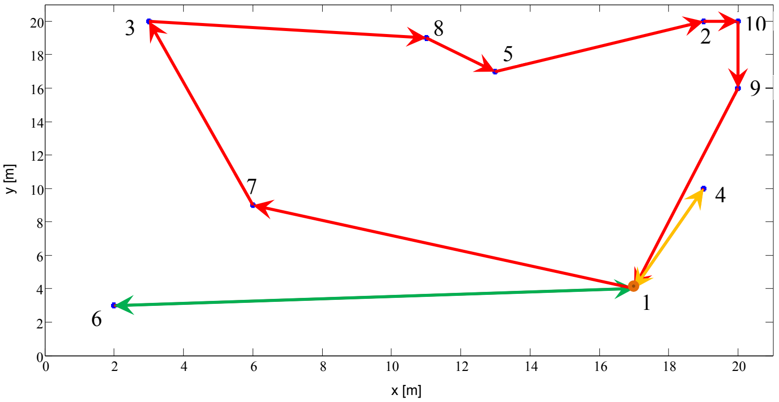


Figure 2.1: The Optimal mTSP for the case where node 1 is the starting depot, and the return trip is allowed $(n = 10, m = 3)$.

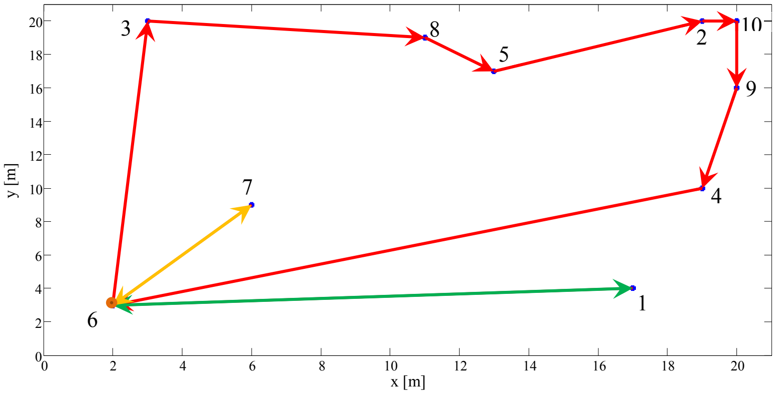


Figure 2.2: The Optimal mTSP for the case where node 6 is the starting depot (worst-case scenario), and the return trip is allowed $(n = 10, m = 3)$.

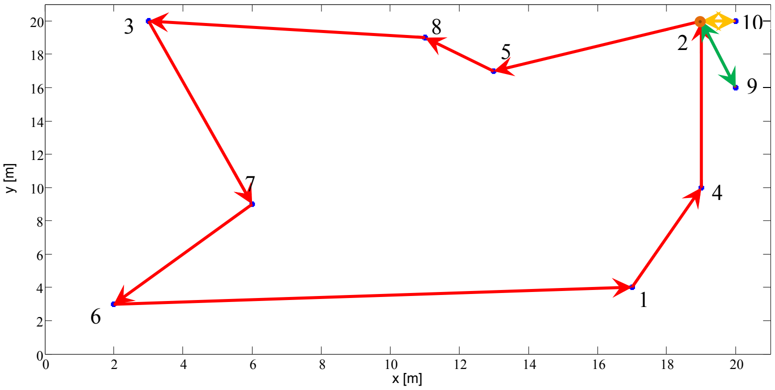


Figure 2.3: The Optimal mTSP for the case where the starting depot is not prespecified and the return trip is allowed $(n = 10, m = 3)$.

performed offline, which is not completely unimportant, but has no impact on the patrolling operation.

Table 2.1: Comparison between the results of scenarios 1-3.

	Scenario 1	Scenario 2 (worst-case scenario)	Scenario 3 (non-prespecified starting depot)
Starting depot	node 1 (prespecified)	node 6 (prespecified)	node 2 (calculated)
Total travel distance per tour	101.1687m	108.5846m	77.8158m
Computation time	0.23s	0.2s	0.44s
Reduction in travel distance per tour	23.3529m (23.08%)	30.7688m (28.34%)	–

Figs. 2.4-2.6 show the optimal results for the case where the return trip is not allowed, analogous to Figs. 2.1-2.3, with a comparison summarized in Table 2.2. As expected, due to the additional constraint on the trajectory (concerning the return trips), the overall travel distance in this case is more than the previous case (when comparing same scenarios in both cases). Furthermore, the optimal starting depot when it is not prespecified is node 5 (see Fig. 2.6), which results in major reduction in total travel distance (22% compared to scenario 1 and 32% compared to scenario 2). It is worth noting that here the worst-case scenario with the return trip not allowed is again when the starting depot is node 6 (see Fig. 2.5).

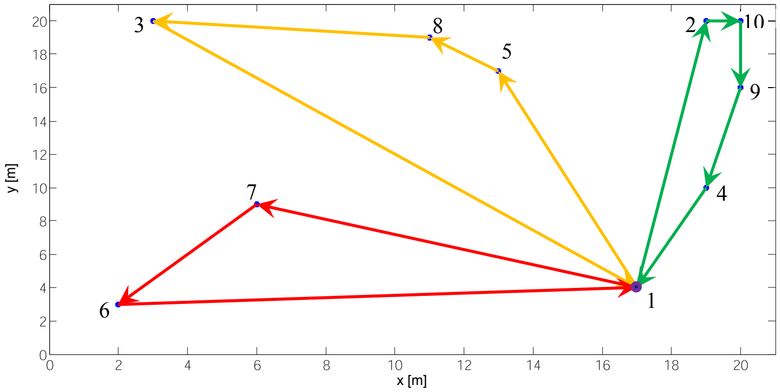


Figure 2.4: The Optimal mTSP for the case where node 1 is the starting depot, and the return trip is not allowed ($n = 10, m = 3$).

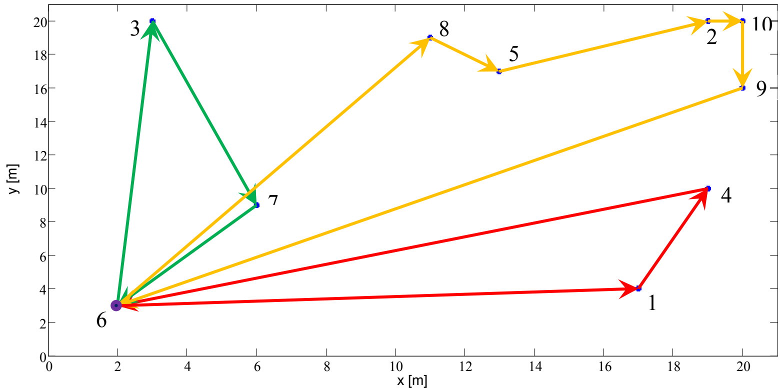


Figure 2.5: The Optimal mTSP for the case where node 6 is the starting depot (worst-case scenario), and the return trip is not allowed ($n = 10, m = 3$).

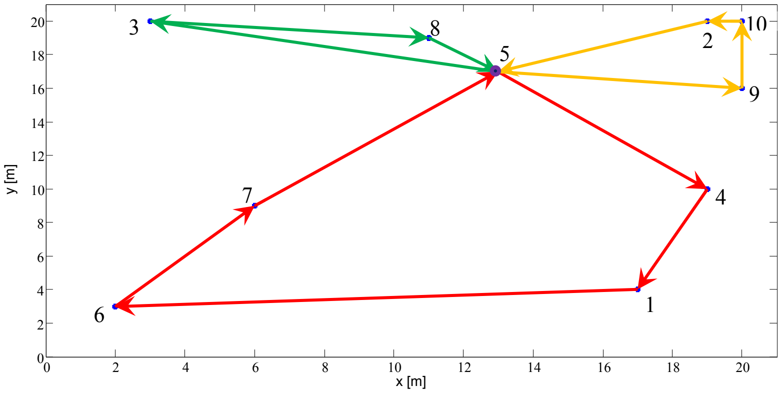


Figure 2.6: The Optimal mTSP for the case where the starting depot is not prespecified and the return trip is not allowed ($n = 10, m = 3$).

Table 2.2: Comparison between the results of scenarios 1-3 for the case where the return trip is not allowed.

	Scenario 1	Scenario 2 (worst-case scenario)	Scenario 3 (non-prespecified starting depot)
Starting depot	node 1 (prespecified)	node 6 (prespecified)	node 5 (calculated)
Total travel distance per tour	113.6117m	130.4827m	88.5289m
Computation time	0.2s	0.31s	1.19s
Reduction in travel distance per tour	25.0828m (22.08%)	41.9538m (32.15%)	–

Fig. 2.7 depicts the average reduction in the total travel distance per tour over 10 different configurations with different numbers of viewpoints. The results

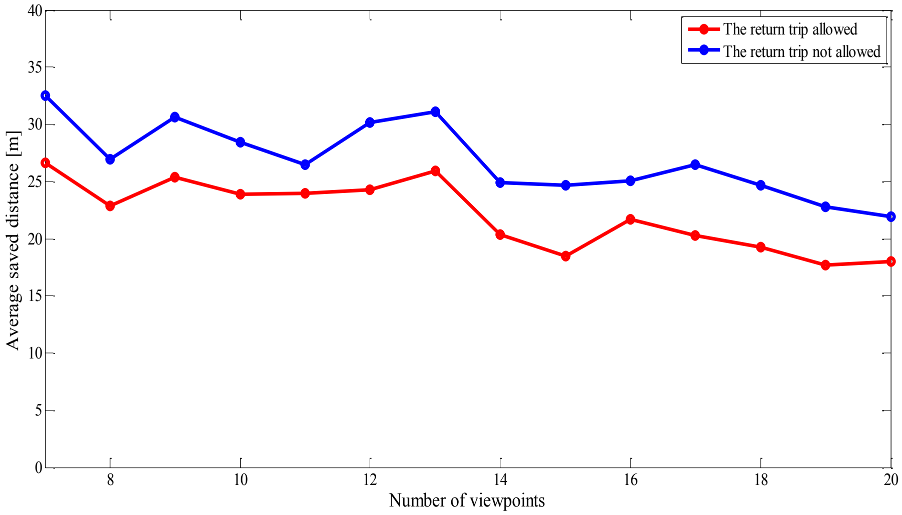


Figure 2.7: The average total travel distance reduction between the best and worst choice of the starting depot, and both cases of the return trip allowed (red curve) and not allowed (blue curve).

are given for both cases of the return trip allowed and not allowed, and are obtained by randomly generating the location of the viewpoints in a 20m by 20m area for different values of n between 7 and 20 (the number of robots in all simulations is $m = 3$). The figure demonstrates the efficacy of the proposed non-prespecified starting depot method compared to existing prespecified starting depot techniques, as considerable reduction in travel distance could be achieved by using the proposed method. In particular, compared to the worst-case scenario (in terms of the starting depots), the resultant average reduction in travel distance per tour will be 25m when the return trip is allowed, and 35m per tour when the return trip is not allowed (note that a tour may be traveled several times in real world applications leading to sizable reduction in overall travel distance, as mentioned earlier).

Fig. 2.8 shows a comparison between the greedy algorithm, where the optimization framework is repeated at each viewpoint, and the result obtained by the new formulation, where the optimization framework is only used once to result in the optimal solution. The average computation time over 10 different random configurations for $n = 3, 4, \ldots, 13$ viewpoints is calculated and repeated for $m = 1, 2, \ldots, n-1$

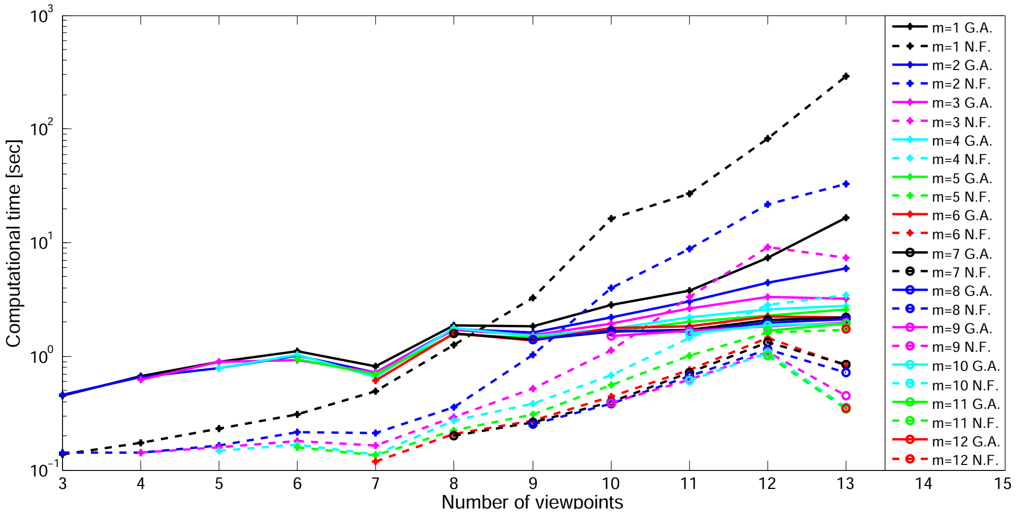


Figure 2.8: A comparison between the greedy algorithm and the result obtained by the new formulation.

robots. The comparison can be summarized as follows:

i) For a given number of viewpoints, the computational time of the greedy algorithm becomes higher than that of the new formulation as the number of the robots increases.

ii) The computational time is highly dependant on the configuration of the viewpoints in the patroled area.

iii) For a given number of viewpoints, as the number of the robots increases, the computational time decreases. Similarly, Fig. 2.9 shows a comparison between the same two proposed approaches, where the average computation time of 10 different random configurations at each possible number of the robots (from $m = 1$ to $m = n-1$) is calculated and repeated this time for all values of number of viewpoints (from $n = 3$ to $n = 13$). The comparison result can be stated as follows:

i) For larger number of robots (m), the computational time of the greedy algorithm is higher than that of the new formulation as the number of the viewpoints (n) increases.

ii) The computational time is dramatically related to the configuration of the viewpoints in the area to be patrolled.

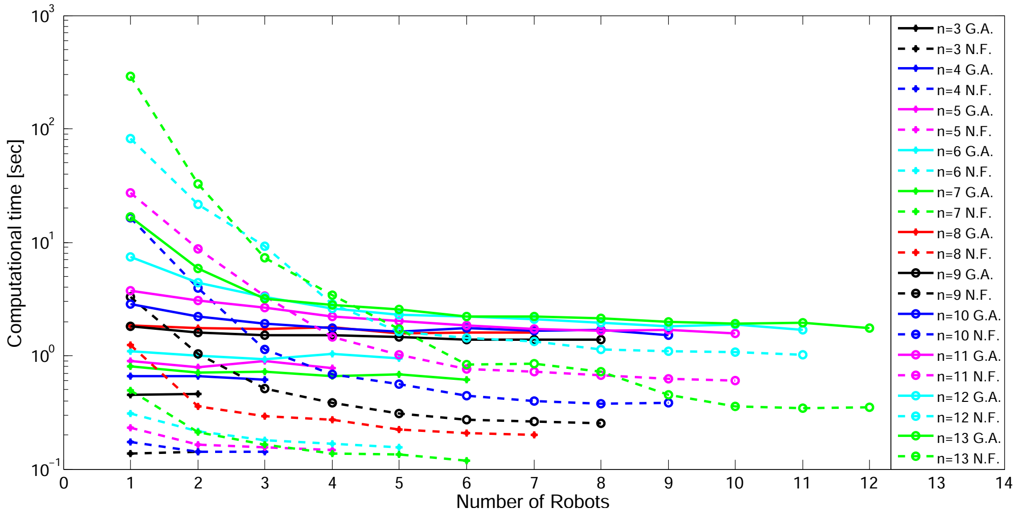


Figure 2.9: A comparison between the greedy algorithm and the result obtained by the new formulation.

iii) For a given number of robots (m), as the number of the viewpoints (n) increases, the computational time increases. Fig. 2.10 provides a 3-D graph to demonstrate the dependancy of the computational time on both the number of the robots and the number of the viewpoints at the same time.

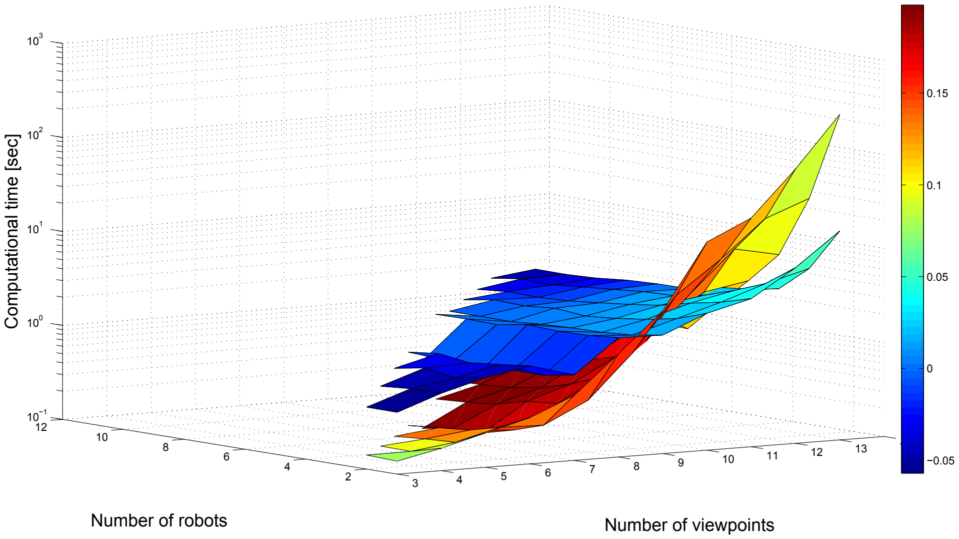


Figure 2.10: Comparison between the new greedy algorithm and the new formulation approaches.

In the next chapter, the MmTSP-based patrolling operation, as a generalization of the mTSP-based one, is investigated to the case where more than one depot exists and there are a number of robots at each depot [46]. The assumptions that exist in the literature for having a prespecified set of depots, a prespesified set costumer nodes and an initial prespecified number of robots at each depot are relaxed. The problem is optimally solved for the case of non-prespecified as well as unknwown number of depots and robots.

Chapter 3

Generalized Formulations for Minimum-Distance Trajectory in Patrolling Problems

This chapter introduces the MmTSP problem, as a generalization of the mTSP to the case where more than one depot exists and there are a number of robots at each depot [46]. Most of the existing results tackle the MmTSP by using prespecified starting depots [46, 47]. Three main assumptions are typically used: (i) a set of starting depots for the robots is prespecified; (ii) a set of customer nodes that contains the rest of all the other nodes which are to be visited by the robots throughout the process is prespecified; (iii) The number of robots at each starting depot is also prespecified. These assumptions have a significant impact on the optimal result.

These assumptions distinguish the present work from the existing literature. These assumptions will be relaxed in order to introduce general formulations which extend the results developed in Chapter 2. Three general formulations are presented for minimum-distance trajectory optimization in patrolling problems. In the first formulation, it is assumed that the starting depots are not prespecified. In the second and the third, the optimal number of starting depots and patrolling robots are unknown. It is assumed that the starting depots are among the nodes (viewpoints) to be visited, and again as assumed in Chapter 2 the robots have the same dynamics as the TWMRs with the same previous assumptions on their physical constraints. Using the proposed new formulations for the non-prespecified starting depots and robots or their unknown numbers, the efficiency of the strategy is evaluated and compared to the well-known prespecified starting MmTSP approach. It is confirmed by simulation that under the proposed method robots travel smaller distance and complete the patrolling mission more rapidly.

The rest of this chapter is organized as follows. The problem statement for the MmTSP is introduced in Section 3.1, Section 3.2 presents the proposed framework and the new formulations. The simulation results are provided in Section 3.3.

3.1 Problem Statement

Consider a complete undirected graph $G(V, E)$, where $V = \{v_1, \ldots, v_n\}$ denotes a set of n viewpoints through which m robots, $m < n$, will perform a patrolling operation for monitoring an area. The set of prespecified depots is denoted by $D = \{v_1, \ldots, v_d\}$, from which a prespecified number of m robots will start their trajectories. Furthermore, $V' = \{v_{d+1}, v_{d+2}, \ldots, v_n\}$ is the set of customer nodes to be visited, m_k denotes the prespecified number of robots initially located at depot v_k, and E is the set of all edges connecting any two nodes, and represent the motion trajectories to be tracked by the robots. Let $[c_{ij}]$ denote the cost matrix (or weight matrix), where the element c_{ij} is between viewpoints v_i and v_j, where $c_{ij} = c_{ji}$, $\forall (i, j) \in E$. Note that the distance between two connected nodes satisfies the triangular inequality, i.e., for any three nodes i, j, k, $c_{ij} + c_{jk} \geq c_{ik}$.

The traditional TSP is formulated using a weighted graph representation introduced above, with the assumption that the starting depots and the number of robots at each depot are prespecified. In the present work, however, three problems are investigated, where only the set of n viewpoints V, the number of depots d, and the total number of robots m are assumed to be given in the first problem. In other words, there is no prespecified set of depots or prespecified set of customer nodes or even prespecified number of robots at each depot. In the second problem, only the set of n viewpoints V, is assumed to be given with no assumption on the number of starting depots and robots. The minimum distance trajectories are required to be computed such that each robot follows exactly one trajectory connecting a subset of the viewpoints starting from some depots which are not prespecified, and returning back to them. There are two cases of interest: (i) when a return trip is allowed, i.e., the robot can visit only one viewpoint before returning back to its starting depot, and (ii) when a return trip is not allowed. The restriction in the second case applies to applications such as pickup and delivery, where visiting only one node is not allowed. It will be shown later that when it is possible to assign the starting depots to any existing node, the minimum-distance trajectories highly depend on the starting depots' positions. This will have a significant impact on the travel distance, and consequently, can increase the life time of the patrolling operation.

3.2 Proposed Framework

Problem 1: Multidepot multiple Traveling Salesmen Problem (MmTSP) with non-prespecified starting depots. With the aforementioned graph representation $G(V, E)$ and cost matrix $[c_{ij}]$, the Multidepot multiple Traveling Salesmen Problem can be formulated in an optimization framework for the case when the return trip is allowed. To this end, let a link connecting two arbitrary nodes v_i, v_j

on the trajectory of depot v_k be represented by a binary variable x_{ijk} which will be equal 1 if the trajectory is optimal and 0 otherwise. The cost function to be minimized can be represented as the overall sum of the product of each element in $[c_{ij}]$ and its corresponding variable x_{ijk}, and can be formulated as:

$$\min_{x_{ijk}}\{\sum_{k\in D}\sum_{j\in V'} c_{kj}x_{kjk} + c_{jk}x_{jkk} + \sum_{k\in D}\sum_{i\in V'}\sum_{\substack{j\in V' \\ j\neq i}} c_{ij}x_{ijk}\} \tag{3.1}$$

The number of departures from any starting depot, which by assumption is equal to the number of arrivals to the depot, is denoted by m_k (note that any starting depot belongs to the set D). This specifies two of the constraints of the optimization problem, corresponding to the number of robots initially located at each depot, as given below:

$$\sum_{j\in V'} x_{kjk} = m_k, \qquad k\in D \tag{3.2}$$

$$\sum_{j\in V'} x_{jkk} = m_k, \qquad k\in D \tag{3.3}$$

In addition, the number of departures from any node other than the starting depots, which by assumption is equal to the number of arrivals to that node, is one, i.e.:

$$\sum_{k\in D} x_{kjk} + \sum_{k\in D}\sum_{i\in V'} x_{ijk} = 1, \qquad j\in V^{'} \tag{3.4}$$

$$\sum_{k\in D} x_{jkk} + \sum_{k\in D}\sum_{i\in V'} x_{jik} = 1, \qquad j\in V^{'} \tag{3.5}$$

To ensure that any optimal route is not shared among more than one depot, the route continuity constraint is imposed as:

$$x_{kjk} + \sum_{i \in V'} x_{ijk} - x_{jkk} - \sum_{i \in V'} x_{jik} = 0, \qquad k \in D,\ j \in V' \tag{3.6}$$

On the other hand, the SEC can be presented by the following inequality:

$$u_i - u_j + (n-m) \sum_{k \in D} x_{ijk} \leq n - m - 1, \qquad i, j \in V' \tag{3.7}$$

where u_i and u_j are the two artificial integer variables that prevent the formation of such sub-tours among nodes not belonging to D, thus this optimization problem is considered an MIP problem. Finally, the binary variable x_{ij} is equal to one if the edge (i, j) is optimal, and is zero otherwise.

For the second case, when the return trip is not allowed, at least two nodes other than the starting depot are required in each trajectory. A formulation similar to the previous case is used here along with the new inequality given below:

$$\sum_{k \in D} x_{kjk} + \sum_{k \in D} x_{jkk} \leq 1, \qquad j \in V' \tag{3.8}$$

which will be used to fulfill the new requirement of the problem.

Remark 3.1. *A similar formulation is presented in [46] with a set of constraints on the maximum and minimum number of nodes that a robot has to visit. However, the relations introduced here do not include such restrictions, as the proposed formulation represents a different optimization framework.*

It is assumed that for a given set of viewpoints, only the total number of depots as well as the total number of robots are given, without pre-defining a certain set of starting depots or a specific number of robots at any starting depot, initially. A set of n auxiliary binary variables ω_k are introduced, each of which corresponds to one of the possible choices of depots among the n viewpoints in such a way that ω_k will be equal to zero if v_k is an optimal starting depot, and one otherwise.

The cost function to be minimized can be reformulated as:

$$\min_{x_{ijk}} \sum_{k\in V} \sum_{i\in V} \sum_{\substack{j\in V \\ j\neq i}} c_{ij} x_{ijk} \tag{3.9}$$

The number of departures from and arrivals to any starting depot is m, which is the total number of robots, i.e.:

$$\sum_{k\in V} \sum_{\substack{j\in V \\ j\neq k}} x_{kjk} = m \tag{3.10}$$

$$\sum_{k\in V} \sum_{\substack{j\in V \\ j\neq k}} x_{jkk} = m \tag{3.11}$$

(note that the set of the starting depots is not prespecified in this case). The total number of departures from and arrivals to any node other than the starting depots equals $(n-d)$, i.e., only one departure and one arrival for any node other than the starting depots. This is expressed by:

$$\sum_{k\in V} \sum_{\substack{j\in V \\ j\neq k}} x_{kjk} + \sum_{k\in V} \sum_{\substack{i\in V \\ i\neq k}} \sum_{\substack{j\in V \\ j\neq i,k}} x_{ijk} = n - d \tag{3.12}$$

$$\sum_{k\in V} \sum_{\substack{j\in V \\ j\neq k}} x_{jkk} + \sum_{k\in V} \sum_{\substack{i\in V \\ i\neq k}} \sum_{\substack{j\in V \\ j\neq i,k}} x_{jik} = n - d \tag{3.13}$$

respectively. To ensure that any optimal tour is not shared among more than one depot, the route continuity constraint is imposed as follows:

$$x_{kjk} + \sum_{\substack{i\in V \\ i\neq j,k}} x_{ijk} - x_{jkk} - \sum_{\substack{i\in V \\ i\neq j,k}} x_{jik} = 0, \qquad k, j \in V, j \neq k \tag{3.14}$$

Moreover, a new constraint on the sum of the new auxiliary variables is given by:

$$\sum_{k \in V} \omega_k = n - d \tag{3.15}$$

which means that the minimum and maximum number of robots that could be placed at an optimal depot are given by:

$$1 \leq \sum_{j \in V} x_{kjk} + (m - (d-1))\,\omega_k \leq m - (d-1) \tag{3.16}$$

$$1 \leq \sum_{j \in V} x_{jkk} + (m - (d-1))\,\omega_k \leq m - (d-1) \tag{3.17}$$

respectively, for any $k \in V, j \neq k$. The total number of arrivals to and departures from any node v_j belonging to a trajectory which starts from depot v_k is one if v_j is not a depot and is zero if v_j is another depot as represented in the following inequalities:

$$0 \leq \sum_{\substack{k \in V \\ k \neq j}} x_{kjk} + \sum_{\substack{k \in V \\ k \neq j}} \sum_{\substack{i \in V \\ i \neq j,k}} x_{ijk} \leq 1, \qquad j \in V \tag{3.18}$$

$$0 \leq \sum_{\substack{k \in V \\ k \neq j}} x_{jkk} + \sum_{\substack{k \in V \\ k \neq j}} \sum_{\substack{i \in V \\ i \neq j,k}} x_{jik} \leq 1, \qquad j \in V \tag{3.19}$$

The following inequality is a constraint that prevents any depot from being part of another depot's trajectory:

$$-4 \leq \sum_{\substack{k \in V \\ k \neq i,j}} x_{iji} + x_{jii} + x_{kji} + x_{jki} + x_{jij} + x_{ijj} + x_{kij} + x_{ikj} - 2(\omega_i + \omega_j) \leq 0, \qquad i, j \in V, i \neq j \tag{3.20}$$

The SEC can be presented by:

$$u_i - u_j + (n-m)\sum_{\substack{k\in V\\k\neq i,j}} x_{ijk} + (n-2(d-1))\omega_j \leq n-m-1+(n-2(d-1)), \qquad i,j\in V, i\neq j \tag{3.21}$$

where x_{ijk}, $\omega_k \in \{0,1\}$. The above inequality is different from inequality (3.7) as a new expression is added in (3.21), where v_j will be an optimal depot if $\omega_j = 0$, for which the set of inequalities representing an optimal tour will be valid. The maximum number of tour links that could exist in any optimal tour is $(n-2(d-1))$. The basic constraint here is that each node can take only one arrival and one departure. This condition is described by the following inequalities:

$$0 \leq \sum_{\substack{i\in V\\i\neq j}} x_{ijk} + \omega_k \leq 1, \qquad j,k\in V, j\neq k \tag{3.22}$$

$$0 \leq \sum_{\substack{i\in V\\i\neq j}} x_{jik} + \omega_k \leq 1, \qquad j,k\in V, j\neq k \tag{3.23}$$

Note that the binary variable x_{ijk} is one if the edge (i,j) belonging to the trajectory of starting depot v_k is optimal, and is zero otherwise. The auxiliary binary value ω_k, on the other hand, is equal to zero if v_k is the optimal starting depot, and is one otherwise. The inequality describing that the return trip is not allowed is expressed by:

$$\sum_{\substack{k\in V\\k\neq j}} x_{kjk} + \sum_{\substack{k\in V\\k\neq j}} x_{jkk} \leq 1, \qquad j\in V \tag{3.24}$$

The previous formulation (Equations 3.9 - 3.24) was introduced in [103] where both the depots and the robots are non-prespecified.

Problem 2: Unknown number of depots and robots in a Traveling Salesman Problem (UTSP). The (MmTSP) discussed above was used to obtain the minimum-distance optimal trajectories, given the number of starting depots and the number of robots, without prespecifying their locations. However, to further reduce the travel distance, one can relax the assumption on the prescribed number of robots and starting depots, making them new minimization variables to obtain by solving the underlying optimization problem. This will lead to the optimal number of robots and starting depots, in addition to their specific locations, for the minimum-distance trajectories for the patrolling operation. Thus, the main difference between the previously introduced MmTSP and Problem 2 defined here is that in this new and more challenging framework only the set of viewpoints V is known. The problem is formulated in the sequel for both cases of the return trip allowed and not allowed, analogously to Problem 1.

Problem 2 can now be formally defined as an mTSP or MmTSP with a given number and configuration of viewpoints, and unknown number of starting depots and robots as well as nonprespecified set of nodes considered as the starting depots. Given the generality of the problem, its solution will outperform that of any variant of TSP in terms of travel distance.

Theorem 3.1. *In the optimal solution of Problem 2, the number of robots is the same as the number of starting depots, i.e., each starting depot has exactly one robot, initially.*

Proof. (i) *Single depot case.* This is a rather trivial case which can be demonstrated through a simple example, with no loss of generality. To this end, consider the configuration in Fig. 3.1, with one robot, with one starting depot and four viewpoints. Assume that the total travel distance $i \to j \to k \to l \to i$ is P, i.e., $c_{ij} + c_{jk} + c_{kl} + c_{li} = P$. Now, if two robots start from depot i and move on two separate trajectories both starting and ending at depot i, due to the triangular inequality, the travel distance will be longer. For the trajectories shown in Fig. 3.2, it can be shown that: $P_1 + P_2 = P + c_{ji} + c_{ik} - c_{jk}$ where P_1 and P_2 denote the length of the trajectories of the two robots. This justification can be generalized to a graph with any number of viewpoints and robots, as long as there is only one starting depot. This implies that in the optimal setting in this case $m = d = 1$.

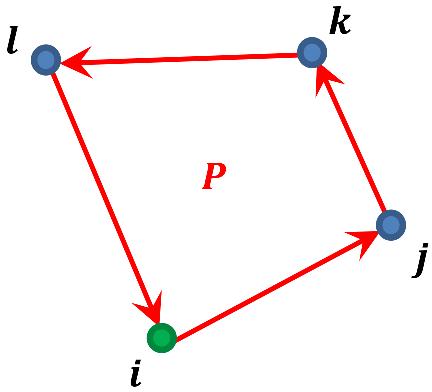


Figure 3.1: A simple example of TSP with 1 depot and 1 robot

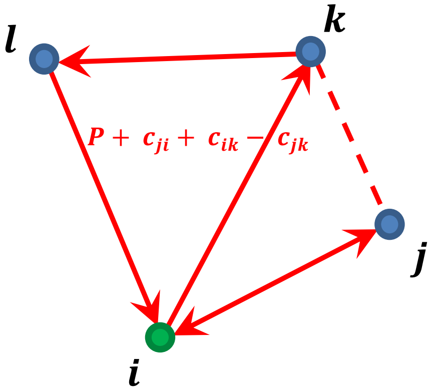


Figure 3.2: The same configuration of Fig. 3.1 but with two robots

(ii) *Multidepot Case.* Using 2 robots for 2 non-prespecified depots using our previous formulation, the non-prespecified MmTSP will result in the optimal minimum distance as shown in Fig. 3.3. The total optimal distance in this case $= P+Q$. Increasing the number of robots can occur in three scenarios:

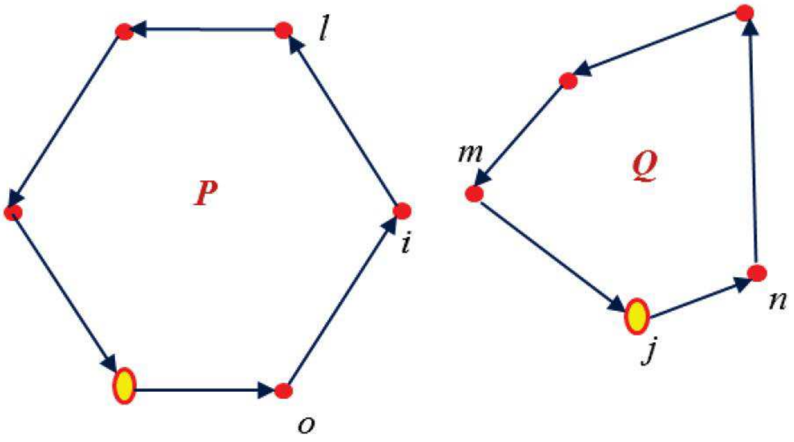


Figure 3.3: Optimal TSP using 2 depots and 2 robots

Scenario 1. A robot starts from a depot in the first route and moves towards another node inside the same route. This is similar to the single depot case which was previously proved to have more travel distance as shown in Fig. 3.4.

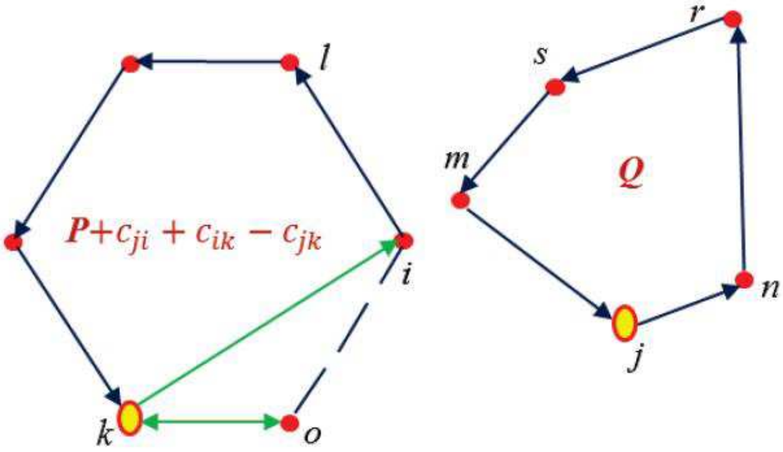


Figure 3.4: Optimal TSP using 2 depots and 3 robots

Scenario 2. A robot starts from a depot in the second route and moves towards another node inside the same route. This is again similar to the single depot case which was previously proved to have more travel distance as shown in Fig. 3.5.

Scenario 3. A robot starts from a depot in one of the two routes and moves towards another node inside the opposite route, respectively, as shown in Fig. 3.6. Searching for the closest nodes in both routes that can be linked in the new optimal

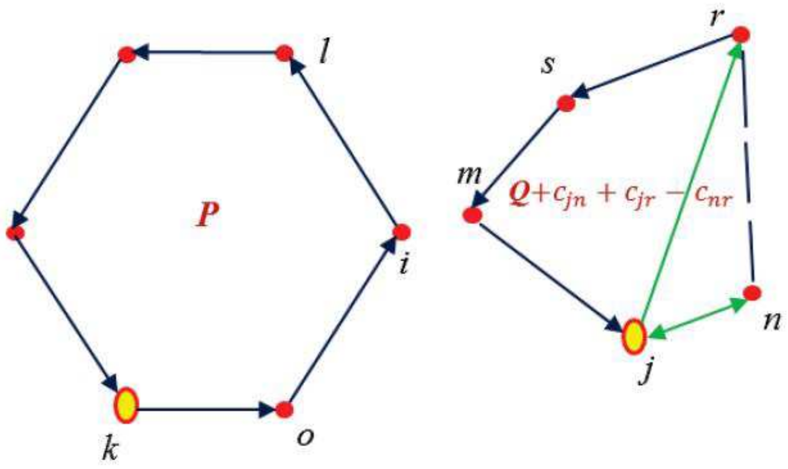


Figure 3.5: Optimal TSP using 2 depots and 3 robots

solution, without loss of generality, here they are i and m. The total optimal distance in this case $= P + Q + 2c_{im} + c_{sj} - c_{sm} - c_{mj}$. Comparing the two distances before and after adding a third robot, if $2c_{im} + c_{sj} > c_{sm} + c_{mj}$, thus the new distance is more than the old one. If $2c_{im} + c_{sj} < c_{sm} + c_{mj}$, then the new distance is less than the old one. If $2c_{im} + c_{sj} = c_{sm} + c_{mj}$, then the new distance is equal to the old one.

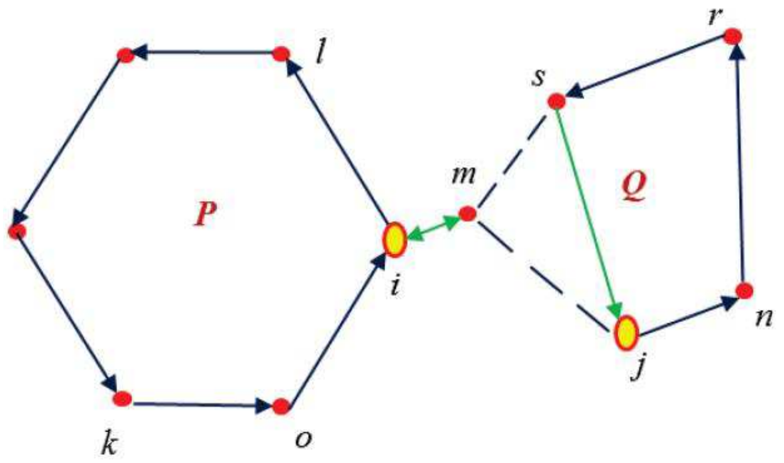


Figure 3.6: Optimal TSP using 2 depots and 3 robots

As node m was not included in the first route in the first solution (2 depots and 2 robots), thus:

$$c_{im} + c_{ml} > c_{il} \tag{3.25}$$

$$P - c_{il} + c_{im} + c_{ml} + Q - c_{sm} - c_{mj} + c_{sj} > P + Q \tag{3.26}$$

$$c_{im} + c_{ml} + c_{sj} > c_{il} + c_{sm} + c_{mj} \tag{3.27}$$

Then there could be two assumptions, either $c_{ml} = c_{im} + \epsilon$ or $c_{ml} = c_{im} - \epsilon$, where ϵ is any positive real number.

First assumption:

$$c_{ml} = c_{im} + \epsilon \tag{3.28}$$

$$2c_{im} + \epsilon + c_{sj} > c_{il} + c_{sm} + c_{mj} \tag{3.29}$$

But

$$c_{im} + c_{il} > c_{ml} \tag{3.30}$$

$$c_{im} + c_{il} > c_{ml} + \epsilon \tag{3.31}$$

$$c_{il} > \epsilon \tag{3.32}$$

Thus

$$2c_{im} + c_{sj} > c_{il} - \epsilon + c_{sm} + c_{mj} \tag{3.33}$$

$$2c_{im} + c_{sj} > c_{sm} + c_{mj} \tag{3.34}$$

Thus, the new distance is more than the old one.

Second assumption:

$$c_{ml} = c_{im} - \epsilon \tag{3.35}$$

$$2c_{im} - \epsilon + c_{sj} > c_{il} + c_{sm} + c_{mj} \tag{3.36}$$

But

$$c_{im} + c_{il} > c_{ml} \tag{3.37}$$

$$c_{im} + c_{il} > c_{ml} - \epsilon \tag{3.38}$$

$$\epsilon > c_{il} \tag{3.39}$$

Thus

$$2c_{im} + c_{sj} > c_{il} + \epsilon + c_{sm} + c_{mj} \tag{3.40}$$

$$2c_{im} + c_{sj} > c_{sm} + c_{mj} \tag{3.41}$$

Thus, the new distance is more than the old one.

From the previous possible two assumptions, the minimum travel distance will occur when $m = d = 2$, and with the same procedures for larger number of depots, the result will be the same, i.e., a depot should have only one starting robot for optimal minimum travel distance.

(iii) *Single and Multiple robots.* For one robot, the only choice is to have one depot (one of the previous mentioned optimal cases). For m robots, the possible choices of the number of depots are $\{1, 2, \ldots, m\}$. Without loss of generality, decreasing the number of depots from the same number of robots will always increase the total travel distance as shown in Figs. 3.7, 3.8. The total travel distance for 2 robots and 2 depots $= P + Q$. The total travel distance for 2 robots and 1 depot $= P + Q - c_{ij} + c_{ik} + c_{jk} > P + Q$. This will always happen for decreasing the number of depots. Thus the minimum travel distance will occur when $d = m$. □

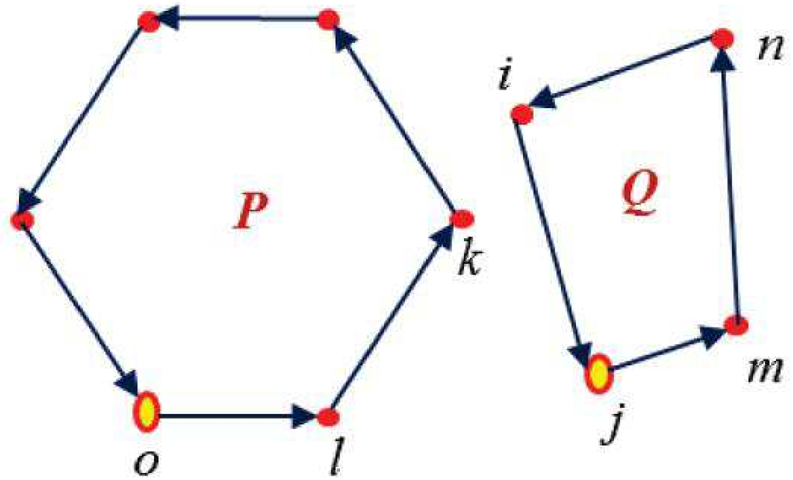


Figure 3.7: Optimal TSP using 2 depots and 2 robots

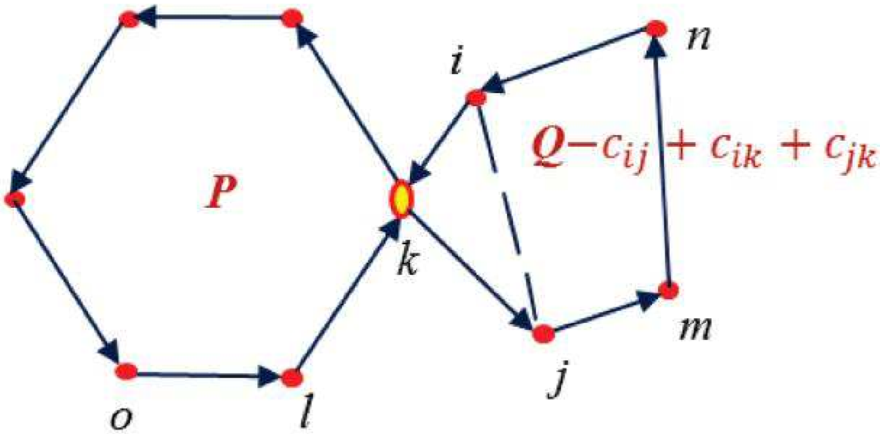


Figure 3.8: Optimal TSP using 1 depot and 2 robots

The new proposed formulation depends only on a given number of viewpoints, with the previous theorem result , i.e., the number of robots should be equal to the number of depots, whereas the number of depots ranges between 1 and $n/2$. A set of n auxiliary binary variables ω_k are introduced, where each possible optimal depot has its corresponding ω_k such that ω_k equals 0 if the corresponding node is an optimal depot, and ω_k equals 1 otherwise. This formulation runs with $O(n^3)$ binary variables, and $O(n^2)$ constraints. The framework of this formulation can be stated as follows. The cost function to be minimized can be formulated as in the following equation:

$$\min_{x_{ijk}} \sum_{k\in V} \sum_{i\in V} \sum_{\substack{j\in V\\ j\neq i}} c_{ij} x_{ijk} \tag{3.42}$$

The number of departures from and arrivals to all the starting depots which are here unknown equals the optimal number of the starting depots as proved in the previous theorem are given by:

$$\sum_{k\in V} \sum_{\substack{j\in V\\ j\neq k}} x_{kjk} + \sum_{k\in V} \omega_k = n \tag{3.43}$$

$$\sum_{k\in V} \sum_{\substack{j\in V\\ j\neq k}} x_{jkk} + \sum_{k\in V} \omega_k = n \tag{3.44}$$

respectively. The total number of departures from and arrivals to all other nodes not including the depots which are here again unknown equals ($n-$the optimal number of depots), i.e., equals the sum of the auxiliary binary variables. This is expressed by:

$$\sum_{k\in V} \sum_{\substack{j\in V\\ j\neq k}} x_{kjk} + \sum_{k\in V} \sum_{\substack{i\in V\\ i\neq k}} \sum_{\substack{j\in V\\ j\neq i,k}} x_{ijk} = \sum_{k\in V} \omega_k \tag{3.45}$$

$$\sum_{k\in V} \sum_{\substack{j\in V\\ j\neq k}} x_{jkk} + \sum_{k\in V} \sum_{\substack{i\in V\\ i\neq k}} \sum_{\substack{j\in V\\ j\neq i,k}} x_{jik} = \sum_{k\in V} \omega_k \tag{3.46}$$

To ensure that any optimal route is not shared among more than one depot, the following equation presents the route continuity constraint:

$$x_{kjk} + \sum_{\substack{i\in V\\ i\neq j,k}} x_{ijk} - x_{jkk} - \sum_{\substack{i\in V\\ i\neq j,k}} x_{jik} = 0, \qquad j,k \in V, j \neq k \tag{3.47}$$

The next equation represents a new constraint on the sum of the new auxiliary variables, where there sum ranges between $n/2$ and $(n-1)$.

$$n/2 \leq \sum_{k \in V} \omega_k \leq n-1 \tag{3.48}$$

The following two equations represent the direct result of the proved theorem, where each depot has only one arrival and one departure.

$$\sum_{\substack{j \in V \\ j \neq k}} x_{kjk} + \omega_k = 1, \qquad k \in V \tag{3.49}$$

$$\sum_{\substack{j \in V \\ j \neq k}} x_{jkk} + \omega_k = 1, \qquad k \in V \tag{3.50}$$

The total arrivals to and departure from any node v_j from any starting depot v_k or from any other node v_i belonging to this depot trajectory is 1 in case of v_j is not a depot and equals 0 in case v_j is another optimal depot or this link is not an optimal link as represented in the following inequalities:

$$0 \leq \sum_{\substack{k \in V \\ k \neq j}} x_{kjk} + \sum_{\substack{k \in V \\ k \neq j}} \sum_{\substack{i \in V \\ i \neq j,k}} x_{ijk} \leq 1, \qquad j \in V \tag{3.51}$$

$$0 \leq \sum_{\substack{k \in V \\ k \neq j}} x_{jkk} + \sum_{\substack{k \in V \\ k \neq j}} \sum_{\substack{i \in V \\ i \neq j,k}} x_{jik} \leq 1, \qquad j \in V \tag{3.52}$$

respectively. The next inequality is a constraint that prevents any depot to be included in another depot trajectory.

$$-4 \leq \sum_{\substack{k \in V \\ k \neq i,j}} x_{iji} + x_{jii} + x_{kji} + x_{jki} + x_{jij} + x_{ijj} + x_{kij} + x_{ikj} - 2(\omega_i + \omega_j) \leq 0, \qquad i,j \in V, i \neq j \tag{3.53}$$

The SEC can be presented by:

$$u_i - u_j + n \sum_{\substack{k \in V \\ k \neq i,j}} x_{ijk} + n\, \omega_j \leq n - 1 + n, \qquad i,j \in V, i \neq j \tag{3.54}$$

The main basic constraint in such problems is that each node has only one arrival and one departure through one optimal starting depot trajectory, this is presented as follows:

$$0 \leq \sum_{\substack{i \in V \\ i \neq j}} x_{ijk} + \omega_k \leq 1, \qquad j,k \in V, j \neq k \tag{3.55}$$

$$0 \leq \sum_{\substack{i \in V \\ i \neq j}} x_{jik} + \omega_k \leq 1, \qquad j,k \in V, j \neq k \tag{3.56}$$

respectively. The definitions of the variable x_{ijk} as well as the auxiliary binary variable ω_k are as presented in Problem 1. The following inequality ensures the case of the return trip not allowed, i.e., visiting only one node is not allowed.

$$\sum_{k \in V} x_{kjk} + \sum_{k \in V} x_{jkk} \leq 1, \qquad j \in V \tag{3.57}$$

Equations 3.42 - 3.57 compose the new TSP-based formulation for the unknown depots and robots.

A simpler proposed formulation non-TSP-based that gives the same optimal results as the previous proposed formulation is introduced without having all these constraints, and thus without using a notation for a depot node with much less computation time. It states that the number of either the arrivals or the departures from any node is at least 1 and at most $n-1$, in addition to the constraint of the route continuity and in case of the return trip not allowed case, the corresponding constraint is also added. This formulation runs with $O(n^2)$ binary variables, and $O(n^2)$ constraints.

$$\min_{x_{ij}} \sum_{i \in V} \sum_{\substack{j \in V \\ j \neq i}} c_{ij} x_{ij} \tag{3.58}$$

$$1 \leq \sum_{\substack{j \in V \\ j \neq i}} x_{ij} \leq n-1, \qquad i \in V \tag{3.59}$$

$$1 \leq \sum_{\substack{j \in V \\ j \neq i}} x_{ji} \leq n-1, \qquad i \in V \tag{3.60}$$

$$\sum_{\substack{i \in V \\ i \neq j}} x_{ij} - \sum_{\substack{i \in V \\ i \neq j}} x_{ji} = 0, \qquad j \in V \tag{3.61}$$

$$x_{ij} + x_{ji} \leq 1, \qquad i, j \in V,\ i \neq j \tag{3.62}$$

Equations 3.58 - 3.62 compose the new non-TSP-based formulation for the unknown depots and robots. Problems 1 and 2 are both expressed in MIP framework with linear constraints. Thus, they are convex and always have a feasible optimal solution.

3.3 Simulation Results

Consider the patrolling problem for a field of size 20m by 20m, where a set of $n = 10$ nodes (viewpoints) are to be visited. Let the number of robots be $m = 4$, with the number of depots $d = 2$. Using the formulation for the prespecified starting set of depots as well as the proposed formulation for the non-prespecified starting depots, it is desired to find the minimum-distance trajectories. MATLAB was used with the Gurobi Optimizer 6.0 [102] optimization software to obtain all the results using Intel Core i7-3537U @ 2.00GHz processor with 8 GB RAM.

Scenario 1. The optimal results are obtained for the case of prespecified depots with the return trip allowed, where the starting depots are considered to be nodes 1 and 2, and the four robots are distributed in different initial locations with all possible combinations: $(m_1, m_2) = (1, 3)$ in Fig. 3.9, $(m_1, m_2) = (2, 2)$ in Fig. 3.10, and $(m_1, m_2) = (3, 1)$ in Fig. 3.11 with the total travel distance ranges between about 80m and 97m.

Scenario 2. In this scenario, the results are computed for the case where the starting depots are considered to be nodes 6 and 7, which is, in fact, the worst-case scenario for the present configuration as far as the minimum travel distance is concerned (with two starting depots). The results for $(m_6, m_7) = (1, 3)$, $(m_6, m_7) = (2, 2)$ and $(m_6, m_7) = (3, 1)$ are depicted in Figs. 3.12-3.14, analogously to the previous figures with the total travel distance ranges between about 116m and 140m.

Scenario 3. In this scenario, it is assumed that unlike the previous two scenarios, the starting depots are not prespecified, and that, like the previous scenarios, the return trip is allowed. Using the proposed formulation, the optimal trajectory depicted in Fig. 3.15 is obtained. The total travel distance in this case is about 74m.

Table 3.1 provides the comparison between the results of scenario 1 and Scenario 3. Similarly, Table 3.2 presents the comparison between the results of scenario 2 (worst-case scenario) and Scenario 3. This shows the significant saving in travel

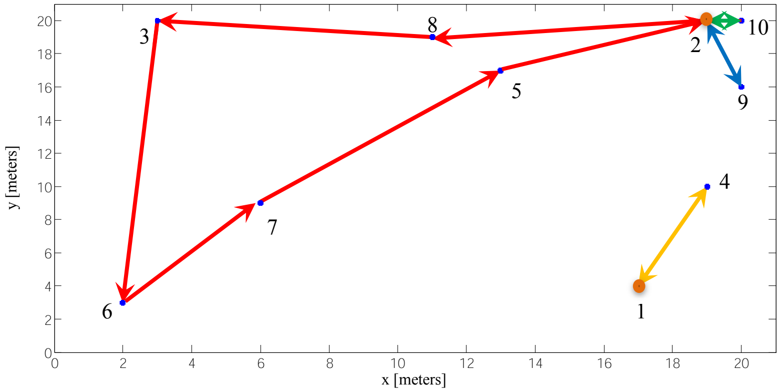


Figure 3.9: The Optimal MmTSP for the case where nodes 1 and 2 are the starting depots, and the return trip is allowed $(n = 10, m = 4, d = 2, (m_1, m_2) = (1, 3))$.

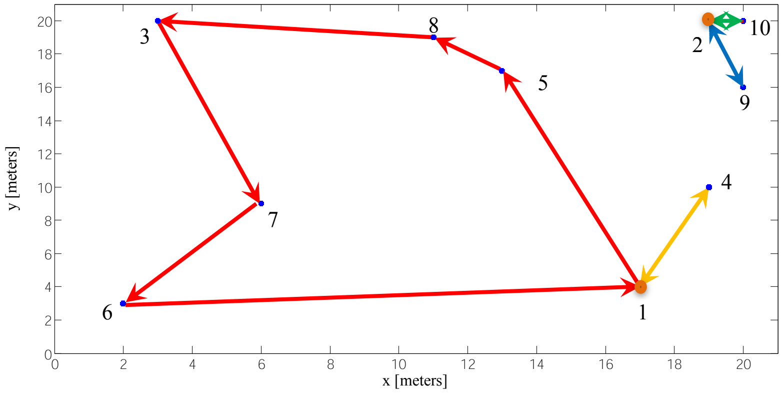


Figure 3.10: The Optimal MmTSP for the case where nodes 1 and 2 are the starting depots, and the return trip is allowed $(n = 10, m = 4, d = 2, (m_1, m_2) = (2, 2))$.

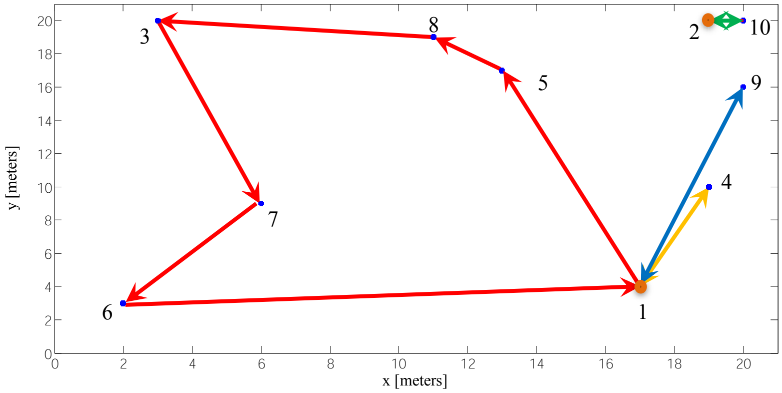


Figure 3.11: The Optimal MmTSP for the case where nodes 1 and 2 are the starting depots, and the return trip is allowed $(n = 10, m = 4, d = 2, (m_1, m_2) = (3, 1))$.

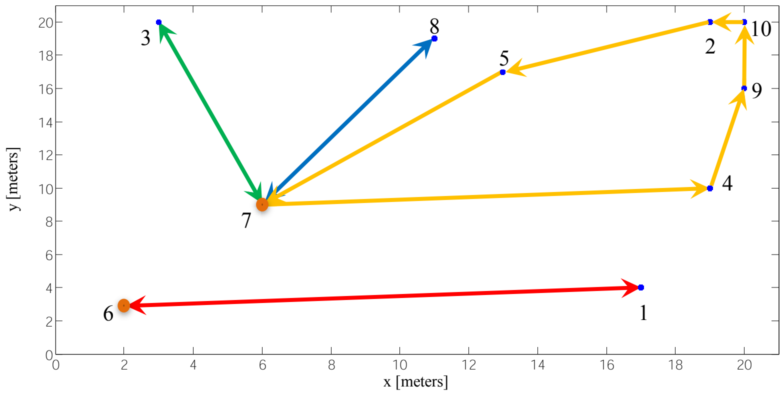


Figure 3.12: The Optimal MmTSP for the case where nodes 6 and 7 are the starting depots (worst-case scenario), and the return trip is allowed ($n = 10, m = 4, d = 2, (m_6, m_7) = (1, 3)$).

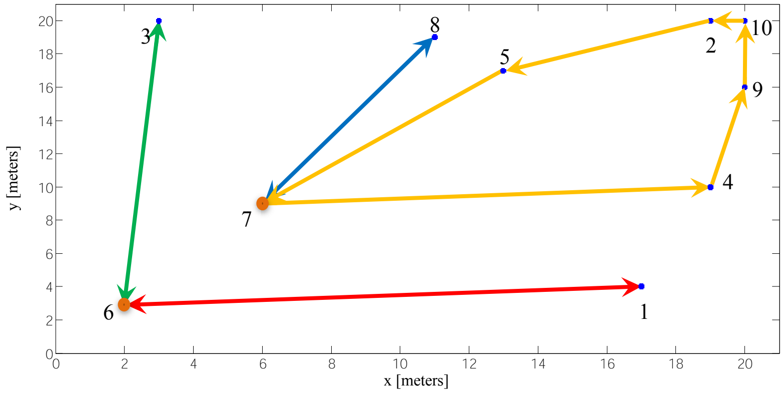


Figure 3.13: The Optimal MmTSP for the case where nodes 6 and 7 are the starting depots (worst-case scenario), and the return trip is allowed ($n = 10, m = 4, d = 2, (m_6, m_7) = (2, 2)$).

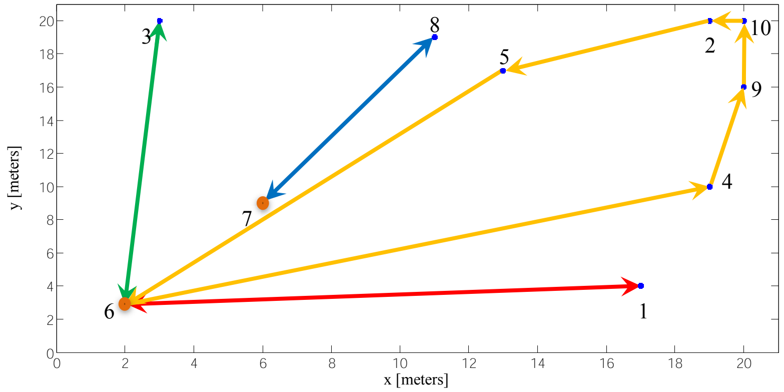


Figure 3.14: The Optimal MmTSP for the case where nodes 6 and 7 are the starting depots (worst-case scenario), and the return trip is allowed ($n = 10, m = 4, d = 2, (m_6, m_7) = (3, 1)$).

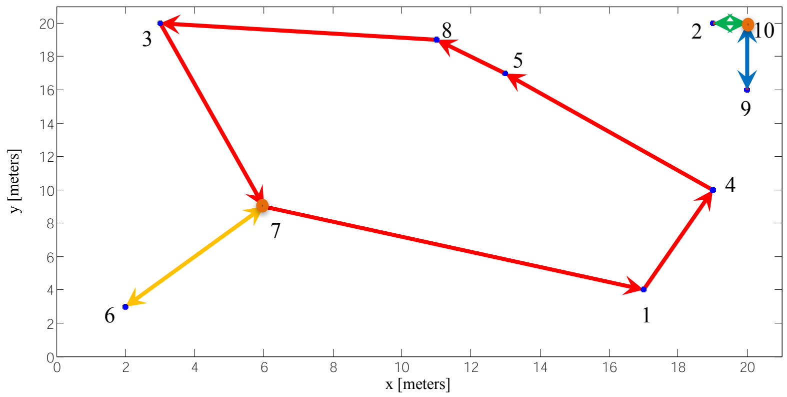


Figure 3.15: The Optimal MmTSP for the case where the starting depots are not prespecified and the return trip is allowed ($n = 10, m = 4, d = 2$).

distance by using the proposed method with non-prespecified starting depots. As

Table 3.1: Comparison between the results of scenarios 1 and 3.

	Scenario 1			Scenario 3 (non-prespecified starting depots)
Starting depots	$\# 1, 2$ (prespecified)			$\# 7, 10$ (calculated)
Initial number of robots	$m_1 = 1$ $m_2 = 3$	$m_1 = 2$ $m_2 = 2$	$m_1 = 3$ $m_2 = 1$	$m_7 = 2$ $m_{10} = 2$
at each depot	(prespecified)			(calculated)
Total travel distance per tour	80.56m	81.03m	97.53m	74.34m
Computation time	0.11s	0.06s	0.03s	2.84s
Reduction in travel distance per tour	6.26m (7.76%)	6.7m (8.258%)	23.18m (23.77%)	–

Table 3.2: Comparison between the results of scenarios 2 and 3.

	Scenario 2 (worst-case scenario)			Scenario 3 (non-prespecified starting depots)
Starting depots	$\# 6, 7$ (prespecified)			$\# 7, 10$ (calculated)
Initial number of robots	$m_6 = 1$ $m_7 = 3$	$m_6 = 2$ $m_7 = 2$	$m_6 = 3$ $m_7 = 1$	$m_7 = 2$ $m_{10} = 2$
at each depot	(prespecified)			(calculated)
Total travel distance per tour	116.69m	127.95m	140.47m	74.34m
Computation time	0.06s	0.05s	0.03s	2.84s
Reduction in travel distance per tour	42.35m (36.3%)	53.61m (41.9%)	66.13m (47.08%)	–

shown in Tables 3.1 and 3.2, the reduced travel distance in only one patrolling tour using non-prespecified starting depots approach (instead of prespecified starting depots) could range between 6.26 m (7.76% of the total travel distance) and 66.13 m (47.08% of the total travel distance), for the scenarios considered in this work, such significant reduction in travel distance can also lead to major improvement in the operation time of the robots due to the increase of the life time of the batteries (note that typically the patrolling operation can repeat for a long period of time). Tables 3.1 and 3.2 also show that the computation time for the new formulation is more than that of the conventional method. Moreover it is important to note that the computation is performed offline, which is not completely unimportant, but has no impact on the patrolling operation.

Figs. 3.16-3.22 show the optimal results for the case where return trip is not allowed, analogous to Figs. 3.9-3.15 with a comparison summarized in Tables 3.3 and 3.4. As expected, due to the additional constraint on the trajectory (concerning the return trips), the overall travel distance in this case is more than the previous case (when comparing same scenarios in both cases). Furthermore, using non-prespecified starting depots in this case (Fig. 3.22), the optimal choice for starting depots is obtained as $(m_7, m_9) = (1, 3)$. In this case, the reduced travel distance in only one patrolling tour using non-prespecified starting depots approach (instead of prespecified starting depots) could range between 12.57 m (12.38% of the total travel distance) and 77.17 m (46.45% of the total travel distance) for the scenarios considered in this case. It is worth noting that here the worst-case scenario with the return trip not allowed corresponds to the last configuration $(m_6, m_7) = (3, 1)$.

On the other hand, Figs. 3.23, 3.24 show the optimal output results with the return trip allowed and not allowed cases for Problem 2, respectively.

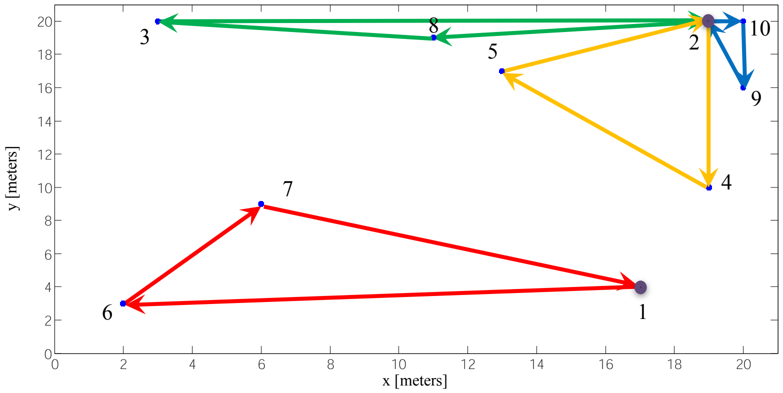


Figure 3.16: The Optimal MmTSP for the case where nodes 1 and 2 are the starting depots, and the return trip is not allowed $(n = 10, m = 4, d = 2, (m_1, m_2) = (1, 3))$.

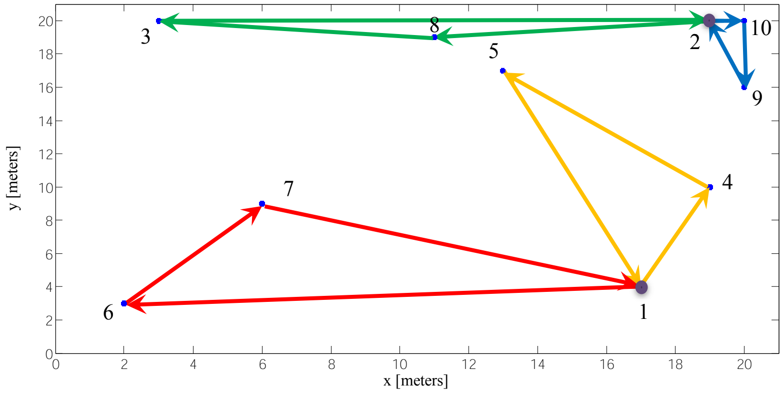


Figure 3.17: The Optimal MmTSP for the case where nodes 1 and 2 are the starting depots, and the return trip is not allowed $(n = 10, m = 4, d = 2, (m_1, m_2) = (2, 2))$.

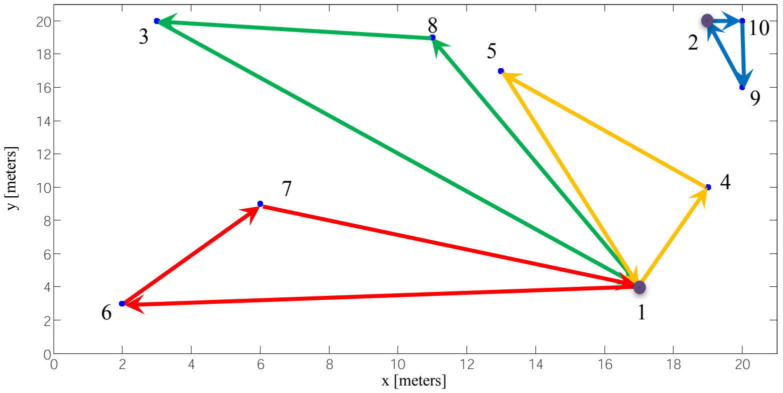


Figure 3.18: The Optimal MmTSP for the case where nodes 1 and 2 are the starting depots, and the return trip is not allowed $(n = 10, m = 4, d = 2, (m_1, m_2) = (3, 1))$.

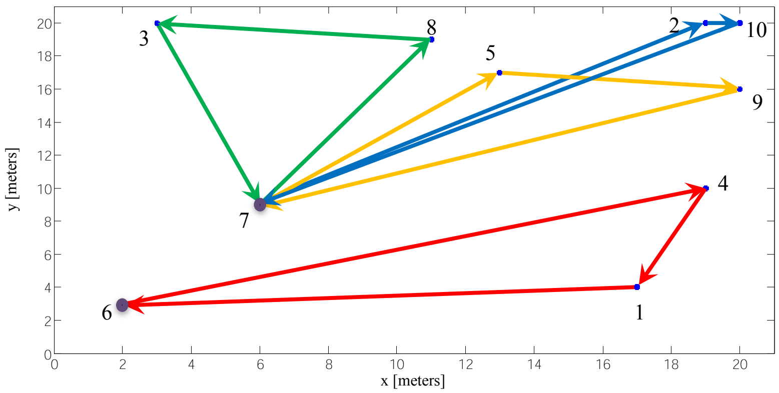


Figure 3.19: The Optimal MmTSP for the case where nodes 6 and 7 are the starting depots (worst-case scenario), and the return trip is not allowed $(n = 10, m = 4, d = 2, (m_6, m_7) = (1, 3))$.

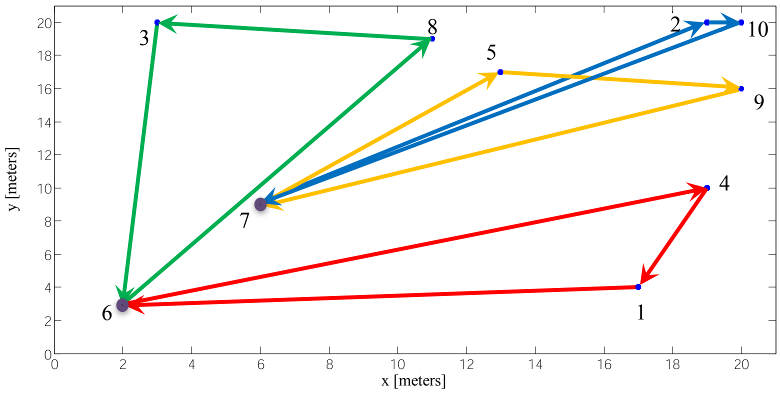


Figure 3.20: The Optimal MmTSP for the case where nodes 6 and 7 are the starting depots (worst-case scenario), and the return trip is not allowed $(n = 10, m = 4, d = 2, (m_6, m_7) = (2, 2))$.

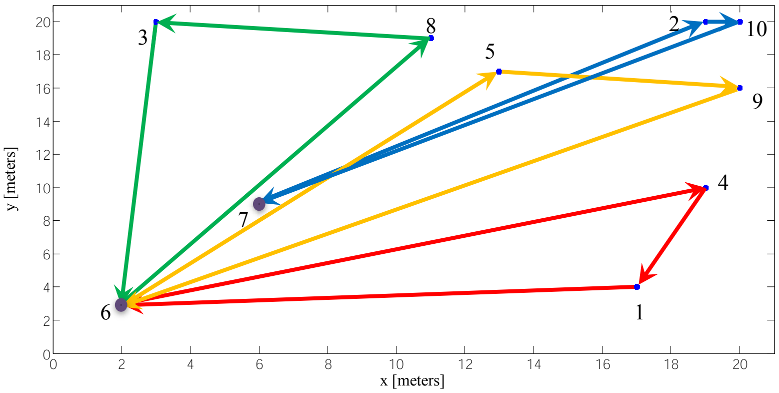


Figure 3.21: The Optimal MmTSP for the case where nodes 6 and 7 are the starting depots (worst-case scenario), and the return trip is not allowed $(n = 10, m = 4, d = 2, (m_6, m_7) = (3, 1))$.

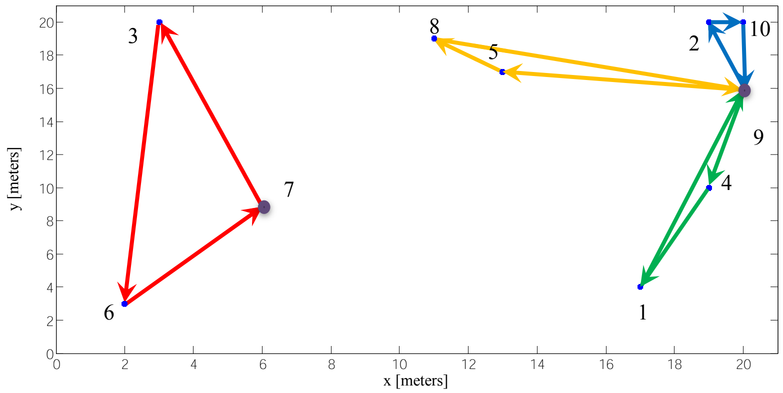


Figure 3.22: The Optimal MmTSP for the case where the starting depots are not prespecified and the return trip is not allowed ($n = 10, m = 4, d = 2$).

Table 3.3: Comparison between the results of scenarios 1 and 3 for the case where the return trip is not allowed.

	Scenario 1			Scenario 3 (non-prespecified starting depots)
Starting depots	# 1, 2 (prespecified)			# 7, 9 (calculated)
Initial number of robots at each depot	$m_1 = 1$ $m_2 = 3$	$m_1 = 2$ $m_2 = 2$	$m_1 = 3$ $m_2 = 1$	$m_7 = 1$ $m_9 = 3$ (calculated)
	(prespecified)			
Total travel distance per tour	101.5m	104.72m	118.07m	88.93m
Computation time	0.02s	0.02s	0.02s	0.98s
Reduction in travel distance per tour	12.57m (12.38%)	15.79m (15.08%)	29.15m (24.69%)	–

Table 3.4: Comparison between the results of scenarios 2 and 3 for the case where the return trip is not allowed.

	Scenario 2 (worst-case scenario)			Scenario 3 (non-prespecified starting depots)
Starting depots	# 6, 7 (prespecified)			# 7, 9 (calculated)
Initial number of robots at each depot	$m_6 = 1$ $m_7 = 3$	$m_6 = 2$ $m_7 = 2$	$m_6 = 3$ $m_7 = 1$	$m_7 = 1$ $m_9 = 3$ (calculated)
	(prespecified)			
Total travel distance per tour	139.57m	152.38m	166.1m	88.93m
Computation time	0.02s	0.03s	0.03s	0.98s
Reduction in travel distance per tour	50.64m (36.3%)	63.45m (41.64%)	77.17m (46.45%)	–

Tables 3.5 and 3.6 show a comparison between the two new formulations results. The first is the proposed TSP-based formulation and the second is the non-TSP-based formulation for the return trip allowed and not allowed cases, respectively. The comparison is related to the outputs in Figs. 3.23, 3.24 respectively.

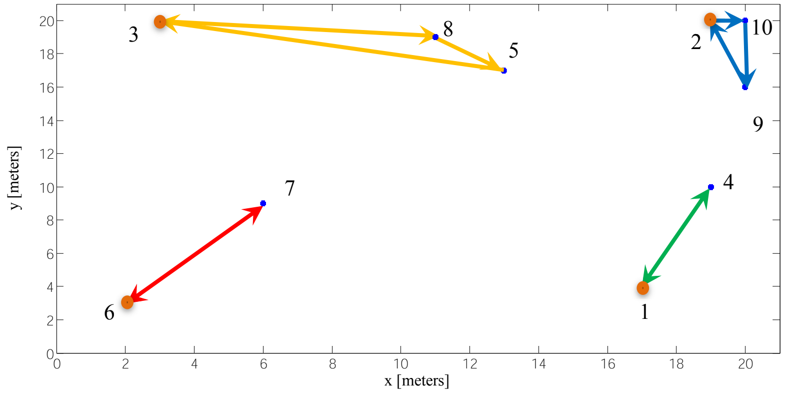


Figure 3.23: The optimal MmTSP for a prespecified number of viewpoints only and unknown number of depots and robots using either the TSP-based or non-TSP-based formulations with the return trip allowed ($n = 10$)

Table 3.5: A comparison between TSP-based and non-TSP-based formulations for the return trip allowed related to Fig. 3.23

	TSP-based Formulation	Non-TSP-based Formulation
Total travel distance per tour	57.5254m	
Computation time	2.25s	0.01s
Number of depots and robots optimally calculated	$m = d = 4$	

As shown in Tables 3.5 and 3.6, the total number of robots needed for the overall minimum travel distance trajectories is only 4 robots in case of return trip allowed, and 1 robot in case of return trip not allowed. Although, there could be more robots available in this configuration, but this would lead to a worse case. Thus if more robots are available, they shouldn't be used (saving their energy) according

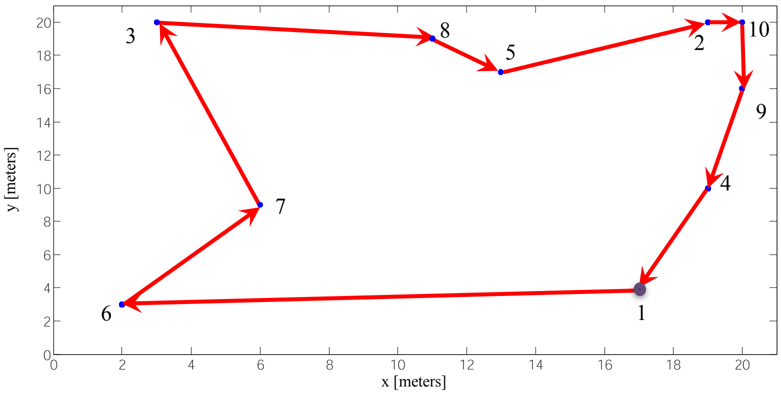


Figure 3.24: The optimal MmTSP for a prespecified number of viewpoints only and unknown number of depots and robots using either the TSP-based or non-TSP-based formulations with the return trip not allowed ($n = 10$)

Table 3.6: A comparison between TSP-based and non-TSP-based formulations for the return trip not allowed related to Fig. 3.24

	TSP-based Formulation	Non-TSP-based Formulation
Total travel distance per tour	68.6524m	
Computation time	1.89s	0.01s
Number of depots and robots optimally calculated	$m = d = 1$	

to the optimal output to obtain the optimal overall minimum distance. It can be also shown from the two tables that the TSP-based formulation gave the same results of another simpler non-TSP-based formulation with much less computation time.

In the next chapter, instead of minimizing the total travel distance in the patrolling problem, the total patrolling time is minimized. This is investigated while adopting the approach of having non-prespecified starting depots and robots, and for the more challenging case when the number of the optimal depots and robots are unknown.

Chapter 4

Time-Efficient Trajectory Optimization in Patrolling Problems

Instead of minimizing the total travel distance in the patrolling problem, it might be more important in some other scenarios to minimize the total time required to completely patrol the desired area. Typically, this can be achieved by dividing the total trajectories that are needed to patrol the whole area among more number of robots for the same given number of starting depots. Thus, increasing the number of robots for a given number of starting depots can lead to decreasing the total patrolling time or at least keeping it as it is.

Two new formulations and two new algorithms are presented in this chapter for the minimum-time trajectories in the patrolling problem, where a number of robots are desired to visit a given set of viewpoints in the shortest possible time. In the first problem, it is assumed that the starting depots and their corresponding robots are non-prespecified. In the second problem, it is desired to obtain the minimum-distance trajectories among all the possible minimum-time trajectories that could be the output of the first problem. In the third problem, the number of starting depots and the number of patrolling robots are also part of the optimization problem. Finally, the fourth problem is analogous to the second one but this time for the unknown number of depots and robots. The problem turns out to be a new variant of the TSP, namely, Minimum-Time Multidepot multiple Traveling Salesmen Problem (MTMmTSP).

Many articles investigate minimum-distance trajectory in the patrolling problem or intend to minimize the waiting time of salesmen in TSP with time windows (TSPTW) [50, 51, 52], which is closely related to the underlying patrolling problem. In the TSPTW problem, each customer has a service time and a time window between the ready time and due date. Each customer must be visited before its due date; otherwise, the tour is said to be infeasible. If, on the other hand, a vehicle arrives before the above-mentioned time window, it must wait.

The distinguishing features of the proposed problem statement is that first of all the starting depots and the number of robots assigned to each are not prespecified. The number of starting depots and robots may also be unknown. The cost function to be minimized is the total travel time of the robots as opposed to the travel distance or waiting time of the salesmen at the visited nodes for a given TSPTW. The number of viewpoints (and their configuration) is assumed to be known. Note that the starting depots belong to the set of viewpoints to be visited. The same assumptions for the TWMR dynamics and physical constraints still hold for these problems. It is also assumed that the robots are agile and can change directions quickly relative to their inter-activity travel times, the time between two activities can be described approximately in terms of the distances between the activities [53].

The results are compared to the results of the minimum-distance problem to clarify the difference between the two problems. Simulations confirm the efficiency of the proposed formulation in describing the minimum-time problem, whose solution demonstrates significant increase in the speed of accomplishing the underlying mission.

The rest of this chapter is organized as follows. The problem statement for the MTMmTSP with unspecified starting depots and robot assignment with both cases of known and unknown number of starting depots and robots are introduced in Section 4.1. Section 4.2 presents the proposed frameworks and the new proposed formulations. Simulation results are presented in Section 4.3.

4.1 Problem Statement

Consider a complete undirected graph $G(V, E)$, where $V = \{v_1, \ldots, v_n\}$ denotes a set of n viewpoints, through which a group of m unspecified robots, $m < n$, represented by the index set $M = \{1, \ldots, m\}$ are to perform a patrolling operation for monitoring an area, starting from d depots (note that here the starting depots and robots assigned to them are unspecified). Also, E denotes the set of all edges connecting any two nodes representing the movement of a robot between the corresponding viewpoints. Let c_{ij} represent the path length of the edge between viewpoints v_i and v_j, $\forall (i, j) \in E$, and define the symmetric cost (weight) matrix $[c_{ij}]$ accordingly. Note that since the path length is the distance between two connected nodes, it satisfies the triangular inequality, i.e., for any three nodes (i, j, k), $c_{ij} + c_{jk} \geq c_{ik}$ $\forall i, j, k \in V$ [38]. Four different types of patrolling problems are investigated here. In the first problem, it is assumed that the number d of starting depots and m of robots are known but nonspecified. The second problem is the same as the first one but with one more requirement, which is to obtain the minimum-distance trajectories among all the possible minimum-time trajectories that were optimally obtained from the first problem. In the third problem, the information about the number of starting depots d and robots m is not prespecified. The minimum-time trajectories are required to be computed such that each robot follows one of the trajectories connecting a subset of the viewpoints starting from one of the depots (which are not prespecified), and returns back to it. In the fourth problem, it is required to obtain the minimum-distance trajectories among all the possible minimum-time trajectories that were optimally obtained from the third problem. In all problems, there are two cases of interest: (i) when a return trip is allowed, i.e., the robot can visit only one viewpoint before returning back to its starting depot, and (ii) when a return trip is not allowed.

4.2 Proposed Frameworks

Problem 1: Minimum-Time Multidepot multiple Traveling Salesmen Problem (MTMmTSP) with unspecified starting depots and robots. Given the graph $G(V, E)$ and cost matrix $[c_{ij}]$ defined in the previous section, it is desired to formulate the MTMmTSP in an optimization framework for the case when the return trip is allowed as follows. Let a link connecting two arbitrary nodes v_i, v_j on the trajectory of depot v_k and tracked by robot l be represented by a binary variable x_{ijkl} which is equal to 1 if the trajectory is optimal and 0 otherwise. The cost function to be minimized can be represented as the upper limit of any travel distance of any robot, assuming that the velocity of all the robots is equal and constant. This can be expressed as:

$$\min_{x_{ijkl}} \epsilon \tag{4.1}$$

where

$$\sum_{k \in V} \sum_{i \in V} \sum_{\substack{j \in V \\ j \neq i}} c_{ij} x_{ijkl} \leq \epsilon, \qquad l \in M \tag{4.2}$$

where the upper limit of the travel distance is denoted by the variable ϵ. Thus, the overall minimum-time trajectory will be equal to the longest travel distance among the distances traveled by each robot. The number of departures from and arrivals at all the starting depots, which are assumed to be unspecified, is equal to the total number of robots m, i.e.:

$$\sum_{l\in M}\sum_{k\in V}\sum_{\substack{j\in V\\ j\neq k}} x_{kjkl} = m \tag{4.3}$$

$$\sum_{l\in M}\sum_{k\in V}\sum_{\substack{j\in V\\ j\neq k}} x_{jkkl} = m \tag{4.4}$$

The total number of departures from and arrivals at all other nodes not including the depots (which are unspecified as well) equals $(n-d)$, i.e., only one departure and one arrival for any node not including the starting d depots. This is expressed by:

$$\sum_{l\in M}\sum_{k\in V}\sum_{\substack{j\in V\\ j\neq k}} x_{kjkl} + \sum_{l\in M}\sum_{k\in V}\sum_{\substack{i\in V\\ i\neq k}}\sum_{\substack{j\in V\\ j\neq i,k}} x_{ijkl} = n-d \tag{4.5}$$

$$\sum_{l\in M}\sum_{k\in V}\sum_{\substack{j\in V\\ j\neq k}} x_{jkkl} + \sum_{l\in M}\sum_{k\in V}\sum_{\substack{i\in V\\ i\neq k}}\sum_{\substack{j\in V\\ j\neq i,k}} x_{jikl} = n-d \tag{4.6}$$

respectively. To ensure that any optimal tour is not shared among more than one depot, the route continuity constraint is imposed as follows:

$$x_{kjkl} + \sum_{\substack{i\in V\\ i\neq j,k}} x_{ijkl} - x_{jkkl} - \sum_{\substack{i\in V\\ i\neq j,k}} x_{jikl} = 0, \qquad j,k\in V, l\in M, j\neq k \tag{4.7}$$

Let ω_k be a set of auxiliary binary variables that corresponds to the optimal depots such that, it is equal to zero if v_k is an optimal depot and equals to one otherwise for any $k\in V$. A constraint on the sum of these auxiliary variables is given by:

$$\sum_{k\in V}\omega_k = n-d \tag{4.8}$$

Since each robot can follow one trajectory only, the sum of all the departing and arriving trajectories of any robot from all starting depots to any other node is always 1.

$$\sum_{k\in V}\sum_{\substack{j\in V\\ j\neq k}} x_{kjkl} = 1, \qquad l \in M \tag{4.9}$$

$$\sum_{k\in V}\sum_{\substack{j\in V\\ j\neq k}} x_{jkkl} = 1, \qquad l \in M \tag{4.10}$$

The minimum and maximum number of robots that could be placed at an optimal depot are given by:

$$1 \leq \sum_{l\in M}\sum_{\substack{j\in V\\ j\neq k}} x_{kjkl} + \left(m-(d-1)\right)\omega_k \leq m-(d-1)\,, \qquad k \in V \tag{4.11}$$

$$1 \leq \sum_{l\in M}\sum_{\substack{j\in V\\ j\neq k}} x_{jkkl} + \left(m-(d-1)\right)\omega_k \leq m-(d-1)\,, \qquad k \in V \tag{4.12}$$

respectively. The total number of arrivals at and departures from any node v_j from any node on a trajectory whose starting depot is v_k (including depot v_k itself) is one if v_j is not a depot and is zero if v_j is another optimal depot or this link is not an optimal link as described by the following inequalities:

$$0 \leq \sum_{l\in M}\sum_{\substack{k\in V\\ k\neq j}} x_{kjkl} + \sum_{l\in M}\sum_{\substack{k\in V\\ k\neq j}}\sum_{\substack{i\in V\\ i\neq j,k}} x_{ijkl} \leq 1, \qquad j \in V \tag{4.13}$$

$$0 \leq \sum_{l\in M}\sum_{\substack{k\in V\\ k\neq j}} x_{jkkl} + \sum_{l\in M}\sum_{\substack{k\in V\\ k\neq j}}\sum_{\substack{i\in V\\ i\neq j,k}} x_{jikl} \leq 1, \qquad j \in V \tag{4.14}$$

Moreover, the following inequality is a constraint that prevents any depot from being part of another depot's trajectory:

$$-4 \leq \sum_{l \in M} \sum_{\substack{k \in V \\ k \neq i,j}} x_{ijil} + x_{jiil} + x_{kjil} + x_{jkil} + x_{jijl} + x_{ijjl} + x_{kijl} + x_{ikjl} - 2(\omega_i + \omega_j) \leq 0, \qquad i,j \in V, i \neq j \tag{4.15}$$

On the other hand, the SEC can be presented by the following inequality:

$$u_i - u_j + (n-m) \sum_{l \in M} \sum_{\substack{k \in V \\ k \neq i,j}} x_{ijkl} + (n - 2(d-1))\omega_j \leq n - m - 1 + (n - 2(d-1)), \qquad i,j \in V, i \neq j \tag{4.16}$$

where v_j is an optimal depot if $\omega_j = 0$, for which the set of inequalities representing an optimal tour will be valid. The maximum number of tour links that could exist in any optimal tour is $(n - 2(d-1))$. The basic constraint here is that each node can take only one arrival and one departure . This condition is described by the following inequalities which guarantee the formation of the trajectories only from optimal depots:

$$0 \leq \sum_{l \in M} \sum_{\substack{i \in V \\ i \neq j}} x_{ijkl} + \omega_k \leq 1, \qquad j,k \in V, k \neq j \tag{4.17}$$

$$0 \leq \sum_{l \in M} \sum_{\substack{j \in V \\ j \neq i}} x_{jikl} + \omega_k \leq 1, \qquad j,k \in V, k \neq j \tag{4.18}$$

In the case when the return trip is not allowed, the following additional constraint is considered:

$$\sum_{l \in M} \sum_{\substack{k \in V \\ k \neq j}} x_{kjkl} + \sum_{l \in M} \sum_{\substack{k \in V \\ k \neq j}} x_{jkkl} \leq 1, \qquad j \in V \tag{4.19}$$

Equations 4.1-4.19 represent the new formulation for the minimum time trajectories for the non-prespecified depots and robots.

Problem 2: Minimum-Distance Multidepot multiple Traveling Salesmen Problem (MmTSP) among the possible MTMmTSP optimal trajectories with unspecified starting depots and robots. Given the problem statement and configuration of problem 1, the optimal result trajectory is not unique, i.e., there could be several optimal trajectories with the same upper bound trajectory length tracked by one of the robots, but for the rest of the robots' trajectories, several results could be obtained. It is desired to obtain the minimum-distance trajectories among all these possible optimal trajectories. This is also desired for both cases, the return trip allowed and not allowed cases.

The new algorithm for obtaining the minimum-distance trajectories among all the possible minimum-time trajectories can be summarized in two steps. The first step is to solve the minimum-time trajectories as was introduced in problem 1. The second step is to assign the obtained optimal minimum-time result ϵ which represents the maximum travel distance among the robots to the first robot, and to minimize the overall distance of the other robots while preventing any of them to exceed ϵ. In other words, the optimal result obtained for problem 1 is considered to be a constraint in problem 2.

Assuming that the optimal result obtained from problem 1 is ϵ^*, which represents the minimum-time optimal result, it will be assigned to the first robot as in Eq. (4.21). Whereas the new cost function representing the minimum overall distance for all the robots not including the first robot is represented in Eq. (4.20).

$$\min_{x_{ijkl}} \sum_{k \in V} \sum_{i \in V} \sum_{\substack{j \in V \\ j \neq i}} c_{ij} x_{ijkl}, \qquad l \in M - \{1\} \tag{4.20}$$

$$\sum_{k\in V}\sum_{i\in V}\sum_{\substack{j\in V\\ j\neq i}} c_{ij}x_{ijk1} = \epsilon^* \tag{4.21}$$

The set of Eqs. (4.3) - (4.18) or the set of Eqs. (4.3) - (4.19) are reapplied for the return trip allowed case or for the return trip not allowed case, respectively, without including the first robot in all the equations. Each robot not including the first one is prevented from exceeding the optimal result (ϵ^*) of problem 1 , this can be represented as:

$$\sum_{k\in V}\sum_{i\in V}\sum_{\substack{j\in V\\ j\neq i}} c_{ij}x_{ijkl} \leq \epsilon^*, \qquad l \in M - \{1\} \tag{4.22}$$

Problem 3: Unknown number of depots and robots in the MTMmTSP problem. The MTMmTSP problem discussed above was used to obtain the minimum-time optimal trajectories with a given number of d depots and a number of m robots, without prespecifying their locations. Solving the minimum-time problem described above without knowing the number of robots or starting depots would obviously result in a faster trajectory (due to the removal of the corresponding constraints from the optimization problem). The possible number of robots to perform a patrolling operation for monitoring an area ranges between 1 and $n-1$ unspecified robots, represented by the new index set $L = \{1,\ldots,n-1\}$ .The problem is formulated for both cases of the return trip allowed and not allowed as introduced before. This formulation has n^4 binary variables, and n^3 constraints (with an impact on the complexity of computations). The minimization problem in this case is described as:

$$\min_{x_{ijkl}}\ \epsilon \tag{4.23}$$

where

$$\sum_{k\in V}\sum_{i\in V}\sum_{\substack{j\in V\\ j\neq i}} c_{ij}x_{ijkl} \leq \epsilon, \qquad l \in L \tag{4.24}$$

The number of departures from and arrivals at all the starting depots (which are unknown in this case), satisfy the following inequalities:

$$n - \sum_{k\in V}\omega_k \leq \sum_{l\in L}\sum_{\substack{k\in V\\ k\neq j}}\sum_{j\in V} x_{kjkl} \leq \sum_{k\in V}\omega_k \tag{4.25}$$

$$n - \sum_{k\in V}\omega_k \leq \sum_{l\in L}\sum_{\substack{k\in V\\ k\neq j}}\sum_{j\in V} x_{jkkl} \leq \sum_{k\in V}\omega_k \tag{4.26}$$

respectively (note that the left side of the above inequalities is, in fact, the optimal number of depots). The total number of departures from and arrivals at all other nodes not including the depots, which are unknown equals the sum of the auxiliary binary variables (excluding the optimal number of depots from the total number of viewpoints). This is expressed by:

$$\sum_{l\in L}\sum_{k\in V}\sum_{\substack{j\in V\\ j\neq k}} x_{kjkl} + \sum_{l\in L}\sum_{k\in V}\sum_{\substack{i\in V\\ i\neq k}}\sum_{\substack{j\in V\\ j\neq i,k}} x_{ijkl} = \sum_{k\in V}\omega_k \tag{4.27}$$

$$\sum_{l\in L}\sum_{k\in V}\sum_{\substack{j\in V\\ j\neq k}} x_{jkkl} + \sum_{l\in L}\sum_{k\in V}\sum_{\substack{i\in V\\ i\neq k}}\sum_{\substack{j\in V\\ j\neq i,k}} x_{jikl} = \sum_{k\in V}\omega_k \tag{4.28}$$

To ensure that an optimal route is not shared between more than one depot, the following route continuity constraint is imposed:

$$x_{kjkl} + \sum_{\substack{i\in V\\ i\neq j,k}} x_{ijkl} - x_{jkkl} - \sum_{\substack{i\in V\\ i\neq j,k}} x_{jikl} = 0, \qquad j,k \in V,\ l \in L, j \neq k \tag{4.29}$$

The next equation represents a new constraint on the sum of the new auxiliary variables, which represents a constraint on the total sum of the optimal depots:

$$n/2 \leq \sum_{k \in V} \omega_k \leq n-1 \tag{4.30}$$

The sum of all the departing and arriving trajectories of any robot from all starting depots to any other node is only 1, as described in the following inequalities:

$$0 \leq \sum_{k \in V} \sum_{\substack{j \in V \\ j \neq k}} x_{kjkl} \leq 1, \qquad l \in L \tag{4.31}$$

$$0 \leq \sum_{k \in V} \sum_{\substack{j \in V \\ j \neq k}} x_{jkkl} \leq 1, \qquad l \in L \tag{4.32}$$

The possible number of arrivals and departures from any depot can be represented by:

$$0 \leq \sum_{l \in L} \sum_{\substack{j \in V \\ j \neq k}} x_{kjkl} + (n-1)\omega_k \leq n-1, \quad k \in V \tag{4.33}$$

$$0 \leq \sum_{l \in L} \sum_{\substack{j \in V \\ j \neq k}} x_{jkkl} + (n-1)\omega_k \leq n-1, \quad k \in V \tag{4.34}$$

The total number of arrivals at and departure from any node v_j from any node on a trajectory whose starting depot v_k (including depot v_k itself) is one in case of v_j is not a depot and equals zero in case v_j is another depot. This is described by the following inequalities:

$$0 \leq \sum_{l \in L} \sum_{\substack{k \in V \\ k \neq j}} x_{kjkl} + \sum_{l \in L} \sum_{\substack{k \in V \\ k \neq j}} \sum_{\substack{i \in V \\ i \neq j,k}} x_{ijkl} \leq 1, \qquad j \in V \tag{4.35}$$

$$0 \leq \sum_{l \in L} \sum_{\substack{k \in V \\ k \neq j}} x_{jkkl} + \sum_{l \in L} \sum_{\substack{k \in V \\ k \neq j}} \sum_{\substack{i \in V \\ i \neq j,k}} x_{jikl} \leq 1, \qquad j \in V \tag{4.36}$$

respectively. Now, in order to ensure that no depot is included in another depot's trajectory, the following constraint is imposed:

$$-4 \leq \sum_{l \in L} \sum_{\substack{k \in V \\ k \neq i,j}} x_{ijil} + x_{jiil} + x_{kjil} + x_{jkil} + x_{jijl} + x_{ijjl} + x_{kijl} + x_{ikjl} - 2(\omega_i + \omega_j) \leq 0, \qquad i,j \in V, i \neq j \tag{4.37}$$

The SEC can also be presented by:

$$u_i - u_j + n \sum_{l \in L} \sum_{\substack{k \in V \\ k \neq i,j}} x_{ijkl} + (n)\omega_j \leq n - 1 + n, \qquad i,j \in V, i \neq j \tag{4.38}$$

One of the basic constraints in this kind of patrolling problem is that each node is arrived at or departed from only once, and by a trajectory which starts from only one optimal depot. This is formulated as follows:

$$0 \leq \sum_{l \in L} \sum_{\substack{i \in V \\ i \neq j}} x_{ijkl} + \omega_k \leq 1, \qquad j,k \in V, j \neq k \tag{4.39}$$

$$0 \leq \sum_{l \in L} \sum_{\substack{i \in V \\ i \neq j}} x_{jikl} + \omega_k \leq 1, \qquad j,k \in V, j \neq k \tag{4.40}$$

respectively. The following additional constraint is imposed if the return trip is not allowed:

$$\sum_{l \in L} \sum_{\substack{k \in V \\ k \neq j}} x_{kjkl} + \sum_{l \in L} \sum_{\substack{k \in V \\ k \neq j}} x_{jkkl} \leq 1, \qquad j \in V \tag{4.41}$$

Equations 4.23-4.41 represent the new formulation for the minimum time trajectories of unknown number of depots and robots.

Problem 4: Minimum-Distance Multidepot multiple Traveling Salesmen Problem (MmTSP) among the possible MTMmTSP optimal trajectories

with unknown number of depots and robots. Given the problem statement and configuration of problem 3, the optimal result trajectory is not unique, i.e., there could be several optimal trajectories with the same upper bound trajectory length tracked by one of the robots, but for the rest of the robots' trajectories, several results could be obtained. It is desired to obtain the minimum-distance trajectories among all these possible optimal trajectories. This is also desired for both cases, the return trip allowed and not allowed cases.

The new algorithm for obtaining the minimum-distance trajectories among all the possible minimum-time trajectories in the case of unknown number of depots and robots can be summarized in two steps. The first step is to solve the minimum-time trajectories as was introduced in problem 3. The second step is to assign the obtained optimal minimum-time result ϵ which represents the maximum travel distance among the robots to one robot, and to minimize the overall distance of the other possible number of robots and depots which are still unknown while preventing any trajectory to exceed ϵ. In other words, the optimal result obtained for problem 3 is considered to be a constraint in problem 4.

Assuming that the optimal result obtained from problem 3 is ϵ^+, which represents the minimum-time optimal result for the case of unknown number of depots and robots, it will be assigned to one robot as in Eq. (4.43). Whereas the new cost function representing the minimum overall distance for all the possible robots is represented in Eq. (4.42).

$$\min_{x_{ijkl}} \sum_{k\in V} \sum_{i\in V} \sum_{\substack{j\in V \\ j\neq i}} c_{ij} x_{ijkl}, \qquad l \in L - \{1\} \tag{4.42}$$

$$\sum_{k\in V} \sum_{i\in V} \sum_{\substack{j\in V \\ j\neq i}} c_{ij} x_{ijk1} = \epsilon^+ \tag{4.43}$$

The set of Eqs. (4.25) - (4.40) or the set of Eqs. (4.25) - (4.41) are reapplied for the return trip allowed case or for the return trip not allowed case, respectively, without including the already obtained robot in all the equations. Each robot not including the first one is prevented from exceeding the optimal result (ϵ^+) of problem 3, this can be represented as:

$$\sum_{k\in V}\sum_{i\in V}\sum_{\substack{j\in V\\ j\neq i}} c_{ij}x_{ijkl} \leq \epsilon^+, \qquad l \in L-\{1\} \tag{4.44}$$

Problems 1-4 are all expressed in MIP frameworks with linear constraints. Thus, they are convex and always have a feasible optimal solution. The proposed formulations introduce efficient optimization frameworks, which can be handled by existing solvers. For example, Gurobi Optimizer 6.0 [102] optimization software or MOSEK optimization software [101] can be used to solve the previous formulations where a linear-programming based branch-and-bound algorithm is used to solve such problems.

4.3 Simulation Results

Consider the patrolling problem for a field of size 20m by 20m, where a set of $n = 9$ nodes (viewpoints) are to be visited. Let the number of robots be $m = 3$, with the number of depots $d = 2$ for the first and second problems. Assuming the robots move with constant velocity 1 m/s, and using the proposed formulations either for the unspecified starting depots and their corresponding starting robots or the unknown number of depots and robots. It is desired to find the minimum-time trajectories and the minimum-distance trajectories among all possible minimum-time trajectories. MATLAB was used with the Gurobi Optimizer 6.0 [102] optimization software to obtain all the results using Intel Core i7-3537U @ 2.00GHz processor with 8 GB RAM.

Figs. 4.1-4.3 show the minimum-time trajectory, the minimum-distance trajectory that can achieve the obtained minimum-time trajectory and the minimum-distance trajectory obtained using Chapter 3 formulation for the unspecified depots and robots, all for the return trip allowed case, respectively. A comparison among the three figures is provided in Table 4.1, which clearly demonstrates the difference among the proposed problems. Figs. 4.4-4.6 show the corresponding figures of Figs. 4.1-4.3, respectively, but this time for the return trip not allowed case. The corresponding comparison among the these figures is given in Table 4.2.

Analogously, Figs. 4.7-4.12 present the optimal trajectories analogous to Figs. 4.1-4.6 but for the case where the number of depots and robots are unknown. Similar comparisons are given in Tables 4.3 and 4.4 among the results for the return trip allowed and not allowed cases, respectively.

As expected, Tables 4.1-4.4 confirm that the total travel distance obtained in the minimum-distance approach is less than that in the minimum-time approach, whereas the total travel time is less in the case of the minimum-time approach for different cases of the unspecified starting depots and robots with known or unknown number, and with the return trip is allowed or not allowed. It can also be verified from the tables that the computation time for the minimum-time approach is more than that of the minimum-distance approach. This is due to the nature of the optimization framework as well as the increase in the number of variables in the case of the minimum-time approach.

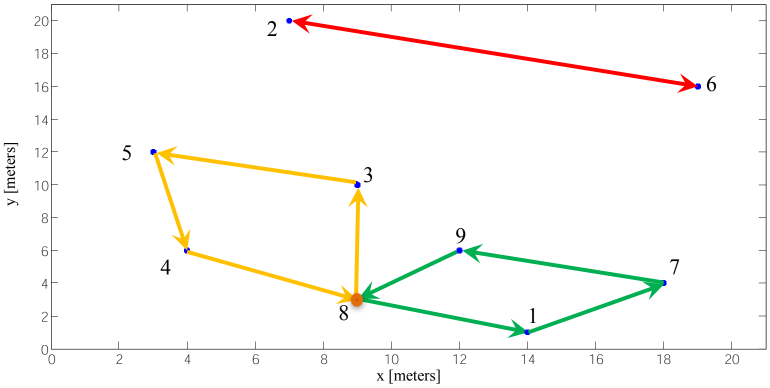


Figure 4.1: MTMmTSP using unspecified starting depots and robots with the return trip allowed $(n = 9, m = 3, d = 2)$

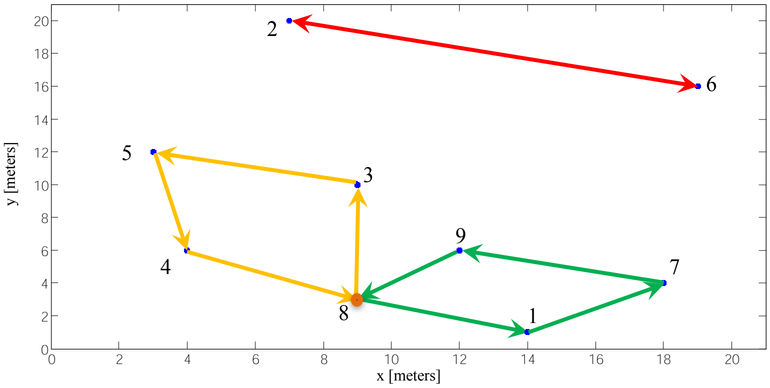


Figure 4.2: The minimum-distance of the possible MTMmTSP outputs using unspecified starting depots and robots with the return trip allowed $(n = 9, m = 3, d = 2)$

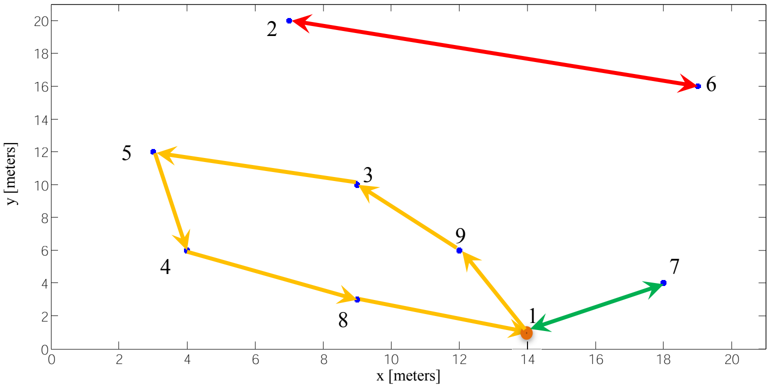


Figure 4.3: Minimum-distance MmTSP using unspecified starting depots and robots with the return trip allowed $(n = 9, m = 3, d = 2)$

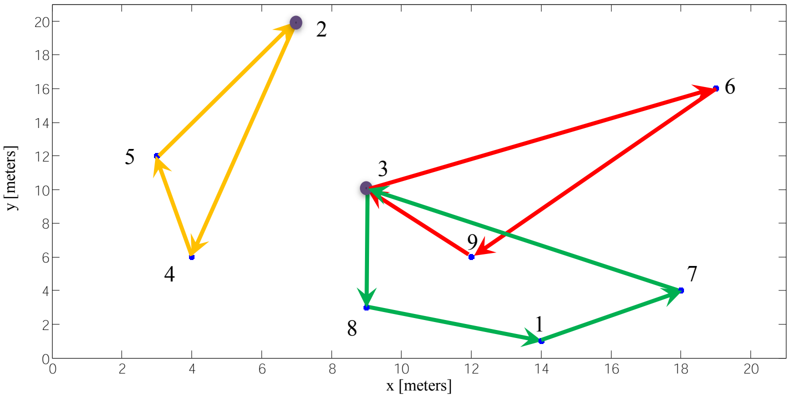


Figure 4.4: MTMmTSP using unspecified starting depots and robots with the return trip not allowed $(n = 9, m = 3, d = 2)$

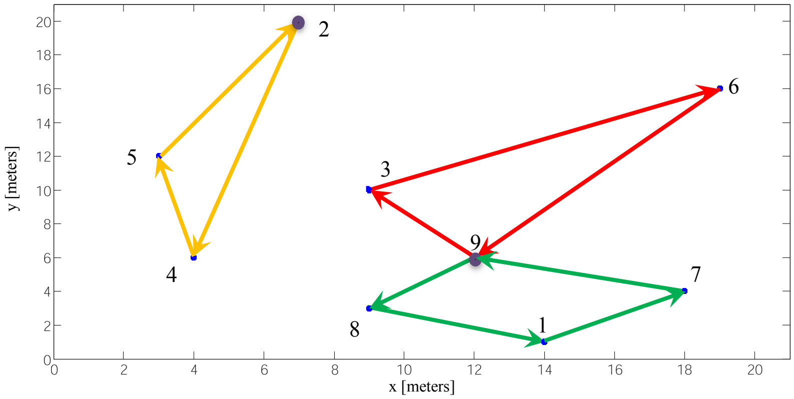


Figure 4.5: The minimum-distance of the possible MTMmTSP outputs using unspecified starting depots and robots with the return trip not allowed $(n = 9, m = 3, d = 2)$

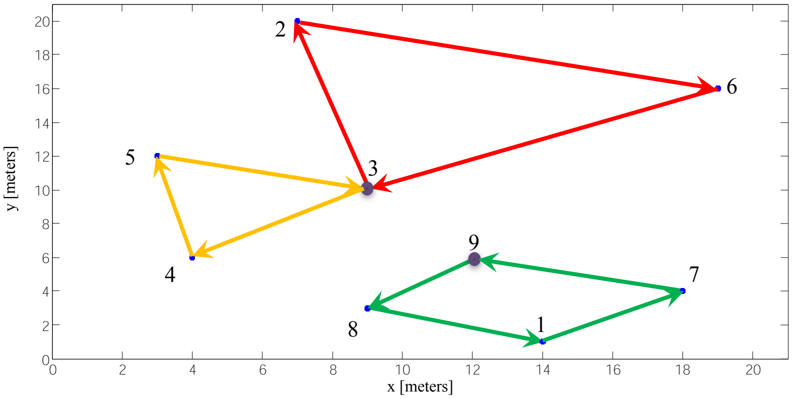


Figure 4.6: Minimum-distance MmTSP using unspecified starting depots and robots with the return trip not allowed $(n = 9, m = 3, d = 2)$

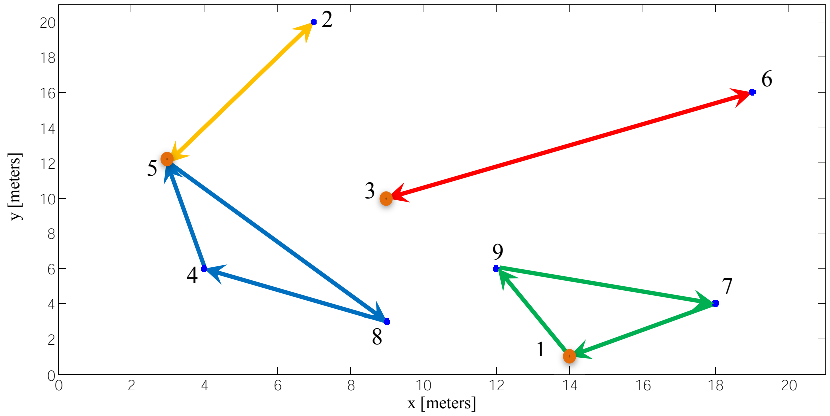


Figure 4.7: MTMmTSP using unknown starting depots and robots with the return trip allowed ($n = 9$)

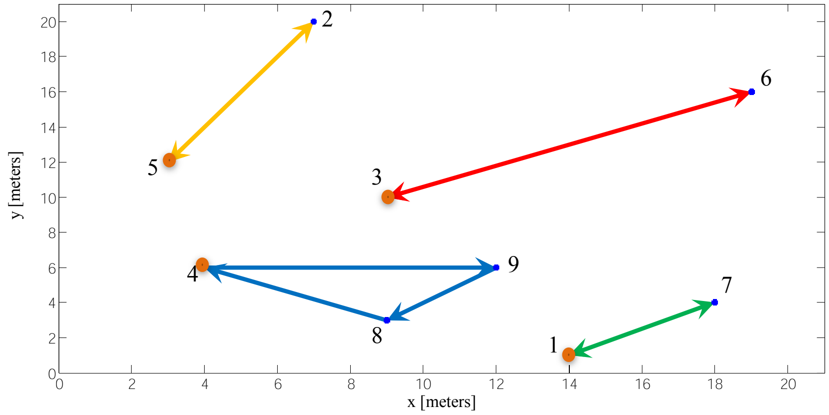


Figure 4.8: The minimum-distance of the possible MTMmTSP outputs using unknown starting depots and robots with the return trip allowed ($n = 9$)

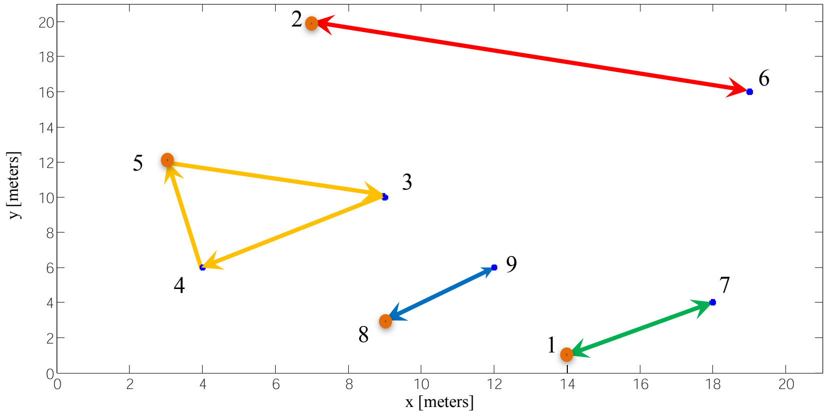


Figure 4.9: Minimum-distance MmTSP using unknown starting depots and robots with the return trip allowed ($n = 9$)

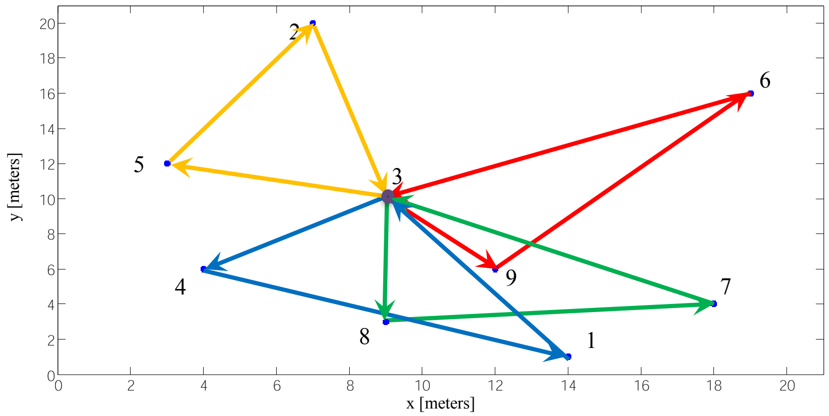


Figure 4.10: MTMmTSP using unknown starting depots and robots with the return trip not allowed ($n = 9$)

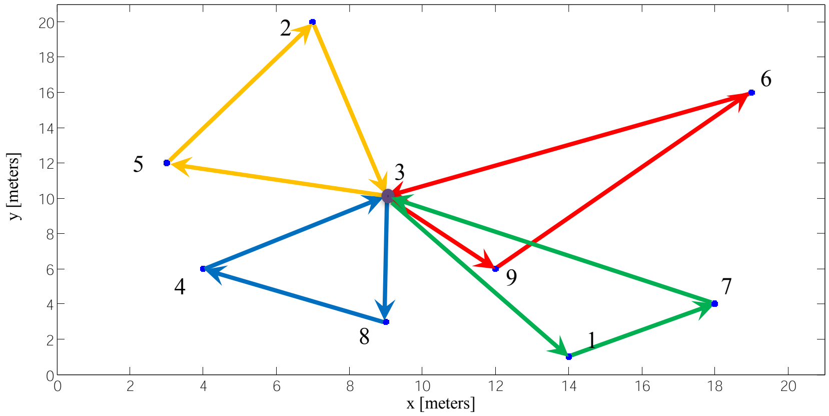


Figure 4.11: The minimum-distance of the possible MTMmTSP outputs using unknown starting depots and robots with the return trip not allowed ($n = 9$)

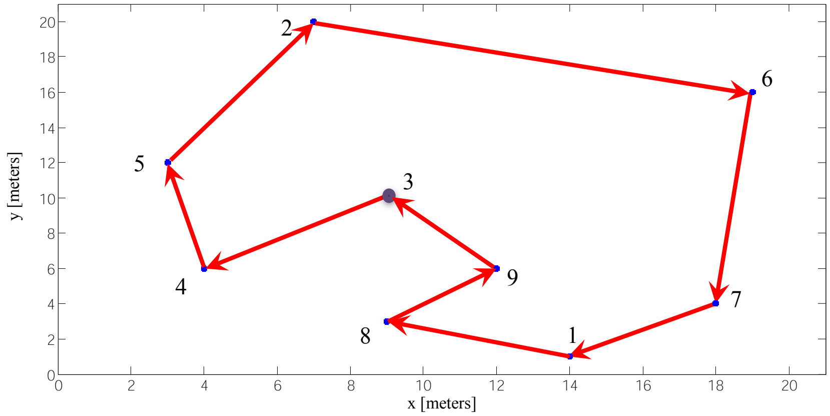


Figure 4.12: Minimum-distance MmTSP using unknown starting depots and robots with the return trip not allowed ($n = 9$)

Table 4.1: Comparison among Figs. 4.1-4.3

	MTMmTSP	Minimum-Distance among MTMmTSP results	MmTSP
Starting depots	# $2, 8$ (calculated)	# $2, 8$ (calculated)	# $1, 2$ (calculated)
Initial number of robots at each depot	$m_2 = 1,\ m_8 = 2$ (calculated)	$m_2 = 1,\ m_8 = 2$ (calculated)	$m_1 = 2,\ m_2 = 1$ (calculated)
Total travel distance per tour	71.4889m	71.4889m	69.3068m
Total travel time per tour	25.2982s	25.2982s	34.0086s
Computation time	16.36s	23.13s	1.16s

Table 4.2: Comparison among Figs. 4.4-4.6

	MTMmTSP	Minimum-Distance among MTMmTSP results	MmTSP
Starting Depots	# $2, 3$ (calculated)	# $2, 9$ (calculated)	# $3, 9$ (calculated)
Initial number of robots at each depot	$m_2 = 1,\ m_3 = 2$ (calculated)	$m_2 = 1,\ m_9 = 2$ (calculated)	$m_3 = 2,\ m_9 = 1$ (calculated)
Total travel distance per tour	86.4152m	79.1658m	74.2718m
Total travel time per tour	29.3449s	29.3449s	34.509s
Computation time	4.24s	59.25s	1.11s

Table 4.3: Comparison among Figs. 4.7-4.9

	MTMmTSP	Minimum-Distance among MTMmTSP results	MmTSP
Starting Depots	# $1, 3, 5$ (calculated)	# $1, 3, 4, 5$ (calculated)	# $1, 2, 5, 8$ (calculated)
Initial number of robots at each depot	$m_1 = m_3 = 1,\ m_5 = 2$ (calculated)	$m_1 = m_3 = m_4 = m_5 = 1$ (calculated)	$m_1 = m_2 = m_5 = m_8 = 1$ (calculated)
Total travel distance per tour	80.6524m	69.2859m	62.5939m
Total travel time per tour	23.3238s	23.3238s	25.2982s
Computation time	112.75s	131.17s	0.1s

Table 4.4: Comparison among Figs. 4.10-4.12

	MTMmTSP	Minimum-Distance among MTMmTSP results	MmTSP
Starting Depots	# 3 (calculated)	# 3 (calculated)	# 1 (calculated)
Initial number of robots at each depot	$m_3 = 4$ (calculated)	$m_3 = 4$ (calculated)	$m_3 = 1$ (calculated)
Total travel distance per tour	109.0865m	99.6818m	65.7487m
Total travel time per tour	28.8685s	28.8685s	65.7487s
Computation time	2541.54s	4088.8s	0.1s

In the next chapter, the more challenging problem of having Dubins' trajectories with minimum turning radius are investigated. A new algorithm is introduced to convert an ETSP optimal solution to a kinematic-feasible Dubins' trajectory suitable to be tracked by Dubins' vehicles.

Chapter 5

Pulleys Algorithm for Obtaining Sub-Optimal Dubins' Trajectories for Patrolling Problem

In the previous chapters, the patrolling problem was introduced to be performed using TWMR that were assumed to have the ability of turning on the spot, i.e., they have two differential servomotors to track the trajectories optimally obtained as a solution to the ETSP. They were also assumed to be agile and can change directions quickly relative to the inter-activity travel times, the time between two activities could then be described approximately in terms of the distances between the activities. But what if the wheeled robots doesn't have this ability, i.e., the wheeled robots can only track planar curvature-bounded trajectories due to significant kinematic constraints such as limited turning radius, and the inability to move in a reverse direction, then the problem will be more challenging because the ETSP solution will provide poor estimates of actual travel time and vehicle location [53]. This kind of robots are known by Dubins' vehicles, they can move with bounded curvature of minimum-turning radius and constant forward velocity.

5.1 Background

In 1889, Andrey Andreevich Markov published a paper [104] where he considered several mathematical problems that represent the minimum-time point-to-point path planning problem with bounded curvature related to the design of railways. The simplest among these problems (and the first one in course of the presentation) is described as follows. Find a minimum length curve between two points in the plane provided that the curvature radius of the curve should not be less than a given quantity and the tangent to the curve should have a given direction at the initial point. In 1957, in American Journal of Mathematics, Lester Eli Dubins considered a problem in the plane on finding the minimum length curve connecting two given points among smooth curves of bounded curvature with minimum radius r_{min} provided that the initial direction at the first point and terminal direction at the second point are prespecified [54]. He proved that circular arcs of radius r_{min} and straight-line segments (of infinite radius) can be employed to plan a minimum-time trajectory. He prescribed the set of sufficient family $\mathcal{F}_D$ of optimal paths as follows:

- Set CCC of types RLR, LRL.

- Set CSC of types RSR, LSL, RSL, LSR.

where C denotes a circular arc of radius of r_{min} which can be clockwise or counter-clockwise and thus represented by R or L, respectively. Whereas the straight-line segment is represented in the different configurations by S. A typical method on how to obtain the optimal path graphically can be found in [105] as in Figs. 5.1, 5.2.

In 1991, Héctor Sussman & Guoqing Tang [106] slightly improved Dubins' curves by adding some constraints and it must also be mentioned that similar results were obtained independently and presented almost simultaneously by Boissonnat, Cerezo and Leblond in [107], these constraints are as follows [105]:

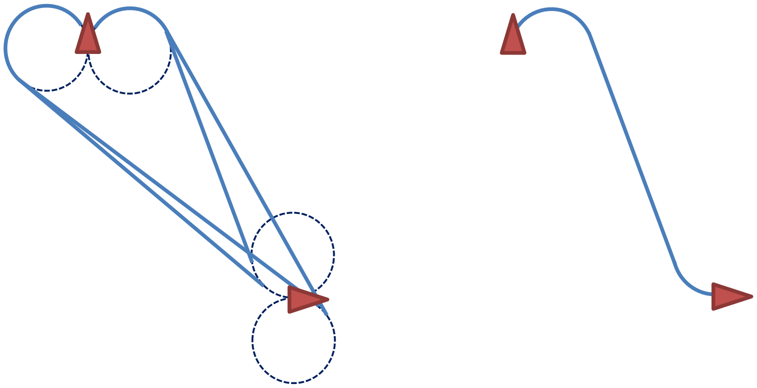


Figure 5.1: An example of obtaining the optimal path for a Dubins' vehicle for CSC configuration

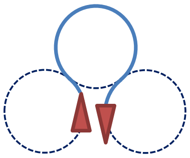


Figure 5.2: An example of obtaining the optimal path for a Dubins' vehicle for CCC configuration

- The set of sufficient family $\mathcal{F}_D$ of optimal paths can be re-written as:
  - $R_aS_bR_c, L_aS_bL_c, R_aS_bL_c, L_aS_bR_c, R_\gamma L_eR_f, L_\gamma R_eL_f$

    such that:
  - $a, c \in [0, 2\pi)$
  - $b \geq 0$
  - $\min\{\gamma, f\} < e - \pi$
  - $\max\{\gamma, f\} < e$

where a, c represent the possible circular arc lengths in CSC configuration, γ, e, f represent the possible circular arc lengths in CCC configuration and b represents the straight-line segment length in CSC configuration.

Several articles introduced different algorithms to obtain a DTSP trajectory such as the Alternating Algorithm (AA) [60, 62, 65], the Two-point Algorithm (2PA), the

Three-point Algorithm (3PA) and the Looking Ahead Algorithm (LAA) [53]. These algorithms only consider solving for the DTSP depending on the ETSP for solving a typical TSP, i.e., the ending heading angle that arrives at the initial point (depot) is not the same as the first departing heading angle from the same point. Thus the DTSP sub-optimal solution obtained from these algorithms are only valid for one tour, starting from a depot and returning back to it, but it is not applicable for the patrolling problem. The AA was presented in [60, 62, 65] with some theoretical derivations for its upper bound, on the other hand, the 2PA, the 3PA, and the LAA were introduced in [53] without any theoretical derivations or representations for their upper bounds. Thus, this chapter first represents the previous mentioned algorithms and introduces theoretical derivations for the 2PA and the 3PA for their upper bounds, followed by proposed enhancements for the 2PA and the AA.

The rest of this chapter is organized as follows. The problem statement for obtaining sub-optimal kinematic-feasible trajectories for patrolling problem applications is introduced in Section 5.2. Section 5.3 presents the existing algorithms in the literature and introduces enhancements for them. Deriving Upper bounds for the 2PA and the 3PA are presented in Section 5.4. A new proposed algorithm named the Pulleys Algorithm (PA) is introduced in Section 5.5.

5.2 Problem Statement

It is required to generate sub-optimal kinematic-feasible trajectories connecting the viewpoints preserving the original order of ETSP optimal solution to be applicable for the patrolling problem using Dubins' vehicles. It is clear from the previous constraints that whenever the euclidean distance between the initial and final configurations is greater than $6r_{min}$, the CCC type of paths are to be disregarded. The reason for this being that the final configuration then clearly is beyond the range of any CCC path, making the $6r_{min}$ a limitation [105]. Now, assuming that r_{min} is much smaller than the euclidean distances which is a convenient and practical assumption for the patrolling problem, the CCC type of trajectories won't be considered in the patrolling problem.

5.3 Existing Algorithms and Enhancements

5.3.1 The 2PA

The 2PA is based on obtaining the optimal DTSP between each two consecutive way-points where the orientation angle at the second way-point is always assumed to be free. This in fact reduces the set family of the optimal trajectories configurations to the following set [53]:

- Set CC_v of types RL, LR with $v > \pi$
- Set CS of types RS, LS

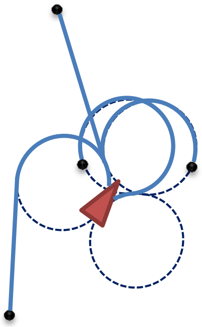


Figure 5.3: Examples of 2PA trajectories for CS and CC_v configurations

Some graphical examples are shown in Fig. 5.3 where the black points represent different possible ending points.

5.3.2 The 3PA

The 3PA is based on obtaining the optimal DTSP between each three consecutive way-points with a free orientation at the third point such that [53]:

- Dubins paths are computed only at every odd point a_{2k-1} where $k = 1, \ldots, (n+1)/2$.
- The solution of the three-point Dubins path from (a_{2k-1},θ_{2k-1}) through a_{2k} to a_{2k+1} yields orientations θ_{2k} and θ_{2k+1}
- Another three-point Dubins path from (a_{2k+1},θ_{2k+1}) is solved to the next two way-points, and repeat the process to cover all the way-points.
- When the number of way-points is even, the last segment will only be a two-point path.
- The trajectories must be in one of these four types: $CSCS, CSCC, CCCS, CCCC$ or their shortest version, but when the way-points are spaced $> 2r_{min}$, the optimal path will be of $CSCS$ type.
- The midpoint bisects the turning arc.

Two graphical examples are shown in Fig. 5.4 that represent two possible formations of the $CSCS$ configuration.

5.3.3 The LAA

The LAA was introduced in [53] and it is based on RHC principles. Based on the 3PA Dubins' path solution, the basic idea of this algorithm is to use this solution to determine only the path and orientation up to the middle way-point. Thus, the solution for (a_1,θ_1), a_2, a_3 is only used to determine θ_2. Note that the choice of θ_2 will be heavily influenced by the location of a_3 in the 3PA Dubins path solution. Once θ_2

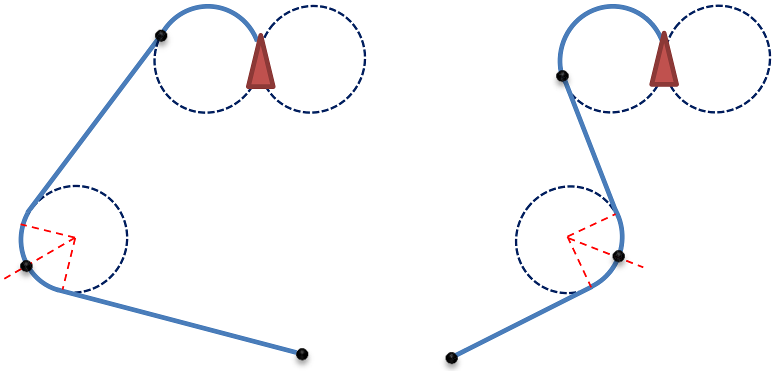


Figure 5.4: Examples of 3PA trajectories for *CSCS* configuration

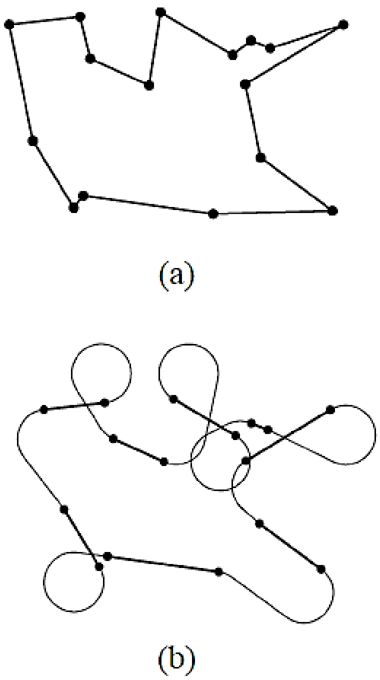
is known, the tour can be extended by solving another 3PA problem starting from (a_2,θ_2). There is no theoretical prespecified graphical representation of the LAA as the LAA optimal trajectory as introduced in in [53], will strongly depend on the performance of the RHC used and consequently on its tuning parameters.

5.3.4 The AA

The AA introduced in [60, 62, 65] depends on keeping the ETSP odd edges as it is while converting the even edges to its corresponding Dubins' curves according to the following AA steps stated in [60, 62, 65]:

Given a set Λ of n points in a compact region $\mathcal{Q} \subset \mathbb{R}^2$, optimally ordered using an ETSP optimal solution as a result of using an ETSP-ALGO(Λ). set A := ETSP-ALGO(Λ)

```
set $\psi_1$ := orientation of the edge from $a_1$ to $a_2$
for $i = 2$ to $n - 1$ do
    if $i$ is even then
        set $\psi_i := \psi_{i-1}$
    else
        set $\psi_i$ := orientation of the edge from $a_i$ to $a_{i+1}$
    end if
end for
if $n$ is even then
    set $\psi_n := \psi_{n-1}$
else
    set $\psi_n :=$ orientation of the edge from $a_n$ to $a_1$
end if
```



(a)

(b)

Figure 5.5: (a) A graph representing the solution of ETSP over a given Λ, (b) A graph representing the solution given by the AA over the same Λ [60, 62, 65]

From the AA it is proved that:

- ETSP(Λ) $\leq$ $\mathrm{L_{AA}}(\Lambda, r_{min}) \leq$ ETSP(Λ) + $k[\frac{n}{2}]\pi r_{min}$, where $k = [2.657,\ 2.658]$
- If there exists $\eta > 0$ such that $\min_{i,j\in\Lambda,\, i\neq j} \mathrm{c}_{ij} \geq \eta r_{min}$, then for $n \geq 3$ $\mathrm{L_{AA}}(\Lambda, r_{min}) \leq (1 + \frac{5k\pi}{6\eta})$ ETSP(Λ)

An important consequence of these results is that, given a point set, for small enough r_{min}, the order of points in the optimal path for the ETSP(Λ) is the same as in the optimal path for the DTSP(Λ, r_{min}).

It is worth to say that the AA with odd number of way-points, the 2PA and the 3PA can be used to obtain a sub-optimal solution for TSP but not for patrolling problem. The reason is that they neglect the necessity that the final edge connecting the last way-point and the initial one should have the same orientation as the initial edge departed from the initial way-point, i.e., the robot can't repeat tracking the same trajectory once more if the patrolling operation is required. Thus, in the previous mentioned algorithms, if they are to be applied to the patrolling problem, the edges arriving at and departing from the initial way-point should have the same orientation. Moreover, no theoretical upper bounds were introduced to the 2PA, the 3PA and the LAA, in addition, the LAA optimal trajectory as introduced, will strongly depend on the performance of the RHC used.

5.3.5 Enhancements for the 2PA and the AA

Before introducing the upper bounds for the 2PA and the 3PA, it is preferably to introduce two enhancements, one for the 2PA and the other for the AA as follows:

- Instead of following the same ETSP(Λ) for the first edge and then applying the 2PA for the rest of the way-points, it would be better if the longest edge in the ETSP(Λ) is followed first with its same length and then apply the 2PA successively for the rest of the way-points.

- Instead of following the same ETSP(Λ) for the first edge and then applying the AA for the rest of the way-points, it would be better if the longest edge in the ETSP(Λ) is followed first with its same length and then apply the AA successively for the rest of the way-points.

Fig. 5.6 shows the effect of these enhancements on both the 2PA and the AA on one simple example of 5 way-points without considering applying them to the patrolling problem. The dashed trajectories represent the optimal DTSP obtained by the algorithms before using the enhancements, whereas the solid trajectories represent the optimal result after using the enhancements.

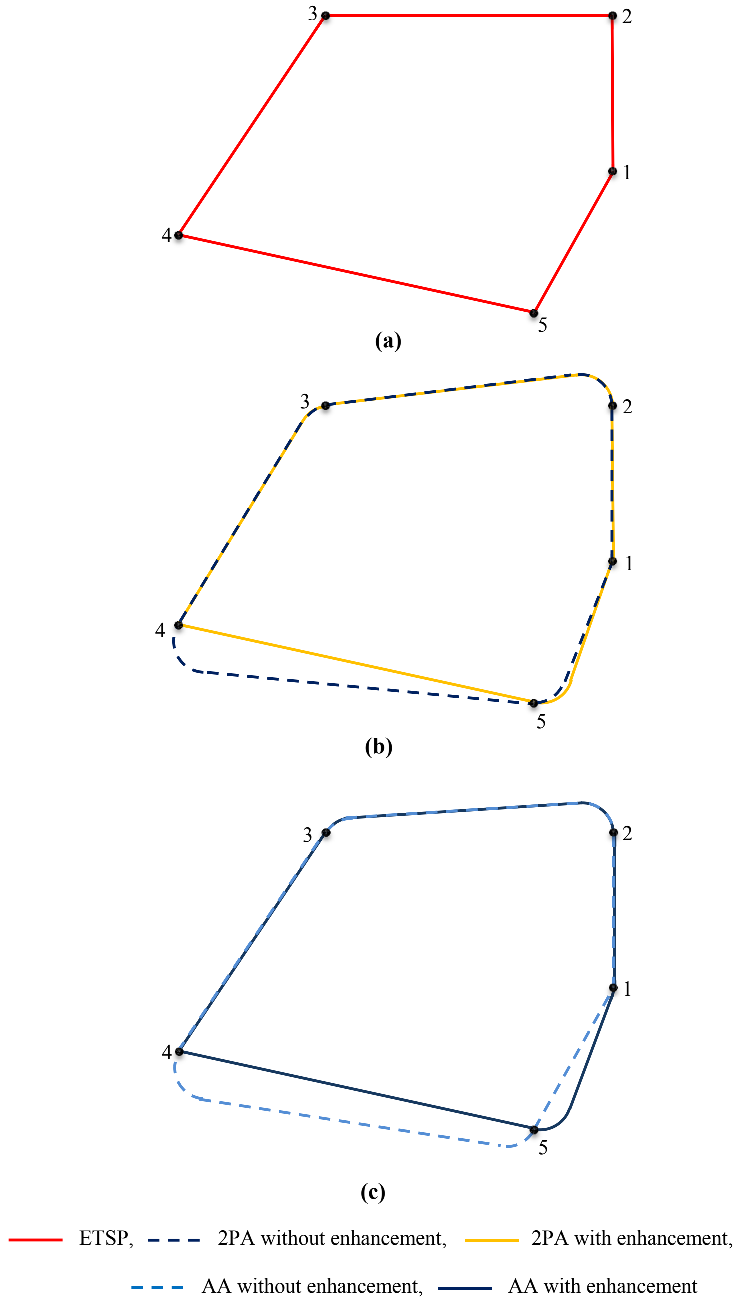


Figure 5.6: (a) A graph representing the solution of ETSP over a given set Λ, (b) A graph representing the solution given by the 2PA over the same set Λ with and without enhancement, (c) A graph representing the solution given by the AA over the same set Λ with with and without enhancement

5.4 Deriving Upper Bounds for the 2PA and the 3PA

- Deriving Upper bounds for the 2PA:
  - For $0 \leq c_{ij}$, $2\text{PA}(\Lambda, r_{min}) < \text{ETSP}(\Lambda) + 2\pi r_{min}$
  - For $r_{min} \leq c_{ij}$, $2\text{PA}(\Lambda, r_{min}) < \text{ETSP}(\Lambda) + \frac{3\pi}{2} r_{min}$
  - For $r_{min} << c_{ij}$, $2\text{PA}(\Lambda, r_{min}) < \text{ETSP}(\Lambda) + \pi r_{min}$

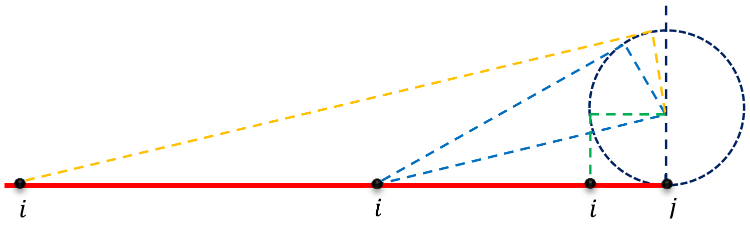


Figure 5.7: A graph representing the upper bound for the 2PA

Proof. As shown in Fig. 5.7, the result of the $\text{ETSP}(\Lambda) = 2c_{ij}$ for $n = 2$ and is represented in red color for different c_{ij} lengths. Since the 2PA result is composed of two lengths, the first is c_{ij}, and the second depends on the Dubins' trajectory from j to i which depends on the departure angle from j. The worst case for this departure angle is 0 so calculating the upper bounds for each possible length of c_{ij} is as follows:

* For $0 \leq c_{ij}$

The distance from j to i will be $2\pi r_{min}$ for very small c_{ij} as represented by the dashed minimum-radius circle. Thus, $2\text{PA}(\Lambda, r_{min}) < \text{ETSP}(\Lambda) + 2\pi r_{min}$.

* For $r_{min} \leq c_{ij}$

The worst scenario in this case where $c_{ij} = r_{min}$, the $\text{ETSP}(\Lambda) = 2r_{min}$ and the distance from j to i will be $\frac{3\pi}{2} r_{min} + r_{min}$ as represented by the green-dashed line. Thus, $2\text{PA}(\Lambda, r_{min}) < \text{ETSP}(\Lambda) + \frac{3\pi}{2} r_{min}$.

* For $r_{min} << c_{ij}$

As c_{ij} increases, the upper bound decreases to be ETSP(Λ) + πr_{min} as represented by the blue-dashed line followed by the yellow-dashed line.

Thus, $2\text{PA}(\Lambda, r_{min}) <$ ETSP(Λ) + πr_{min}.

As a result, for obtaining general upper bounds for $n \geq 3$, the first edge will always be the same as in the ETSP(Λ)optimal solution. Therefore the general upper bounds for n way-points can be as follows:

- For $0 \leq c_{ij}$, $2\text{PA}(\Lambda, r_{min}) <$ ETSP(Λ) + $(n-1)2\pi r_{min}$
- For $r_{min} \leq c_{ij}$, $2\text{PA}(\Lambda, r_{min}) <$ ETSP(Λ) + $(n-1)\frac{3\pi}{2} r_{min}$
- For $r_{min} << c_{ij}$, $2\text{PA}(\Lambda, r_{min}) <$ ETSP(Λ) + $(n-1)\pi r_{min}$

□

- Deriving Upper bounds for the 3PA:

  To derive the upper bounds for the 3PA, it is assumed that at least $2r_{min} \leq c_{ij}$ to guarantee having a minimum-radius circle at the intermediate way-point of any three consecutive way-points. It is also necessary to differentiate between the n edges as each edge can have a different upper bound as follows:

  $n =$ first edge + even edges + odd edges.

  - The upper bound of the first & even edges:
    - For $2r_{min} \leq c_{ij}$, $3\text{PA}(\Lambda, r_{min}) <$ ETSP(Λ) + $(\pi - 2)r_{min}$
    - For $2r_{min} << c_{ij}$, $3\text{PA}(\Lambda, r_{min}) <$ ETSP(Λ) + $(\frac{\pi}{2} - 1)r_{min}$
  - The upper bound of odd edges:
    - For $2r_{min} \leq c_{ij}$, $3\text{PA}(\Lambda, r_{min}) <$ ETSP(Λ) + $2\pi r_{min}$
    - For $2r_{min} << c_{ij}$, $3\text{PA}(\Lambda, r_{min}) <$ ETSP(Λ) + $(\frac{3\pi}{2} - 1)r_{min}$

Proof. First, for the upper bounds of the first & even edges:

As shown in Fig. 5.8, the result of the ETSP(Λ) = c_{ij} and is represented in red color for different c_{ij} lengths. Calculating the upper bounds for each possible length of c_{ij} is as follows:

* For $2r_{min} \leq c_{ij}$

The worst scenario in this case where $c_{ij} = 2r_{min}$, the ETSP(Λ) = $2r_{min}$ and the distance from i to j will be πr_{min} as represented by the blue-dashed circle or the yellow-solid arc. Thus, 3PA(Λ, r_{min}) < ETSP(Λ) + $(\pi - 2)r_{min}$.

* For $2r_{min} << c_{ij}$

As c_{ij} increases, the upper bound decreases to be ETSP(Λ) + $(\frac{\pi}{2} - 1)r_{min}$ as represented by the green-dashed line. Thus, 3PA(Λ, r_{min}) < ETSP(Λ) + $(\frac{\pi}{2} - 1)r_{min}$.

Second, for the upper bounds of the odd edges:

As shown in Fig. 5.9, the result of the ETSP(Λ) = c_{ij} and is represented in red color for different c_{ij} lengths. The corresponding Dubins' trajectory from j to i depends on the departure angle from j where the worst case for this departure angle is 0. Calculating the upper bounds for each possible length of c_{ij} is as follows:

* For $2r_{min} \leq c_{ij}$

The worst scenario in this case where $c_{ij} = 2r_{min}$, the ETSP(Λ) = $2r_{min}$ and the distance from j to i will be $2\pi r_{min} + 2r_{min}$ as represented by the yellow trajectory. Thus, 3PA(Λ, r_{min}) < ETSP(Λ) + $2\pi r_{min}$.

* For $2r_{min} << c_{ij}$

As c_{ij} increases, the upper bound decreases to be ETSP(Λ) + $(\pi r_{min} + \frac{\pi}{2} r_{min})$ as represented by the green trajectory.

Thus, 3PA(Λ, r_{min}) < ETSP(Λ) + $(\frac{3\pi}{2} - 1)r_{min}$. □

Thus the total upper bound of the 3PA solution for n way-points can be summarized in the following table:

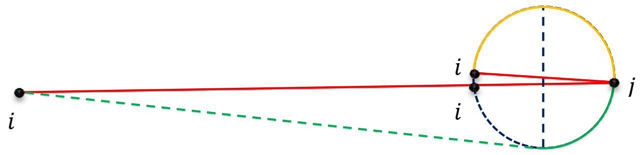


Figure 5.8: The upper bound derivation of the first and even edges in 3PA

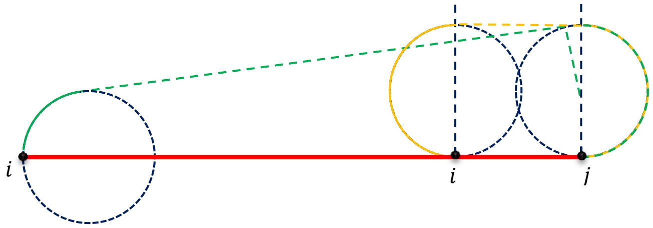


Figure 5.9: The upper bound derivation of the odd edges in 3PA

Table 5.1: Upper bounds for the 3PA

	$2r_{min} \leq c_{ij}$	$2r_{min} << c_{ij}$
n is odd	$3\text{PA}(\Lambda, r_{min}) \leq \text{ETSP}(\Lambda) + (\frac{n+1}{2})(\pi - 2)r_{min} + (\frac{n-1}{2})2\pi r_{min}$ $=$ $3\text{PA}(\Lambda, r_{min}) \leq \text{ETSP}(\Lambda) + n(\frac{3\pi}{2} - 1)r_{min} + (-\frac{\pi}{2} - 1)r_{min}$	$3\text{PA}(\Lambda, r_{min}) \leq \text{ETSP}(\Lambda) + (\frac{n+1}{2})(\frac{\pi}{2} - 1)r_{min} + (\frac{n-1}{2})(\frac{3\pi}{2} - 1)r_{min}$ $=$ $3\text{PA}(\Lambda, r_{min}) \leq \text{ETSP}(\Lambda) + n(\pi - 1)r_{min} - \frac{\pi}{2}r_{min}$
n is even	$3\text{PA}(\Lambda, r_{min}) \leq \text{ETSP}(\Lambda) + (\frac{n+2}{2})(\pi - 2)r_{min} + (\frac{n-2}{2})2\pi r_{min}$ $=$ $3\text{PA}(\Lambda, r_{min}) \leq \text{ETSP}(\Lambda) + n(\frac{3\pi}{2} - 1)r_{min} + (-\pi - 2)r_{min}$	$3\text{PA}(\Lambda, r_{min}) \leq \text{ETSP}(\Lambda) + (\frac{n+2}{2})(\frac{\pi}{2} - 1)r_{min} + (\frac{n-2}{2})(\frac{3\pi}{2} - 1)r_{min}$ $=$ $3\text{PA}(\Lambda, r_{min}) \leq \text{ETSP}(\Lambda) + n(\pi - 1)r_{min} - \pi r_{min}$

5.5 Pulleys Algorithm

In all the previous algorithms, the main key in each algorithm was how to find the proper orientation angle at each way-point. What if this is directly taken from a natural physical system, this system is in fact the system of pulleys. For a given number of viewpoints, the arriving and departing angles at each viewpoint can be considered as in the case of a pulley. A typical pulley with a belt tensioned around it is shown in Fig. 5.10. A new algorithm is introduced with derived upper bounds, named the Pulleys Algorithm (PA) and it is as follows:

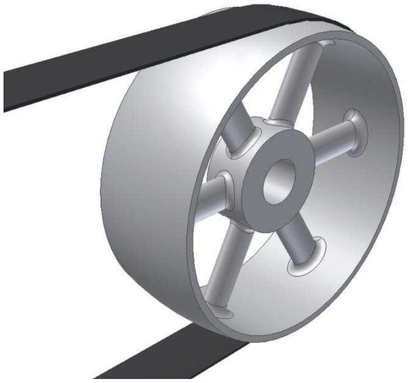


Figure 5.10: A typical pulley with a belt tensioned around it

1- Place a circle of radius r_{min} at each viewpoint such that it intersects the arriving and departing trajectories in two equal chords with the viewpoint in the middle of the arc that starts with the first intersection and ends with the second one.

2- Connect the common tangents of each two consecutive circles with these arcs.

This can be illustrated clearly in a simple 3-viewpoints example as in Fig. 5.11, where the red trajectory represents the ETSP optimal result and the green one represents the proposed PA sub-optimal trajectory. There are some special cases that are discussed later in this chapter while deriving the upper bounds.

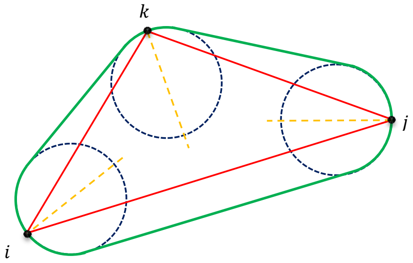


Figure 5.11: A sub-optimal trajectory obtained using the proposed Pulley Algorithm

- Deriving Upper bounds for the PA:
  - For $n = 2$:
    - For $2r_{min} \leq c_{ij}$, $\text{PA}(\Lambda, r_{min}) = \text{ETSP}(\Lambda) + (\pi - 2)r_{min}$

Proof. As shown in Fig. 5.12, the result of the ETSP form i to j is c_{ij} and is represented in a red color for different c_{ij} lengths. The length of the sub-optimal trajectory using the PA in this case is given by ETSP + $2(\frac{\pi}{2} - r_{min})$ as represented by the yellow trajectory.

Thus, PA(Λ, r_{min}) = ETSP(Λ) + $(\pi - 2)r_{min}$. □

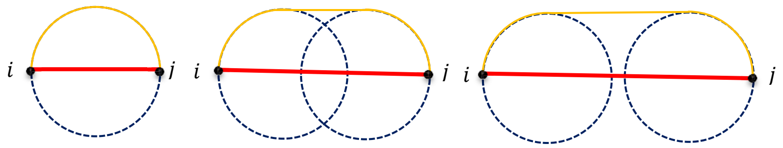


Figure 5.12: A graph for deriving the upper bound of the PA if $2r_{min} \leq c_{ij}$ for $n = 2$

∗ For $r_{min} \leq c_{ij} < 2r_{min}$, PA$(\Lambda, r_{min})$ = ETSP(Λ) + $3\pi r_{min}$ (not applicable for the patrolling problem)

Proof. As shown in Fig. 5.13, the worst scenario for this case is when $c_{ij} = 2r_{min}$. The result of the ETSP form i to j is $2r_{min}$ and is represented in a red color. The length of the sub-optimal trajectory using the PA in this case is given by $2(3\frac{\pi}{2}r_{min} + r_{min})$ as represented by the yellow trajectory.

Thus, PA(Λ, r_{min}) = ETSP(Λ) + $3\pi r_{min}$. □

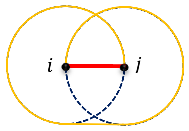


Figure 5.13: A graph for deriving the upper bound of the PA if $r_{min} \leq c_{ij} < 2r_{min}$ for $n = 2$

∗ For $c_{ij} < r_{min}$, PA(Λ, r_{min})= ETSP(Λ) + $(3\pi + 2)r_{min}$ (not applicable for the patrolling problem)

Proof. As shown in Fig. 5.14, the worst scenario for this case is when c_{ij} is too small. The result of the ETSP form i to j is c_{ij} and is represented in a red color. The length of the sub-optimal trajectory using the PA in this case is given by $2(3\frac{\pi}{2}r_{min}+2r_{min})$ as represented by the yellow trajectory.

Thus, PA(Λ, r_{min})= ETSP(Λ) + $(3\pi + 2)r_{min}$. □

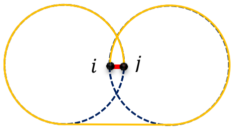


Figure 5.14: A graph for deriving the upper bound of the PA if $c_{ij} < r_{min}$ for $n = 2$

- For $n \geq 3$:
  - ∗ For $2r_{min} \leq c_{ij}$, PA$(\Lambda, r_{min}) \leq$ ETSP(Λ) + $n(\pi - 2)r_{min}$

    Proof. As shown in Fig. 5.15, the result of the ETSP form i to j is c_{ij} and is represented in a red color. From the figure it is clear that:

$$\text{ETSP}(\Lambda) = c_{ik} + c_{kl} + c_{lj} \tag{5.1}$$

where k and l are the two intersecting points between the two perpendiculars from the two centers of the two circles and the link E_{ij} On the other hand the length of the PA sub-optimal result can be calculated as follows.

$$\text{PA}(\Lambda, r_{min}) = \overset{\frown}{im} + \overset{\frown}{mn} + c_{ns} + \overset{\frown}{sj} \tag{5.2}$$

with

$$\overset{\frown}{mn} = \overset{\frown}{rs} \tag{5.3}$$

where m and r are the two intersecting points between the two perpendiculars from the two centers of the two circles and the circumferences of the two circles and n and s are the two external tangential points of the two circles. Eq. 5.2 can then be rewritten as:

$$\mathrm{PA}(\Lambda, r_{min}) = \widehat{im} + c_{ns} + \widehat{rj} \tag{5.4}$$

The worst case then will happen when $c_{ij} = 2r_{min}$, then the upper bound of the PA trajectory which is represented by the yellow trajectory will be the same as the corresponding case for $n = 2$. Thus, $\mathrm{PA}(\Lambda, r_{min}) \leq \mathrm{ETSP}(\Lambda) + n(\pi - 2)r_{min}$. □

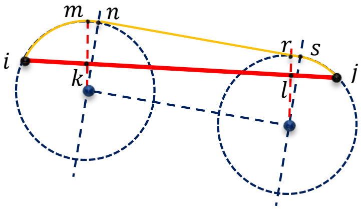


Figure 5.15: A graph for deriving the upper bound of the PA if $2r_{min} \leq c_{ij}$ for $n \geq 3$

∗ For $c_{ij} < 2r_{min}$ (not applicable for the patrolling problem)

- If c_{ij} intersects the two circles in two chords,

  $\mathrm{PA}(\Lambda, r_{min}) \leq \mathrm{ETSP}(\Lambda) + n(\pi - 2)r_{min}$

  As shown in Fig. 5.16, the same upper bound will be applied as for the previous case with the same proof.

- If c_{ij} intersects the two circles in one chord only,

  $\mathrm{PA}(\Lambda, r_{min}) \leq \mathrm{ETSP}(\Lambda) + 3n\pi r_{min}$

  Proof. As shown in Fig. 5.17, the result of the ETSP form i to j is c_{ij} and is represented in a red color. From the figure it is clear

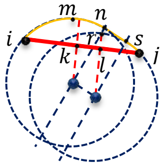


Figure 5.16: A graph for deriving the upper bound of the PA if $c_{ij} < 2r_{min}$ for $n \geq 3$ and c_{ij} intersects the two circles in two chords

that:

$$\text{ETSP}(\Lambda) = c_{ik} + c_{kl} + c_{lj} \tag{5.5}$$

where k and l are as defined previously. On the other hand the length of the PA sub-optimal result can be calculated as follows.

$$\text{PA}(\Lambda, r_{min}) = \widehat{im} + \widehat{mn} + c_{ns} + \widehat{sr} + \widehat{rj} \tag{5.6}$$

with

$$\widehat{mn} = \widehat{rs} \tag{5.7}$$

where m and r are as defined before. Eq. 5.6 can then be rewritten as:

$$\text{PA}(\Lambda, r_{min}) = \widehat{im} + c_{ns} + \widehat{rj} + 2\pi r_{min} \tag{5.8}$$

The worst case then will happen when c_{ij} is too small which will lead to too small c_{ns} and the upper bound of the PA trajectory which is represented by the yellow color can be formulated as: $\text{PA}(\Lambda, r_{min}) \leq \text{ETSP}(\Lambda) + 3n\pi r_{min}$. □

- Special cases:
  - The external tangent of the two circles touches one of them after the viewpoint position:

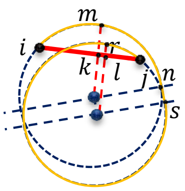


Figure 5.17: A graph for deriving the upper bound of the PA if $c_{ij} < 2r_{min}$ for $n \geq 3$ and c_{ij} intersects the two circles in one chord only

$*$ For $2r_{min} \leq c_{ij}$, $\mathrm{PA}(\Lambda, r_{min}) \leq \mathrm{ETSP}(\Lambda) + n(\frac{5\pi}{2} + \sqrt{2} - 2)r_{min}$

i.e.,

For $2r_{min} \leq c_{ij}$, $\mathrm{PA}(\Lambda, r_{min}) \leq \mathrm{ETSP}(\Lambda) + 7.268\ nr_{min}$

To decrease the impact of this case, it is assumed that $6r_{min} \leq c_{ij}$ which not only prevents the formation of a CCC type trajectory but also decreases the possible difference between the actual location of the viewpoint and the tangential point to 0.2 r_{min} which is convenient for the patrolling operation.

Proof. To find out the worst scenario that can happen for this case when $2r_{min} \leq c_{ij}$ it is required to search for the longest possible distance that can occur between the missed tangential point and the exact location of the way-point, this can be achieved if the link (i, j) is considered to be a diameter in one circle and is tangential to the other circle at one of its terminal points as illustrated in Fig. 5.18. It is clear from the geometry of the figure that maximum length between the viewpoint i and the external tangential point n which is represented in the figure by green-dashed line is:

$$c_{ni} = 2r_{min}\cos(135/2) \tag{5.9}$$

$$c_{ni} = 0.765r_{min} \tag{5.10}$$

Thus, ETSP from i to j is $2r_{min}$, whereas the length of the sub-optimal result of PA is given by:

$$\mathrm{PA}(\Lambda, r_{min}) \leq \overset{\frown}{iy} + \overset{\frown}{yn} + c_{ns} + \overset{\frown}{sj} \tag{5.11}$$

with

$$\overset{\frown}{iy} = \overset{\frown}{jx} \tag{5.12}$$

Eq. 5.12 can then be rewritten as:

$$\mathrm{PA}(\Lambda, r_{min}) \leq (3\frac{\pi}{2}r_{min} + \pi r_{min} + \sqrt{2}r_{min}) \tag{5.13}$$

i.e.,

$$\mathrm{PA}(\Lambda, r_{min}) \leq \mathrm{ETSP}(\Lambda) + 7.268nr_{min} \tag{5.14}$$

□

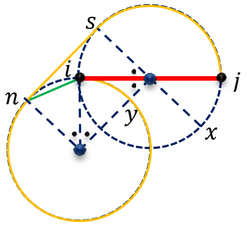


Figure 5.18: A graph for the worst special case where the external tangential point comes before the viewpoint

To calculate a general formulation for this case, as represented in Fig.5.19, the difference between the tangential point and the actual location of the view-point c_{ni} decreases as the distance c_{ij} increases. The difference can be clearly seen from Fig. 5.19 to be the length between the external tangential point and the tangential point caused by the edge (i, j) itself to the same circle when it is taken to be the diameter inthe second circle.

The relation between c_{ni} and c_{ij} can be formulated as:

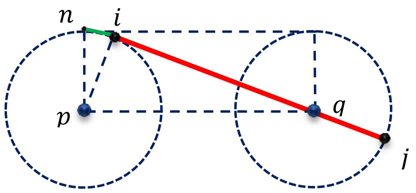


Figure 5.19: A graph for derivation of a relation between the special case difference and c_{ij}

$$c_{ni} = 2r_{min}\cos(90 - \frac{\sin^{-1}(\frac{r_{min}}{c_{pq}})}{2}) \tag{5.15}$$

Thus, for $6r_{min} \leq c_{ij}$, $c_{ni} = 0.2\ r_{min}$.

- The formation of reflex angles:
  - For $4r_{min} \leq\ c_{ij}$, PA $\leq$ ETSP(Λ) + $n(2\pi - 4)r_{min}$
  - For $4r_{min} <<\ c_{ij}$, PA $\leq$ ETSP(Λ) + $n(\pi - 2)r_{min}$

Proof. It is clear from Fig. 5.20 that for the occurrence of reflex angles among the trajectory that connects the viewpoints, as c_{ij} which is represented by red color increases, the upper bound decreases relative to r_{min}.

For the case $4r_{min} \leq\ c_{ij}$, the worst case is to consider as if $n = 2$, and $c_{ij} \leq 4r_{min}$, thus the upper bound will be

For $4r_{min} \leq\ c_{ij}$, PA $\leq$ ETSP(Λ) + $n(2\pi - 4)r_{min}$.

Similarly, As c_{ij} increases the upper bound decreases as shown in Fig. 5.21. Thus, For $4r_{min} <<\ c_{ij}$, PA $\leq$ ETSP(Λ) + $n(\pi - 2)r_{min}$.

Finally, for patrolling operations, assuming $6r_{min} \leq\ c_{ij}$, the PA can lead to an upper bound of :

- PA$(\Lambda, r_{min}) \leq$ ETSP(Λ) + $n(\pi - 2)r_{min}$ if no reflex angle is included.

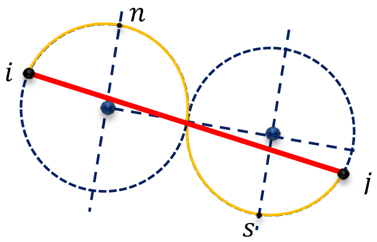


Figure 5.20: A graph for derivation of the reflex angle case when $c_{ij} \leq 4r_{min}$

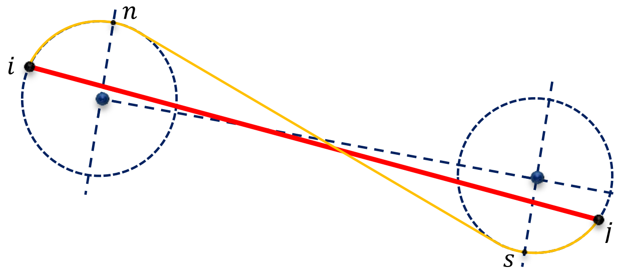


Figure 5.21: A graph for derivation of the reflex angle case when $c_{ij} << 4r_{min}$

- $\text{PA}(\Lambda, r_{min}) \leq \text{ETSP}(\Lambda) + n(2\pi - 4)r_{min}$ if any reflex angle is included.

It is worth to state that for typical patrolling operations, $6r_{min} << c_{ij}$, and thus, the PA can lead to a general upper bound of :

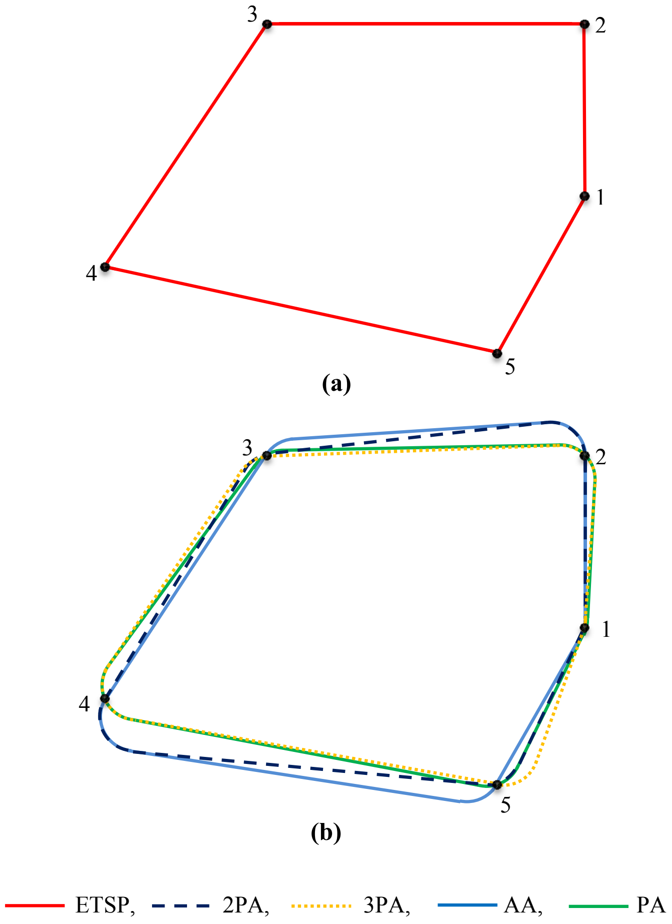
- $\text{PA}(\Lambda, r_{min}) \leq \text{ETSP}(\Lambda) + n(\pi - 2)r_{min}$

□

Fig. 5.22 represents a graphical comparison among all the previous mentioned algorithms using an example of 5-viewpoints configuration to illustrate the efficacy of using the PA.

As illustrated in Fig. 5.22, it is clear that the PA not only achieved less upper bound than the other algorithms but also it achieved the less maximum deviation from the ETSP(Λ) trajectory among all the optimal solutions of the other algorithms.

The next chapter introduces the experimental work of this research, MPC controller is applied on the linearized model of the TWMR to feasibly-achieve tracking



(a)

(b)

Figure 5.22: (a) A graph representing the solution of ETSP over a given set Λ, (b) A graphical comparison among the 2PA, the 3PA, the AA, and the PA for the same set Λ

the obtained trajectories, either the ETSP optimal trajectories or the Dubins' trajectories.

Chapter 6

Experimental Work

In this chapter it is desired to design a controller for the TWMR and the Dubins' vehicles to track the ETSP optimal and DTSP sub-optimal trajectories for a specific setting. A model predictive control (MPC) scheme is used for trajectory tracking, and to reduce its computational complexity, the nonlinear model of the TWMR is linearized accordingly. Three main experiments are performed. The purpose of the first two experiments is to track the exact sharp-turning optimal patrolling trajectories characterized in Chapters 2-4. The third experiment, on the other hand, uses the sub-optimal trajectories presented in Chapter 5 by softening the optimal sharp-turning patrolling trajectories based on Dubins' vehicles.

The rest of this chapter is organized as follows. The LMPC for the TWMR is first introduced in Section 6.1. Section 6.2 presents a description of the testbed used in the experimental work including the TWMR, the camera tracking system and the control software. Finally, the results are represented in Section 6.3.

6.1 LMPC for TWMR

As the first step in controller design, it is important to study the kinematics of the TWMR. Fig. 6.1 shows a typical TWMR with a third caster wheel which adds stability to the system. The kinematic equations of this mobile robot are given by:

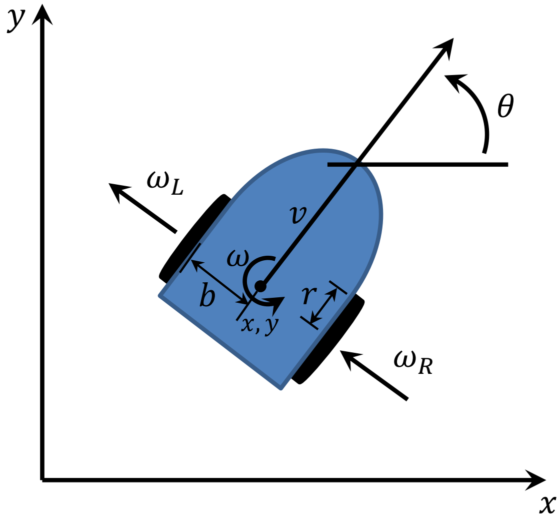


Figure 6.1: A TWMR with its kinematic model parameters.

$$\dot{x} = v cos\theta \tag{6.1}$$

$$\dot{y} = v sin\theta \tag{6.2}$$

$$\dot{\theta} = \omega \tag{6.3}$$

where x, y represent the coordinates of the center of the axle of the robot w.r.t. the global coordinate frame, b denotes half the distance between the two wheels, r denotes the radius of the wheel, and $\dot{x}$, $\dot{y}$ are the components of the linear velocity v in the x and y directions, respectively. Furthermore, θ is the heading of the robot w.r.t. the x axis in the global coordinate frame, ω is the angular velocity of the robot, ω_L and ω_R are the angular velocities of the left and right wheels, respectively. Assuming that the wheel does not slip in the lateral direction, the following nonholonomic constraint is imposed:

$$\dot{x}\ sin\theta = \dot{y}\ cos\theta \tag{6.4}$$

The physical significance of the nonholonomic constraints is that there is no possible movement in the axial direction. In other words, the direction of the translational velocity is the direction of the tangent to the path [56].

The trajectory tracking problem involves a reference robot as shown in Fig. 6.2, where all kinematic constraints are implicitly expressed by the reference trajectory. The kinematic model of the TWMR given by (6.1-6.3) can be provided in the form presented in [82, 83, 85] as follows:

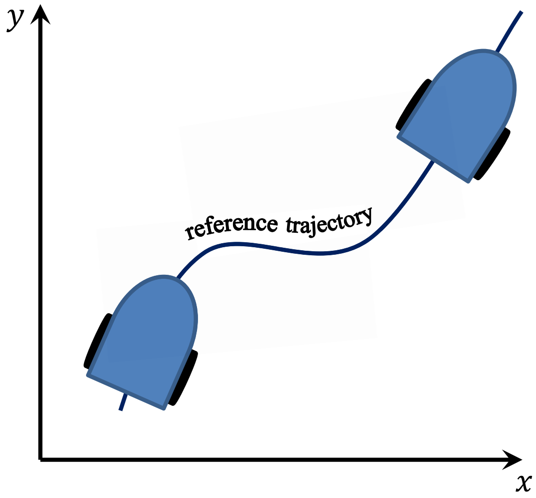


Figure 6.2: Trajectory tracking of a TWMR

$$\dot{\mathrm{x}} = f(\mathrm{x}, \mathrm{u}) \tag{6.5}$$

where $\mathrm{x} \triangleq [x\, y\, \theta]^T$, and $\mathrm{u} \triangleq [v\, \omega]^T$ is the control input. Now, consider the reference trajectory generated by a reference robot as follows:

$$\dot{\mathrm{x}}_r = f(\mathrm{x}_r, \mathrm{u}_r) \tag{6.6}$$

Expanding the right side of 6.5 using Taylor series around $(\mathrm{x}_r, \mathrm{u}_r)$ and neglecting the second-and higher-order terms yields:

$$\dot{\mathrm{x}} = \; f(\mathrm{x}_r, \mathrm{u}_r) \; + \; \frac{\partial f(\mathrm{x}, \mathrm{u})}{\partial \mathrm{x}}\bigg|_{\substack{\mathrm{x}=\mathrm{x}_r \\ \mathrm{u}=\mathrm{u}_r}} (\mathrm{x} - \mathrm{x}_r) + \quad \frac{\partial f(\mathrm{x}, \mathrm{u})}{\partial \mathrm{u}}\bigg|_{\substack{\mathrm{x}=\mathrm{x}_r \\ \mathrm{u}=\mathrm{u}_r}} (\mathrm{u} - \mathrm{u}_r) \tag{6.7}$$

or simply:

$$\dot{\mathrm{x}} = \; f(\mathrm{x}_r, \mathrm{u}_r) \; + \; f_{x_r}(\mathrm{x} - \mathrm{x}_r) + f_{u_r}(\mathrm{u} - \mathrm{u}_r) \tag{6.8}$$

where f_{x_r}, f_{u_r} are the Jacobians of f w.r.t. x and u, respectively, evaluated around the reference point $(\mathrm{x}_r, \mathrm{u}_r)$. Let $\tilde{\mathrm{x}} \triangleq \mathrm{x}\text{-}\mathrm{x}_r$ represent the error w.r.t. the reference robot, and $\tilde{\mathrm{u}} \triangleq \mathrm{u}\text{-}\mathrm{u}_r$ be the perturbation control input for $\tilde{\mathrm{x}}$. One can write:

$$\dot{\tilde{\mathrm{x}}} = f_{x_r}\tilde{\mathrm{x}} + f_{u_r}\tilde{\mathrm{u}} \tag{6.9}$$

Approximating $\dot{\tilde{\mathrm{x}}}$ using forward differences leads to the following discrete-time model:

$$\tilde{\mathrm{x}}(k+1) = \mathrm{A}(k)\tilde{\mathrm{x}}(k) + \mathrm{B}(k)\tilde{\mathrm{u}}(k) \tag{6.10}$$

with

$$\mathrm{A}(k) \triangleq \begin{bmatrix} 1 & 0 & -v_r(k)\sin\theta_r(k)T \\ 0 & 1 & v_r(k)\cos\theta_r(k)T \\ 0 & 0 & 1 \end{bmatrix} \tag{6.11}$$

,

$$\mathrm{B}(k) \triangleq \begin{bmatrix} \cos\theta_r(k)T & 0 \\ \sin\theta_r(k)T & 0 \\ 0 & T \end{bmatrix} \tag{6.12}$$

where T is the sampling time and k is the sampling number.

As shown in Fig. 6.3, the essence of any MPC scheme is to optimize predictions of system variables over a sequence of future control inputs. Such a prediction is accomplished by using a process model over a finite time interval, called the prediction horizon. At each sampling time, the MPC attempts to generate the best possible control sequence by solving an optimization problem. The first element of this sequence is applied to the system, and then the problem is solved again at the next sampling time using the updated process measurements and a shifted horizon [82, 83, 108]. This procedure continues as the new samples are generated.

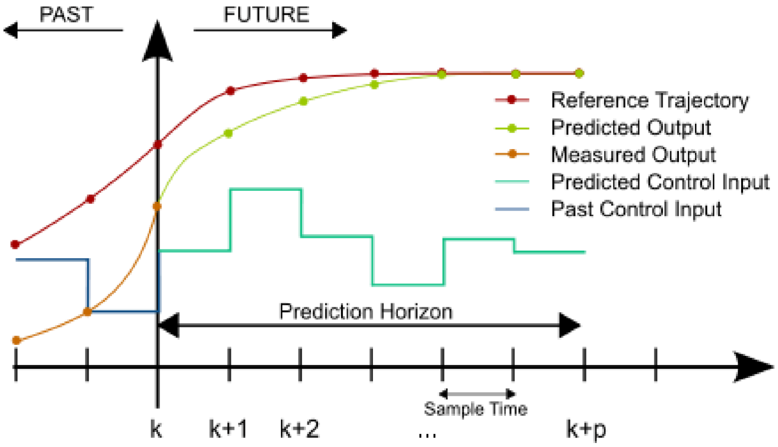


Figure 6.3: A basic working principle of Model Predictive Control [109]

The MPC minimizes an objective function which can be introduced here as follows:

$$\Phi(k) = \sum_{j=1}^{N} \tilde{\mathrm{x}}^T(k+j|k)\mathrm{Q}\tilde{\mathrm{x}}(k+j|k) + \tilde{\mathrm{u}}^T(k+j-1|k)\mathrm{R}\tilde{\mathrm{u}}(k+j-1|k) \tag{6.13}$$

where N is the prediction horizon, Q and R are properly chosen positive definite and positive semi-definite weighting matrices, respectively. The notation a$(j|k)$ is the value of a at the instant j predicted at an earlier instant k. Hence, the optimization problem is to find $\tilde{\mathrm{u}}^*$ such that:

$$\tilde{\mathrm{u}}^* = \min_{\tilde{\mathrm{u}}}\{\Phi(k)\} \tag{6.14}$$

The problem of minimizing (6.13) is solved at each time step k, which yields a sequence of optimal control signals $\{\tilde{\mathrm{u}}^*(k|k), \ldots, \tilde{\mathrm{u}}^*(k+N-1|k)\ \}$ as well as the optimal cost $\Phi^*(k)$. The MPC control law is implicitly given by the first control action of $\tilde{\mathrm{u}}^*(k|k)$. A block diagram of the system is shown in Fig. 6.4. Moreover,

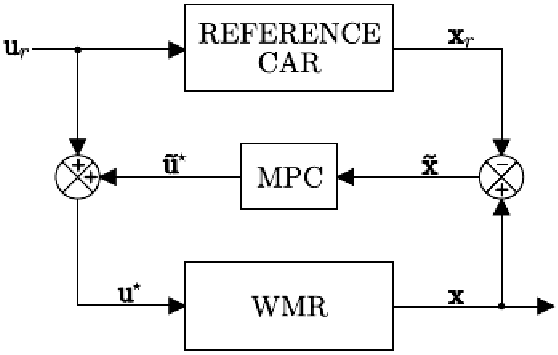


Figure 6.4: The block diagram of an LMPC [82, 83]

(6.13) can be rewritten as follows:

$$\Phi(k) = \bar{\mathrm{x}}^T(k+1)\bar{\mathrm{Q}}\bar{\mathrm{x}}(k+1) + \bar{\mathrm{u}}^T(k)\bar{\mathrm{R}}\bar{\mathrm{u}}(k) \tag{6.15}$$

where:

$$\bar{\mathrm{x}}(k+1) \triangleq \begin{bmatrix} \tilde{\mathrm{x}}(k+1|k) \\ \tilde{\mathrm{x}}(k+2|k) \\ \vdots \\ \tilde{\mathrm{x}}(k+N|k) \end{bmatrix} \tag{6.16}$$

$$\bar{\mathrm{u}}(k) \triangleq \begin{bmatrix} \tilde{\mathrm{u}}(k|k) \\ \tilde{\mathrm{u}}(k+1|k) \\ \vdots \\ \tilde{\mathrm{u}}(k+N-1|k) \end{bmatrix} \tag{6.17}$$

$$\bar{\mathrm{Q}} \triangleq \begin{bmatrix} \mathrm{Q} & 0 & \dots & 0 \\ 0 & \mathrm{Q} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathrm{Q} \end{bmatrix} \tag{6.18}$$

$$\bar{\mathrm{R}} \triangleq \begin{bmatrix} \mathrm{R} & 0 & \dots & 0 \\ 0 & \mathrm{R} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathrm{R} \end{bmatrix} \tag{6.19}$$

This leads to the following equation:

$$\bar{\mathrm{x}}(k+1) = \bar{\mathrm{A}}(k)\tilde{\mathrm{x}}(k|k) + \bar{\mathrm{B}}(k)\tilde{\mathrm{u}}(k) \tag{6.20}$$

with

$$\bar{\mathrm{A}}(k) \triangleq \begin{bmatrix} \mathrm{A}(k|k) \\ \mathrm{A}(k|k)\mathrm{A}(k+1|k) \\ \vdots \\ \alpha(k,0) \end{bmatrix} \tag{6.21}$$

$$\bar{\mathrm{B}}(k) \triangleq \begin{bmatrix} \mathrm{B}(k|k) & 0 & \dots & 0 \\ \mathrm{A}(k+1|k)\mathrm{B}(k|k) & \mathrm{B}(k+1|k) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha(k,0)\mathrm{B}(k|k) & \alpha(k,2)\mathrm{B}(k+1|k) & \dots & \mathrm{B}(k+N-1|k) \end{bmatrix} \tag{6.22}$$

where $\alpha(k, j)$ is defined as:

$$\alpha(k,j) \triangleq \prod_{i=j}^{N-1} \mathrm{A}(k+i|k) \tag{6.23}$$

Thus, the objective function (6.13) can be rewritten in a standard quadratic form as follows:

$$\Phi(k) = \frac{1}{2}\bar{\mathrm{u}}^T(k)\mathrm{H}(k)\bar{\mathrm{u}}(k) + \mathrm{f}^T\bar{\mathrm{u}}(k) + \mathrm{d}(k) \tag{6.24}$$

with

$$\mathrm{H}(k) \triangleq 2(\bar{\mathrm{B}}^T(k)\bar{\mathrm{Q}}(k)\bar{\mathrm{B}}(k) + \bar{\mathrm{R}}) \tag{6.25}$$

$$\mathrm{f}(k) \triangleq 2\bar{\mathrm{B}}^T(k)\bar{\mathrm{Q}}(k)\bar{\mathrm{A}}(k)\tilde{\mathrm{x}}(k|k) \tag{6.26}$$

$$\mathrm{d}(k) \triangleq \tilde{\mathrm{x}}^T(k|k)\bar{\mathrm{A}}^T(k)\bar{\mathrm{Q}}(k)\bar{\mathrm{A}}(k)\tilde{\mathrm{x}}(k|k) \tag{6.27}$$

Note that $\mathrm{H}(k)$ is a *Hessian* matrix, and hence positive definite. It describes the quadratic part of the objective function, while the vector $\mathrm{f}(k)$ describes the linear part. Note also that $\mathrm{d}(k)$ is independent of $\tilde{\mathrm{u}}$ and has no impact on u^*.

The control inputs in a real world system are subject to physical limitations. On the other hand, due to model uncertainties there is always a mismatch between the plant dynamics and the equations describing them. These practical limitations and uncertainties should be taken into consideration in the development of the control inputs. For example, as far as control limitations are concerned, one can define some upper and lower bounds on the magnitude of the control input. The optimization problem should then be solved while ensuring that the control will remain between certain lower and upper bounds [82, 83]. Since the free variable in the underlying optimization problem is ũ, the control constraint can be described by:

$$\tilde{u}_{min}(k) \leq \tilde{u}(k) \leq \tilde{u}_{max}(k) \tag{6.28}$$

Thus, Eq. (6.15) can be rewritten as:

$$\tilde{u}^{*} = \min_{\tilde{u}}\{\Phi(k)\} \tag{6.29}$$

with the constraint:

$$\begin{bmatrix} I \\ -I \end{bmatrix} \tilde{u} \leq \begin{bmatrix} \tilde{u}_{max} \\ -\tilde{u}_{min} \end{bmatrix} \tag{6.30}$$

6.2 Testbed Description

The Quanser unmanned ground vehicle (QGV) is a testbed shown in Fig. 6.5, designed and manufactured by Quanser Inc. [110], and is available at the Network Autonomous Vehicle (NAV) lab in the Department of Mechanical and Industrial Engineering of Concordia University. It represents an innovative vehicle platform suitable for a wide variety of unmanned ground vehicle (UGV) applications. The QGV is a differential drive ground robot which has a 4 DOF robotic manipulator with a gripper. To measure on-board sensors and drive the motors, the QGV utilizes Quanser's embedded data acquisition card HiQ DAQ, and the embedded Gumstix computer. The HiQ DAQ is a high-resolution input/output (I/O) card designed to accommodate a wide variety of research problems. The Quanser's real-time control software (QUARC) allows researchers and developers to develop and test controllers on actual hardware through a MATLAB Simulink interface. QUARC can target the Gumstix embedded computer, automatically generating code and executing controllers on-board the vehicle. During the operation of the system, while the controller runs on the Gumstix, users can tune parameters in real-time and observe sensor measurements from a host ground station computer (PC or laptop). The interface to the QGV is MATLAB Simulink with QUARC. The controllers are developed in Simulink with QUARC on the host computer, and these models are downloaded and compiled to be executable on the target (Gumstix) seamlessly. A diagram of this framework is shown in Fig. 6.6. Each robot needs to be energized using two 3-cell 11.1 v, 2500 mAh Lithium-Polymer batteries. The tracking system is equipped with 24 OptiTrack cameras shown in Fig. 6.7, which can offer integrated image capture, processing, and motion tracking.

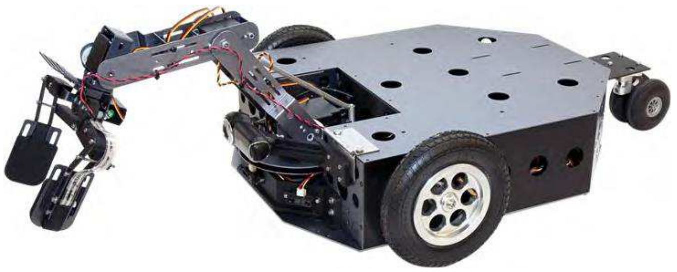


Figure 6.5: The Quansar QGV [110]

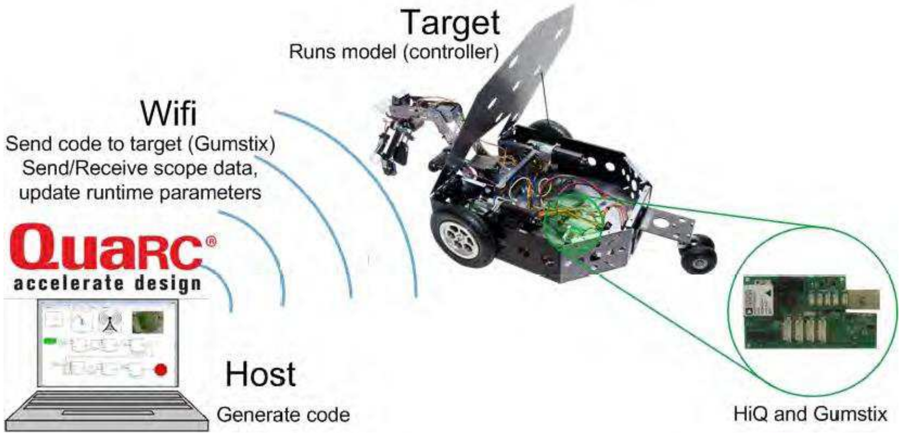


Figure 6.6: The configuration framework used in the experiments [110]

Figure 6.7: An OptiTrack camera [110]

6.3 Experimental Results

Three experiments are performed here. In the first experiment, it is aimed to follow the exact sharp-turning minimum-distance patrolling trajectories obtained in Chapters 2-4 for the TWMRs. One robot is used to patrol a prescribed 2m by 2m area starting from some depot. By applying the proposed LMPC to a QGV to patrol the area with 6 viewpoints the results depicted in Fig. 6.8 are obtained. The parameters used in the optimization problem are Q $=$ *diag* (30,30,8), R $=1$ $\mathbf{I}_{2X2}$, N $=$ 5, x(0) =[0 -1.5 $\pi/4]^T$, and the constraints on the magnitude of the control variables are $v_{min} = -0.5m/s, v_{max} = 0.5m/s$, $\omega_{min} = -1rad/s$, and $\omega_{max} = 1rad/s$. Fig. 6.9 shows the errors in the x, y directions as well as in the angle θ, and Fig. 6.10 gives the control inputs for the robot.

The second experiment represents the implementation of the exact sharp-turning patrolling trajectories optimally-obtained in Chapters 2-4 which represent the minimum-time optimal trajectories using the TWMR dynamics. Fig. 6.11 represents the experimental result of applying LMPC on two QGV to patrol an area of size 2m by 2m, where a set of $n = 7$ viewpoints are to be visited. Two robots are used to track the minimum time optimal trajectories starting from two depots, where the design parameters for the first robot are : Q $=$ *diag* (30,30,8), R $=1$ $\mathbf{I}_{2X2}$, N $=$ 5, x(0) =[0 -2 $\pi/6]^T$. Whereas the design parameters for the second robot are: Q $=$

$diag$ (30,30,5), R = 1 $\mathbf{I}_{2X2}$, N = 5, x(0) =[-1 1.5 -π/6]^T. the constraints on the amplitude of the control variables for both robots are: $v_{min} = -0.5m/s, v_{max} = 0.5m/s$, $\omega_{min} = -1rad/s$, and $\omega_{max} = 1rad/s$. Fig. 6.12 shows the errors in the x, y and θ components for both robots and Fig. 6.13 shows the corresponding robots' control inputs.

The third experiment represents the implementation of sub-optimal patrolling trajectories by fulfilling the Pulleys Algorithm introduced in Chapter 5 for softening the optimally-obtained sharp-turning patrolling trajectories into sub-optimal trajectories that can be tracked by Dubins' vehicles. Fig. 6.14 represents the experimental result of applying LMPC on two QGV to patrol an area of size 2m by 2m, where the same set of $n = 7$ viewpoints are to be visited but this time after applying the Pulleys Algorithm introduces in Chapter 5, so the robots are not going to stop and turn on the spot but will keep moving with constant velocity and with minimum turning radius r_{min}.Two robots are used to track the minimum time sub-optimal trajectories starting from two depots, where the design parameters for the first robot are : Q = $diag$ (5,5,12), R = 1 $\mathbf{I}_{2X2}$, N = 5, x(0) =[0 -2 1.75π]^T. Whereas the design parameters for the second robot are: Q = $diag$ (6,6,11), R = 1 $\mathbf{I}_{2X2}$, N = 5, x(0) =[-1 1.5 0.1π]^T. the constraints on the amplitude of the control variables for both robots are: $v_{min} = -0.5m/s, v_{max} = 0.5m/s$, $\omega_{min} = -1rad/s$, and $\omega_{max} = 1rad/s$. Fig. 6.15 shows the errors in the x, y and θ components for both robots and Fig. 6.16 shows the corresponding robots' control inputs.

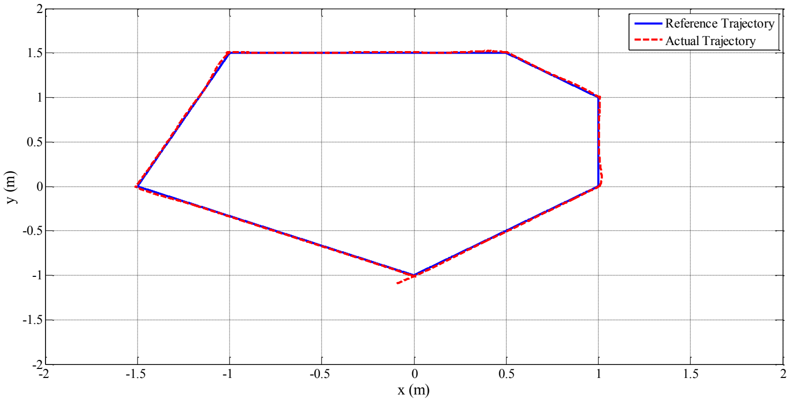


Figure 6.8: Experimental result for optimal patrolling trajectory using LMPC

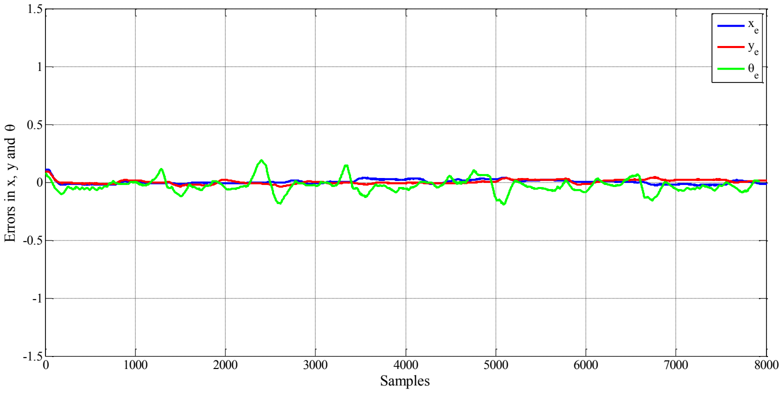


Figure 6.9: The errors in x, y and θ

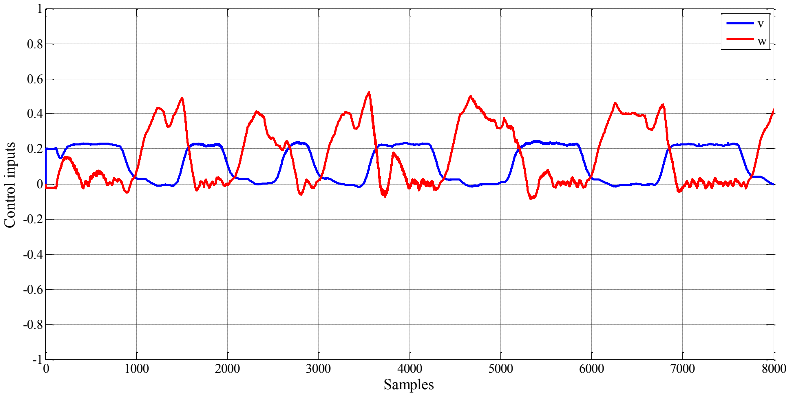


Figure 6.10: Control inputs v and ω

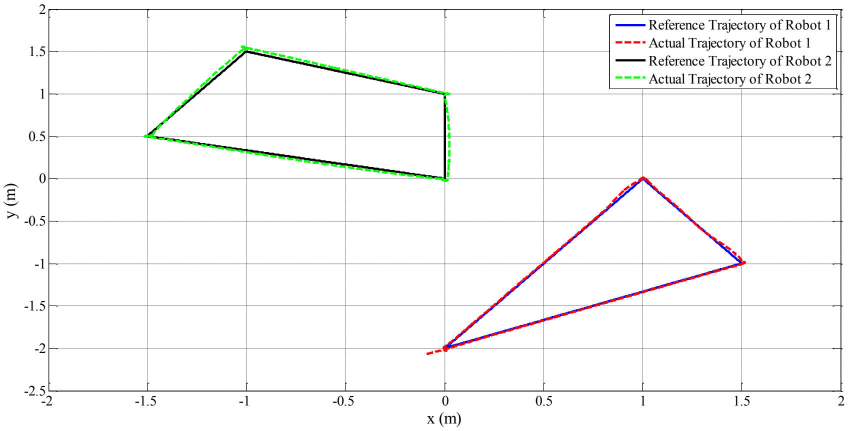


Figure 6.11: Experimental result for optimal patrolling trajectory using LMPC for two robots

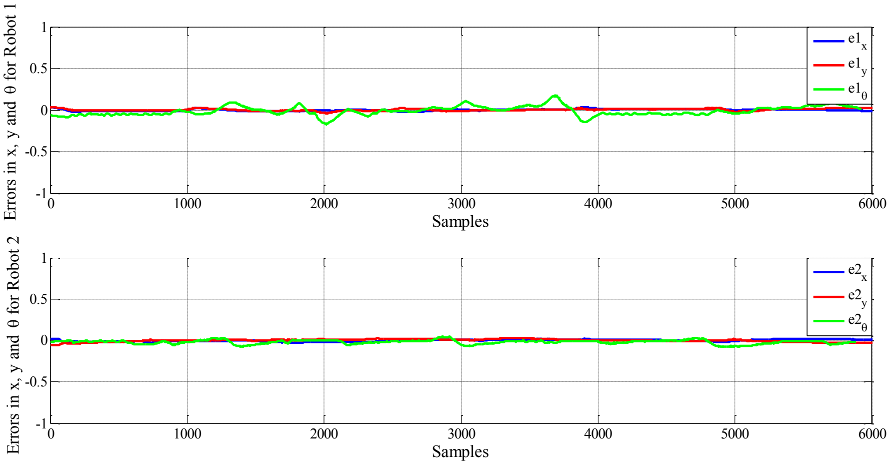


Figure 6.12: The errors in x, y and θ for the two robots

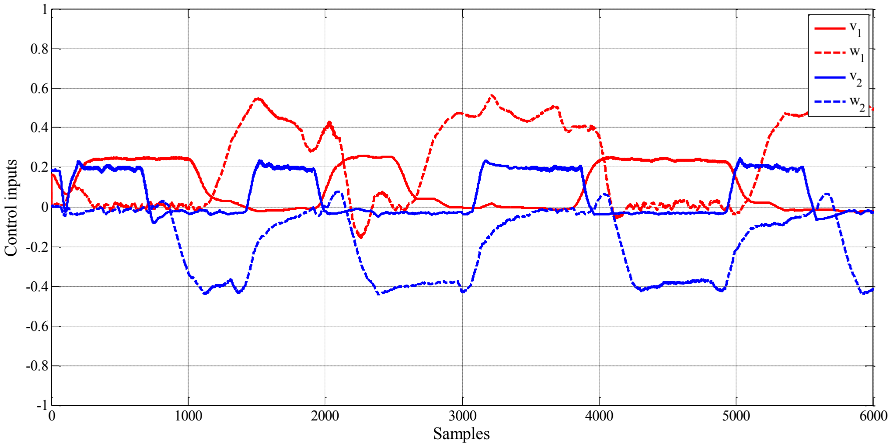


Figure 6.13: Control inputs v and ω for the two robots

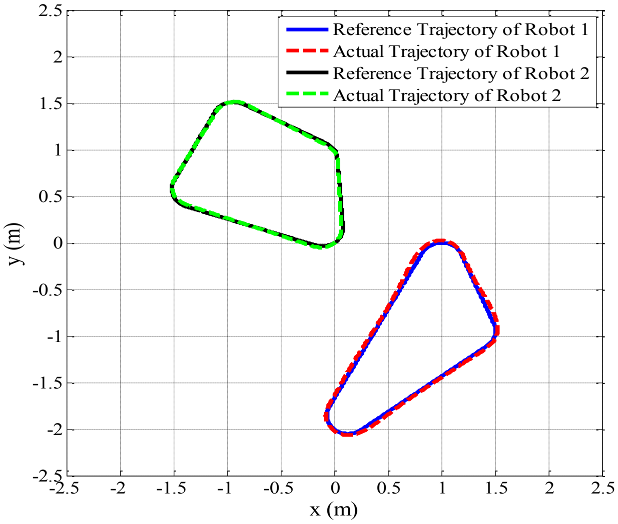


Figure 6.14: Experimental result for sub-optimal patrolling trajectory using LMPC for two robots

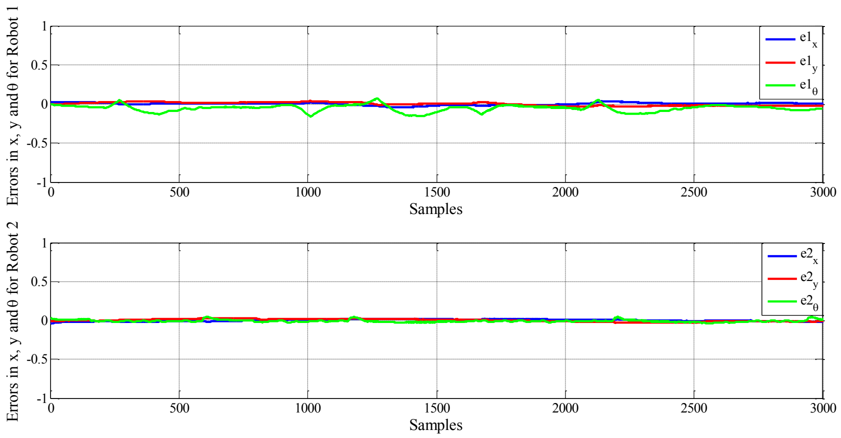


Figure 6.15: The errors in x, y and θ for the two robots

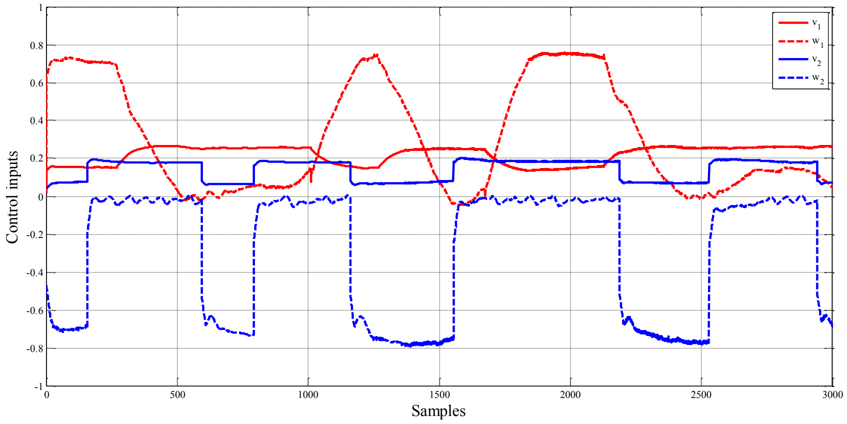


Figure 6.16: Control inputs v and ω for the two robots

Chapter 7

Conclusions and Future Work

7.1 Conclusions

Path planning for mobile robots in the patrolling problem is strongly related to the Travelling Salesman problem (TSP) and its variants the single depot multiple Travelling Salesman problem (mTSP) and the multidepot multiple Travelling Salesman problem (MmTSP). However, in the patrolling problem the viewpoints visiting action continues repeatedly as long as the operation lasts. Therefore, moving on the optimal path (which needs to be determined) with no constraint on the starting depots and/or the number of robots assigned to each one (which are typically imposed on traditional TSP) can have a significant impact on the overall performance of the system. The present work addresses various aspects of the patrolling problem introduced above. The results of each chapter can be summarized as follows:

In Chapter 2, two new formulations are developed for the minimum distance trajectory in patrolling problem with different sets of constraints. Given a set of viewpoints (depots) and mobile agents (robots), the objective is to visit the viewpoints while minimizing the overall travel distance but unlike existing work, it is assumed that the starting depot is not prespecified. The new formulations employ a set of binary variables, one for each pair of nodes, with a value equal to one if an edge between two nodes is part of the optimal trajectory, and zero otherwise. Another set of binary variables is also used to determine the optimal starting depot. The problem is investigated for both cases of return trip allowed and not allowed. It is shown that solving the minimum distance problem without specifying the starting depot, as suggested here, can lead to significantly better results compared to the conventional case of prespecified starting depot. Simulations demonstrate the superiority of the proposed formulations, and also compares the computation time of the proposed methods for different number of robots and viewpoints.

Chapter 3 introduces three new formulations for the minimum-distance trajectory in patrolling problems with different sets of constraints for a more general problem, namely, the multidepot multiple Travelling Salesman Problem (MmTSP). In the first problem, unlike existing work, it is assumed that the starting depots and starting robots are not prespecified. Moreover, in the second problem the number of depots and robots are considered unknown. For the minimum-distance trajectories, each depot can only have a single robot. Simulations show that solving the minimum-distance problem without specifying the starting depots and robots, as suggested in this chapter, can lead to significantly better results compared to the conventional case of prespecified starting depots and robots.

Chapter 4 introduces new formulations and algorithms for the minimum-time trajectories in the patrolling problem with different sets of constraints. Unlike existing results which are mainly focused on minimizing either the total travel distance or the time window, in this chapter the total travel time is investigated for the case where neither the starting depots and robots nor their numbers are prespecified. The objective is to visit a given set of viewpoints while minimizing the overall travel time. It is shown that the minimum-time trajectories may not be unique. Simulations confirm that solving the minimum-time problem without specifying the starting depots, as suggested in this chapter, can lead to completely different results compared to conventional minimum-distance patrolling problems. Hence, in the time-sensitive applications (such as rescue missions) using the minimum-time trajectory can be very important.

Chapter 5 introduces a new patrolling method inspired by the movement of a pulley to convert the Euclidean Travelling Salesman Problem (ETSP) optimal solution to a sub-optimal kinematic-feasible Dubins' Travelling Salesman Problem (DTSP) solution that preserves the original order of ETSP. The new algorithm, referred to as pulleys algorithm (PA), introduces an upper bound that is less than the upper bounds obtained by using the two point algorithm (2PA), the three point algorithm (3PA) and the alternating algorithm (AA). Two enhancements are also introduced to the 2PA and the AA to improve their optimal solutions.

Chapter 6 introduces the experimental results, where the optimal ETSP trajectories for Chapters (2 - 4) and the sub-optimal DTSP trajectories of Chapter 5 are tested in the lab. To this end, both two-wheeled mobile robots (TWMRs) and Dubins' vehicles are employed and the linearized model predictive control (LMPC) approach is used.

7.2 Future Work

In what follows, some of the possible extensions of the results obtained in this dissertation as well as some relevant problems for future study are presented.

- The patrolling operation for heterogeneous robots and viewpoints would be an interesting problem to investigate given its practical significance. The problem involves deriving formulations for the optimal trajectories for both cases of prespecified and non-prespecified depots and robots.

- Developing a cloud-based patrolling operation for a distributed autonomous robotics network using the *robot operating system* (ROS) would be very important for emerging applications.

- The patrolling operation with reduced latency, where it is aimed to reduce the frequency of visiting each viewpoint would be an interesting problem. In particular, the problem can have applications where secure patrolling performance is desired.

- Designing fault-tolerant strategies for patrolling operations to account for typical faults in this type of system would extend the scope of the proposed solution to more practical scenarios.

- Studying the patrolling operation in a 3D environment would be another direction for future work, as far as applications involving unmanned aerial vehicles (UAV) and unmanned underwater vehicles (UUV) are concerned.

Bibliography

[1] R. P. M. Chan, K. A. Stol, and C. R. Halkyard, “Review of modelling and control of two-wheeled robots,” *Annual Reviews in Control*, vol. 37, no. 1, pp. 89 – 103, 2013.

[2] M. H. Amoozgar and Y. M. Zhang, “Trajectory tracking of wheeled mobile robots: A kinematical approach,” in *Proceedings of the 8th IEEE/ASME International Conference on Mechatronics and Embedded Systems and Applications*, 2012, pp. 275 – 280.

[3] J. Clark and R. Fierro, “Mobile robotic sensors for perimeter detection and tracking,” *ISA Transactions*, vol. 46, pp. 3 – 13, 2007.

[4] D. Kingston, R. W. Beard, and R. S. Holt, “Decentralized perimeter surveillance using a team of UAVs,” *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1394 – 1404, 2008.

[5] S. Susca, F. Bullo, and S. Martinez, “Monitoring environmental boundaries with a robotic sensor network,” *IEEE Transactions on Control Systems Technology*, vol. 16, no. 2, pp. 288 – 296, 2008.

[6] Y. Elmaliach, A. Shiloni, and G. A. Kaminka, “A realistic model of frequency-based multi-robot polyline patrolling,” in *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems*, 2008.

[7] F. Pasqualetti, J. W. Durham, and F. Bullo, “Cooperative patrolling viaweighted tours: Performance analysis and distributed algorithms,” *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1181 – 1188, 2012.

[8] F. Pasqualetti, A. Franchi, and F. Bullo, “On cooperative patrolling: Optimal trajectories, complexity analysis, and approximation algorithms,” *IEEE Transactions on Robotics*, vol. 28, no. 3, pp. 592 – 606, 2012.

[9] D. Portugal and R. P. Rocha, “Distributed multi-robot patrol: A scalable and fault-tolerant framework,” *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1572 – 1587, 2013.

[10] A. R. Girard, A. S. Howell, and J. K. Hedrick, “Border patrol and surveillance missions using multiple unmanned air vehicles,” in *Proceedings of the 43rd IEEE Conference on Decision and Control*, vol. 1, 2004, pp. 620 – 625.

[11] J. L. Jenkin, J. Marquardson, J. G. Proudfoot, J. S. Valacich, E. Golob, and J. F. Nunamaker, “The checkpoint simulation a tool for informing border patrol checkpoint design and resource allocation,” in *Proceedings of the European Intelligence and Security Informatics Conference*, 2013, pp. 252 – 255.

[12] A. S. Matveev, H. Teimoori, and A. V. Savkin, “A method for navigation of an autonomous vehicle for border patrol,” in *Proceedings of the American Control Conference*, 2010, pp. 6187 – 6190.

[13] Y. Elmaliach, N. Agmon, and G. A. Kaminka, “Multi-robot area patrol under frequency constraints,” *Annals of Mathematics and Artificial Intelligence*, vol. 57, no. 3-4, pp. 293 – 320, 2009.

[14] A. Glad, O. Simonin, O. Buffet, and F. Charpillet, “Influence of different execution models on patrolling ant behaviors: from agents to robots,” in *Proceedings*

of the 9th International Conference on Autonomous Agents and Multiagent Systems, vol. 3, 2010, pp. 1173 – 1180.

[15] Y. Guo, L. E. Parker, and R. Madhavan, "Towards collaborative robots for infrastructure security applications," in *Proceedings of the International Symposium on Collaborative Technologies and Systems*, 2004, pp. 235 – 240.

[16] ——, "Collaborative robots for infrastructure security applications," *Studies in Computational Intelligence*, vol. 50, pp. 185 – 200, 2007.

[17] Y. Chevaleyre, "Theoretical analysis of the multi-agent patrolling problem," in *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, 2004, pp. 302 – 308.

[18] N. Agmon, C.-L. Fok, Y. Emaliah, P. Stone, C. Julien, and S. Vishwanath, "On coordination in practical multi-robot patrol," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2012, pp. 650 – 656.

[19] C. Pippin and H. Christensen, "Trust modeling in multi-robot patrolling," in *Proceedings of the IEEE International Conference on Robotics & Automation*, 2014, pp. 59 – 66.

[20] D. Portugal and R. P. Rocha, "Scalable, fault-tolerant and distributed multi-robot patrol in real world environments," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 4759 – 4764.

[21] S. gang Cui and J. lei Dong, "Detecting robots path planning based on improved genetic algorithm," in *Proceedings of the Third International Conference on Instrumentation, Measurement, Computer, Communication and Control*, 2013, pp. 204 – 207.

[22] F. Pasqualetti, A. Franchi, and F. Bullo, “On optimal cooperative patrolling,” in *Proceedings of the 49th IEEE Conference on Decision and Control*, 2010, pp. 7153 – 7158.

[23] F. Pasqualetti, A. Franchi, , and F. Bullo, “Hierarchical autonomous mobile control system of a patrol robot for nuclear power plants,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1, 1995, pp. 837 – 842.

[24] F. Xu, Z. Li, and K. Yuan, “The design and implementation of an autonomous campus patrol robot,” in *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, 2007, pp. 250 – 255.

[25] J. Jin, C. Jung, D. Kim, and W. Chung, “Development of an autonomous outdoor patrol robot in private road environment,” in *Proceedings of the International Conference on Control, Automation and Systems*, 2010, pp. 1918 – 1921.

[26] V. An and Z. Qu, “Triangulation-based path planning for patrolling by a mobile robot,” in *Proceedings of the Australian Control Conference*, 2013, pp. 183 – 188.

[27] D. Portugal, C. Pippin, R. P. Rocha, and H. Christensen, “Finding optimal routes for multi-robot patrolling in generic graphs,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 363 – 369.

[28] Y. Elmaliach, N. Agmon, and G. A. Kaminka, “Multi-robot area patrol under frequency constraints,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2007, pp. 385 – 390.

[29] D. Portugal and R. P. Rocha, "On the performance and scalability of multi-robot patrolling algorithms," in *Proceedings of the IEEE International Symposium on Safety, Security and Rescue Robotics*, 2011, pp. 50 – 55.

[30] ——, "Decision methods for distributed multi-robot patrol," in *Proceedings of the IEEE International Symposium on Safety, Security and Rescue Robotics*, 2012, pp. 1 – 6.

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

[32] P. Sujit, D. Kingston, and R. Beard, "Cooperative forest fire monitoring using multiple UAVs," in *Proceedings of the 46th IEEE Conference on Decision and Control*, 2007, pp. 4875 – 4880.

[33] A. Sonmez, E. Kocyigit, and E. Kugu, "Optimal path planning for UAVs using genetic algorithm," in *Proceedings of the International Conference on Unmanned Aircraft Systems*, 2015, pp. 50 – 55.

[34] T. Turker, O. K. Sahingoz, and G. Yilmaz, "2d path planning for UAVs in radar threatening environment using simulated annealing algorithm," in *Proceedings of the International Conference on Unmanned Aircraft Systems*, 2015, pp. 56 – 61.

[35] J. R. Raol and A. K. Gopal, *Mobile Intelligent Autonomous Systems.* CRC Press, 2015.

[36] R. Carona, A. P. Aguiar, and J. Gaspar, "Control of unicycle type robots tracking, path following and point stabilization," in *Proceedings of the IV Jornadas de Engenharia Electronica e Telecomunicacoes e de Computadores*, 2008, pp. 180 – 185.

[37] R. Matai, S. Singh, and M. L. Mittal, *Traveling Salesman Problem: an Overview of Applications, Formulations, and Solution Approaches.* InTech, 2010.

[38] T. Bektas, "The multiple traveling salesman problem: an overview of formulations and solution procedures," *Omega*, vol. 34, no. 3, pp. 209 – 219, 2006.

[39] J. A. Svestka and V. E. Huckfeldt, "Computational experience with an m-salesman traveling salesman algorithm," *Management Science*, vol. 19, no. 7, pp. 790 – 799, 1973.

[40] R. D. Angel, W. L. Caudle, R. Noonan, and A. Whinston, "Computer assisted school bus scheduling," *Management Science*, vol. 18, no. 6, pp. 279 – 288, 1972.

[41] B. L. Brumitt and A. Stentz, "Dynamic mission planning for multiple mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3, 1996, pp. 2396 – 2401.

[42] ——, "Grammps: a generalized mission planner for multiple mobile robots in unstructured environments," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, 1998, pp. 1564 – 1571.

[43] Z. Yu, L. Jinhai, G. Guochang, Z. Rubo, and Y. Haiyan, "An implementation of evolutionary computation for path planning of cooperative mobile robots," in *Proceedings of the 4th World Congress on Intelligent Control and Automation*, vol. 3, 2002, pp. 1798 – 1802.

[44] J. Ryan, T. Bailey, J. Moore, and W. Carlton, "Reactive tabu search in unmanned aerial reconnaissance simulations," in *Proceedings of the Winter Simulation Conference*, vol. 1, 1998, pp. 873 – 879.

[45] S. A. S. Alhamdy, M. Garakani, and M. Abvali, “Multi depot multiple traveling salesman problem using genetic algorithm,” *Journal of American Science*, vol. 8, no. 12, pp. 1483 – 1489, 2012.

[46] I. Kara and T. Bektas, “Integer linear programming formulations of multiple salesman problems and its variations,” *European Journal of Operational Research*, vol. 174, no. 3, pp. 1449 – 1458, 2006.

[47] S. Yadlapalli, W. Malik, S. Darbha, and M. Pachter, “A lagrangian-based algorithm for a multiple depot, multiple traveling salesmen problem,” *Nonlinear Analysis: Real World Applications*, vol. 10, no. 4, pp. 1990 – 1999, 2009.

[48] K. Sundary and S. Rathinamz, “An exact algorithm for a heterogeneous, multiple depot, multiple traveling salesman problem,” in *Proceedings of the International Conference on Unmanned Aircraft Systems*, 2015, pp. 366 – 371.

[49] C.-W. Lim, S. Park, C.-K. Ryoo, K. Choi, and J.-H. Cho, “A path planning algorithm for surveillance UAVs with timing mission constrains,” in *Proceedings of International Conference on Control, Automation and Systems*, 2010, pp. 2371 – 2375.

[50] K. Karabulut and M. F. Tasgetiren, “A variable iterated greedy algorithm for the traveling salesman problem with time windows,” *Information Sciences*, vol. 279, pp. 383 – 395, 2014.

[51] M. Lopez-Ibanez, C. Blum, J. W. Ohlmann, and B. W. Thomas, “The traveling salesman problem with time windows: Adapting algorithms from travel-time to makespan optimization,” *Applied Soft Computing*, vol. 13, no. 9, pp. 3806 – 3815, 2013.

[52] R. F. da Silva and S. Urrutia, “A general VNS heuristic for the traveling

salesman problem with time windows," *Discrete Optimization*, vol. 7, no. 4, pp. 203 – 211, 2010.

[53] X. Ma and D. A. Castanon, "Receding horizon planning for Dubins travelling salesman problems," in *Proceedings of the 45th IEEE Conference on Decision and Control*, 2006, pp. 5453 – 5458.

[54] L. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, pp. 497 – 516, 1957.

[55] X.-N. Bui, J.-D. Boissonnat, P. Soueres, and J.-P. Laumond, "Motion planning and control for non-holonomic mobile robots," in *Proceedings of the IEEE International Symposium on Intelligent Control*, 1995, pp. 551 – 557.

[56] W. Wu, H. Chen, and P.-Y. Woo, "Time optimal path planning for a wheeled mobile robot," *Journal of Robotic Systems*, vol. 17, no. 11, pp. 585 – 591, 2000.

[57] A. M. Shkela and V. Lumelsky, "Classification of the Dubins set," *Robotics and Autonomous Systems*, vol. 34, no. 4, pp. 179 – 202, 2001.

[58] J. W. Yeol, Y. S. Ryu, and M. A. Montalvo, "Shortest trajectory planning of wheeled mobile robots with constraints," in *Proceedings of the IEEE on Networking, Sensing and Control*, 2005, pp. 883 – 888.

[59] X.-N. Bui, J.-D. Boissonnat, P. Soueres, and J.-P. Laumond, "Shortest path synthesis for Dubins non-holonomic robot," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1, 1994, pp. 2 – 7.

[60] K. Savla, E. Frazzoli, and F. Bullo, "On the point-to-point and traveling salesperson problems for Dubins' vehicle," in *Proceedings of the American Control Conference*, 2005, pp. 786 – 791.

[61] J. L. Ny and E. Feron, "An approximation algorithm for the curvature-constrained traveling salesman problem," in *Proceedings of the 43rd Annual Allerton Conference on Communications, Control and Computing*, 2005, pp. 620 – 629.

[62] K. Savla, F. Bullo, and E. Frazzoli, "On traveling salesperson problems for Dubins' vehicle: stochastic and dynamic environments," in *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005, pp. 4530 – 4535.

[63] K. Savla, E. Frazzoli, and F. Bullo, "On the Dubins traveling salesperson problems: Novel approximation algorithms," in *Proceedings of the Robotics: Science and Systems II Conference*, 2006.

[64] K. Savla, F. Bullo, and E. Frazzoli, "On traveling salesperson problems for a double integrator," in *Proceedings of the 44th IEEE Conference on Decision and Control*, 2006, pp. 5305 – 5310.

[65] K. Savla, E. Frazzoli, and F. Bullo, "Traveling salesperson problems for the Dubins vehicle," *IEEE Transactions on Automatic Control*, vol. 53, no. 6, pp. 1378 – 1391, 2008.

[66] J. L. Ny, E. Feron, and E. Frazzoli, "On the Dubins traveling salesman problem," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 265 – 270, 2012.

[67] X. Yu and H. J.Y., "Optimal path planning for an autonomous robot-trailer system," in *Proceedings of the 38th Annual Conference on IEEE Industrial Electronics Society*, 2012, pp. 2762 – 2767.

[68] J. S. Kim and B. K. Kim, "Minimum-time trajectory generation algorithm along curved paths for mobile robots with a motor control input constraint,"

in *Proceedings of the IEEE/SICE International Symposium on System Integration*, 2012, pp. 224 – 229.

[69] J. T. Isaacs and J. P. Hespanha, "Dubins traveling salesman problem with neighborhoods: A graph-based approach," *Algorithms*, vol. 6, no. 1, pp. 84 – 99, 2013.

[70] P. Isaiah and T. Shima, "Motion planning algorithms for the Dubins travelling salesperson problem," *Automatica*, vol. 53, pp. 247 – 255, 2015.

[71] R. W. Brockett, "Asymptotic stability and feedback stabilization," in *Differential Geometric Control Theory.* Birkhauser, 1983, pp. 181 – 191.

[72] H. Guangxin and Z. Yanhui, "Global trajectory tracking control of nonholonomic WMRs with driving motor dynamics being considered," in *Proceedings of the 2nd International Conference on Intelligent Control and Information Processing*, vol. 2, no. 2, 2011, pp. 640 – 643.

[73] M. Brezak, I. Petrovic, and N. Peric, "Experimental comparison of trajectory tracking algorithms for nonholonomic mobile robots," in *Proceedings of the 35th Annual Conference on the IEEE Industrial Electronics Society*, 2009, pp. 2229 – 2234.

[74] G. Oriolo, A. D. Luca, R. D. Luca, and M. Vendittelli, "WMR control via dynamic feedback linearization: Design, implementation, and experimental validation," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 6, pp. 835 – 852, 2002.

[75] R. Solea and U. Nunes, "Trajectory planning with velocity planner for fully-automated passenger vehicles," in *Proceedings of the 9th IEEE International Conference on Intelligent Transportation Systems*, 2006, pp. 474 – 480.

[76] D. Soetanto, L. Lapierre, and A. Pascoal, “Adabtive non-singular path-following, control of dynamic wheeled robots,” in *Proceedings of the 42nd IEEE Conference on Decision and Control*, vol. 2, no. 42, 2003, pp. 1765 – 1770.

[77] M.Aicardi, G.Casalino, A. Bicchi, and A.Balestrino, “Closed loop steering of unicycle like vehicles via Lyapunov techniques,” *Robotics and Automation Magazine*, vol. 2, no. 1, pp. 27 – 35, 1995.

[78] K. Kanjanawaniskul, “Motion control of a wheeled mobile robot using model predictive control: A survey,” *KKU Research Journal*, vol. 17, no. 5, pp. 811 – 837, 2012.

[79] F. Xie and R. Fierro, “Stabilization of nonholonomic robot formations: A first-state contractive model predictive control approach,” *Journal of Computing and Information Technology*, vol. 17, no. 1, pp. 37 – 50, 2009.

[80] D. Gu and H. Hu, “Receding horizon tracking control of wheeled mobile robots,” *IEEE Transactions on Control Systems Technology*, vol. 17, no. 4, pp. 743 – 749, 2006.

[81] G. Klancar and I. Skrjanc, “Tracking-error model-based predictive control for mobile robots in real time,” *Robotics and Autonomous Systems*, vol. 55, no. 6, pp. 460 – 469, 2007.

[82] F. Kuhne, W. F. Lages, and J. M. G. da Silva Jr., “Model predictive control of a mobile robot using linearization,” in *Proceedings of the Mechatronics and Robotics Conference*, 2004, pp. 525 – 530.

[83] W. F. Lages and J. A. V. Alves, “Real-time control of a mobile robot using linearized model predictive control,” in *Proceedings of the 4th IFAC Symposium on Mechatronic Systems*, vol. 4, no. 1, 2006, pp. 968 – 973.

[84] M. Seyr and S. Jakubek, “Mobile robot predictive trajectory tracking,” in *Proceedings of the 2nd International Conference on Informatics in Control, Automation and Robotics*, 2005.

[85] F. Kuhne, , J. M. G. da Silva Jr., and W. F. Lages, “Mobile robot trajectory tracking using model predictive control,” in *Proceedings of the IEEE 2nd Latin American Robotics Symposium*, 2005.

[86] S. Wei, M. Zefran, K. Uthaichana, and R. A. DeCarlo, “Hybrid model predictive control for stabilization of wheeled mobile robots subject to wheel slippage,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2007, pp. 2373 – 2378.

[87] S. G. Vougioukas, “Reactive trajectory tracking for mobile robots based on non linear model predictive control,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2007, pp. 3074 – 3079.

[88] A. Rosales, M. Pena, G. Scaglia, V. Mut, and F. di Sciascio, “Dynamic model based predictive control for mobile robots,” in *Proceedings of the Procesamiento de la Informacion y Control*, 2007.

[89] X. Kewei, Z. Yan, and W. Wei, “Path tracking and trajectory planning for nonholonomic wheeled mobile robots,” in *Proceedings of the Proceedings of the 27th Chinese Control Conference*, 2008, pp. 415 – 420.

[90] J. Lavaei, A. Momeni, and A. G. Aghdam, “A model predictive decentralized control scheme with reduced communication requirement for spacecraft formation,” *IEEE Transactions on Control Systems Technology*, vol. 16, no. 2, pp. 268 – 278, 2008.

[91] P. Petrov and L. Dimitrov, “Nonlinear path control for a differential drive

mobile robot," in *Proceedings of the Proceedings of the 12th WSEAS International Conference on Automatic Control, Modeling and Simulation*, vol. 11, no. 1, 2010, pp. 41 – 45.

[92] S. Akiba, T. Zanma, and M. Ishida, "Optimal tracking control of two-wheeled mobile robots based on model predictive control," in *Proceedings of the 11th IEEE International Workshop on Advanced Motion Control*, 2010, pp. 454 – 459.

[93] R. Hedjar, M. Alsulaiman, and K. Almutib, "Approximated nonlinear predictive control for trajectory tracking of a wheeled mobile robot," in *Proceedings of the First International Conference on Robot, Vision and Signal Processing*, 2011, pp. 296 – 299.

[94] C.-H. Hsieh and J.-S. Liu, "Nonlinear model predictive control for wheeled mobile robot in dynamic environment," in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2012, pp. 363 – 368.

[95] M.-M. Ma, S. Li, and X.-J. Liu, "Tracking control and stabilization of wheeled mobile robots by nonlinear model predictive control," in *Proceedings of the 31st Chinese Control Conference*, 2012, pp. 4056 – 4061.

[96] D. Kiss and G. Tevesz, "Advanced dynamic window based navigation approach using model predictive control," in *Proceedings of the 17th International Conference on Methods and Models in Automation and Robotics*, 2012, pp. 148 – 153.

[97] N. Nakpong and S. Yamamoto, "Just-in-time predictive control for a two-wheeled robot," in *Proceedings of the 10th International Conference on ICT and Knowledge Engineering*, 2012, pp. 95 – 98.

[98] R. Singhal and P. B. Sujit, “3d trajectory tracking for a quadcopter using mpc on a 3d terrain,” in *Proceedings of the International Conference on Unmanned Aircraft Systems*, 2015, pp. 1385 – 1390.

[99] M. A. Kamel and Y. M. Zhang, “Decentralized leader-follower formation control with obstacle avoidance of multiple unicycle mobile robots,” in *Proceedings of the IEEE 28th Canadian Conference on Electrical and Computer Engineering*, 2015, pp. 406 – 411.

[100] W. Ghadiry, J. Habibi, A. Aghdam, and Y. M. Zhang, “A new formulation for energy-efficient trajectory optimization in patrolling problem with non-predefined starting depot,” in *Proceedings of the 10th International Conference on Intelligent Unmanned Systems*, 2014.

[101] “Mosek 7.0 optimization software,” 2009.

[102] “Gurobi 6.0 optimization software,” 2014.

[103] W. Ghadiry, J. Habibi, A. G. Aghdam, and Y. M. Zhang, “Generalized formulation for trajectory optimization in patrolling problems,” in *Proceedings of the 28th annual IEEE Canadian Conference on Electrical and Computer Engineering*, 2015, pp. 231–236.

[104] A. A. Markov, “Some examples of the solution of a special kind problem in greatest and least quantities,” *Soobscenija Charkovskogo Matematiceskogo Obscestva*, vol. 1, no. 5, pp. 250 – 276, 1889 (in Russian).

[105] D. A. Anisi, “Optimal motion control of a ground vehicle,” *Swedish Defense Research Agency, Technical Report*, 2003.

[106] H. Sussman and G. Tang, “Shortest paths for the Reeds-Shepp car: a worked out example of the use of geometric techniques in nonlinear optimal control,”

in *Rutgers Center for Systems and Control, Technical Report*, 1991, pp. 1 – 71.

[107] J.-D. Boissonnat, A. Cerezo, and J. Leblond, "Shortest paths of bounded curvature in the plane," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1992, pp. 2315 – 2320.

[108] L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB*. Springer, 2009.

[109] M. Behrendt, "A basic working principle of model predictive control," in *Wikipedia, 2 Oct 2009*, 2009.

[110] Quanser, "QGV user manual," 2009.