Monitoring using Heterogeneous Autonomous Agents

by

Jonathan Las Fargeas

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in the University of Michigan
2015

To my family, friends, and mentors.

ACKNOWLEDGMENTS

I first and foremost would like to thank my advisors, Professors Pierre Kabamba and Anouck Girard, for all the support they have given me over the course of my graduate career. Their advice has ranged a wide variety of topics, not restricted to academics, and has been instrumental to bringing me to where I am today.

I am very grateful to Professors Ilya Kolmanovsky, Necmiye Ozay, and Daniel Inman for joining my committee and offering very useful feedback about my research.

I would like to express my gratitude to Professor James Cutler for supporting me in my first years at Michigan and giving me the opportunity to work on spacecraft.

I am also very appreciative of the help I have received over the years from my lab mates, especially Ricardo Bencatel, Baro Hyun, Moritz Niendorf, Dave Oyler, Johnhenri Richardson, and Jinwoo Seok.

I would like to acknowledge the United States Air Force for funding my research. I would also like to thank the Air Force Research Laboratory, the Michigan Space Grant Consortium, and the University of Michigan Department of Aerospace Engineering for their support.

Finally, I would like to thank my family and friends for their encouragement over the duration of my graduate studies.

TABLE OF CONTENTS

Dedication . . . . **ii**

Acknowledgments . . . . **iii**

List of Figures . . . . **vii**

List of Tables . . . . **viii**

Abstract . . . . **ix**

Chapter

1 Introduction **1**

1.1 Motivation 1
1.2 Problem statement 2
1.3 Generalized problem formulation 4
1.4 Original contributions 5
1.5 Organization 7

2 Literature survey **8**

2.1 Path planning for surveillance 8
2.1.1 Vehicle routing problems 9
2.1.2 Traveling salesman problems 10
2.1.3 Patrolling 10
2.1.4 Target tracking 11
2.2 Sensor placement 14
2.2.1 Graph theoretic approaches 14
2.2.2 Optimization approaches 15
2.3 Marsupial operations 19
2.3.1 Marsupial architectures 19
2.3.2 Path planning for marsupial systems 20
2.4 Intractability 21
2.4.1 Complexity 21
2.4.2 Complexity class 22
2.4.3 Overcoming complexity 23
2.5 Research directions 24

3 Paths for mobile agents **26**

3.1 Monitoring stationary objects with no fuel constraints . . . 26
3.1.1 Model . . . 27
3.1.2 Problem formulation . . . 30
3.1.3 Solutions & periodicity . . . 30
3.1.4 Complexity . . . 31
3.1.5 Heuristics . . . 33
3.1.6 Simulations . . . 41
3.2 Monitoring stationary objects with fuel considerations . . . 43
3.2.1 Model . . . 43
3.2.2 Dynamics . . . 45
3.2.3 Periodicity . . . 46
3.2.4 Problem formulation . . . 48
3.2.5 Solution . . . 50
3.2.6 Algorithm correctness, completeness, and complexity . . . 55
3.3 Pursuing mobile objects . . . 56
3.3.1 Model of agents . . . 57
3.3.2 Mobile object model . . . 61
3.3.3 Mobile object interception . . . 62
3.3.4 Problem formulation . . . 64
3.3.5 Mobile agent path selection . . . 66
3.3.6 Local search heuristic . . . 72
3.3.7 Simulations . . . 72
3.4 Summary . . . 82

4 Configurations for stationary agents . . . 83

4.1 Model . . . 83
4.1.1 Revisit deadlines . . . 84
4.1.2 Mobile object motion in graph . . . 84
4.2 Problem formulation . . . 91
4.3 Stationary agent placement . . . 92
4.3.1 Background . . . 92
4.3.2 Fisher information . . . 93
4.3.3 Probability of misclassification . . . 96
4.3.4 Penalty incurred . . . 97
4.4 Special cases . . . 99
4.4.1 No combined stationary agents . . . 99
4.4.2 Computational complexity . . . 103
4.5 Stationary agent timing . . . 103
4.5.1 Lower bound . . . 104
4.5.2 Feasibility of revisit deadlines . . . 104
4.5.3 Selecting revisit deadlines . . . 105
4.6 Simulations . . . 107
4.6.1 Comparison of stationary agent placements . . . 109
4.6.2 Qualitative comparison of stationary agent placements . . . 111
4.6.3 Example data . . . 113

4.7 Summary 114

5 Paths for marsupial agents 118

5.1 Optimal control for marsupial agents 119
5.1.1 Generic marsupial path planning problem formulation 119
5.1.2 Costates and Hamiltonian 121
5.1.3 Optimal control 122
5.1.4 Natural boundary conditions 122
5.1.5 Transversality conditions 122
5.2 Pair of marsupial agents observing a stationary object 131
5.2.1 Model 132
5.2.2 Problem formulation 134
5.2.3 Technical approach 134
5.2.4 Simulations 141
5.3 Summary 142

6 Conclusions 146

6.1 Summary 146
6.2 Concluding remarks 147
6.3 Future directions 148

Bibliography 150

LIST OF FIGURES

1.1 Cooperative surveillance and pursuit problem using mobile and stationary agents. 2
1.2 Cooperative reconnaissance problem using marsupial agents. 3
1.3 Cooperative monitoring problems. 4

3.1 Failure of earliest deadline first heuristic and success of two step maximum minimum slack time heuristic on example A. 36
3.2 Earliest deadline first success and failure of one step heuristics on example B. . 38
3.3 Failure of two step heuristics on example B. 40
3.4 Performance of zero and one step heuristics on example C. 42
3.5 Visualization of base, stationary agents, and roads for scenario A. 76
3.6 Visualization of base, stationary agents, and roads for scenario B. 77
3.7 Capture index for scenario A2 using *mobileAgentsAction*() (top) and *localSearch*() (bottom). 78
3.8 Capture index for scenario A3 using *mobileAgentsAction*() (top) and *localSearch*() (bottom). 79
3.9 Capture index for scenario B1 using *mobileAgentsAction*() (top) and *localSearch*() (bottom). 80
3.10 Capture index for scenario B2 using *mobileAgentsAction*() (top) and *localSearch*() (bottom). 81

4.1 Quality of detections from nodes and edges. 98
4.2 Example of identical and similar placements. 108
4.3 Example of similar and distinct placements. 109

5.1 Carrier and passenger optimal paths for different initial conditions in scenario A. 143
5.2 Carrier and passenger optimal paths for different initial conditions in scenario B. 144
5.3 Carrier and passenger optimal paths for different initial conditions in scenario C. 145

LIST OF TABLES

3.1 Configurations of examples used 35
3.2 Base and stationary agents parameters for scenarios A and B. 75
3.3 Vehicle parameters for scenarios A and B. 76

4.1 Comparison of all stationary agent placements schemes. 110
4.2 Pairwise comparison of stationary agent placements schemes. 116
4.3 Comparison of quality of stationary agent configurations. 117

5.1 Simulation parameters. 141

ABSTRACT

Monitoring using Heterogeneous Autonomous Agents

by

Jonathan Las Fargeas

Co-Chairs: Anouck Renee Girard and Pierre Tshimanga Kabamba

This dissertation studies problems involving different types of autonomous agents observing objects of interests in an area. Three types of agents are considered: mobile agents, stationary agents, and marsupial agents, i.e., agents capable of deploying other agents or being deployed themselves. Objects can be mobile or stationary.

The problem of a mobile agent without fuel constraints revisiting stationary objects is formulated. Visits to objects are dictated by revisit deadlines, i.e., the maximum time that can elapse between two visits to the same object. The problem is shown to be NP-complete and heuristics are provided to generate paths for the agent. Almost periodic paths are proven to exist. The efficacy of the heuristics is shown through simulation. A variant of the problem where the agent has a finite fuel capacity and purchases fuel is treated. Almost periodic solutions to this problem are also shown to exist and an algorithm to compute the minimal cost path is provided.

A problem where mobile and stationary agents cooperate to track a mobile object is formulated, shown to be NP-hard, and a heuristic is given to compute paths for the mobile agents. Optimal configurations for the stationary agents are then studied. Several methods are provided to optimally place the stationary agents; these methods are the maximization of Fisher information, the minimization of the probability of misclassification, and the minimization of the penalty incurred by the placement. A method to compute optimal revisit deadlines for the stationary agents is given. The placement methods are compared and their effectiveness shown using numerical results.

The problem of two marsupial agents, one carrier and one passenger, performing a general monitoring task using a constrained optimization formulation is stated. Necessary conditions for optimal paths are provided for cases accounting for constrained release of the passenger, termination conditions for the task, as well as retrieval and constrained retrieval of the passenger. A problem involving two marsupial agents collecting information about a stationary object while avoiding detection is then formulated. Necessary conditions for optimal paths are provided and rectilinear motion is demonstrated to be optimal for both agents.

CHAPTER 1

Introduction

1.1 Motivation

The use of autonomous vehicles for monitoring missions has grown steadily in recent years. With the increase in their use, these autonomous vehicles (agents) are tasked with increasingly large and complex missions (e.g., extended duration missions with many objects of interest and multiple agents available). Such missions include military surveillance missions performed by unmanned aerial vehicles, monitoring of ocean areas for potential oil spills using unmanned underwater vehicles, surveillance of radiation levels in a power plant using unmanned ground vehicles, or weather monitoring spacecraft.

These missions typically involve a number of objects of interest present in an area with properties that vary over time. The goal of the agents is to monitor the properties of the objects of interest in the area from which the agents can infer information directly related to their goal; e.g., measuring temperature and wind velocities to assess weather patterns in an area. The agents must select their paths and when to visit and gather information from the objects of interest such that their goal is accomplished. This may be achieved by observing the properties of the objects of interest while minimizing the mission time, the fuel consumed, or the risk accumulated by the agents. The agents use an estimate of how the properties of the objects of interest change over time to make their decisions.

One such problem may be mobile unmanned aerial vehicles cooperating with stationary

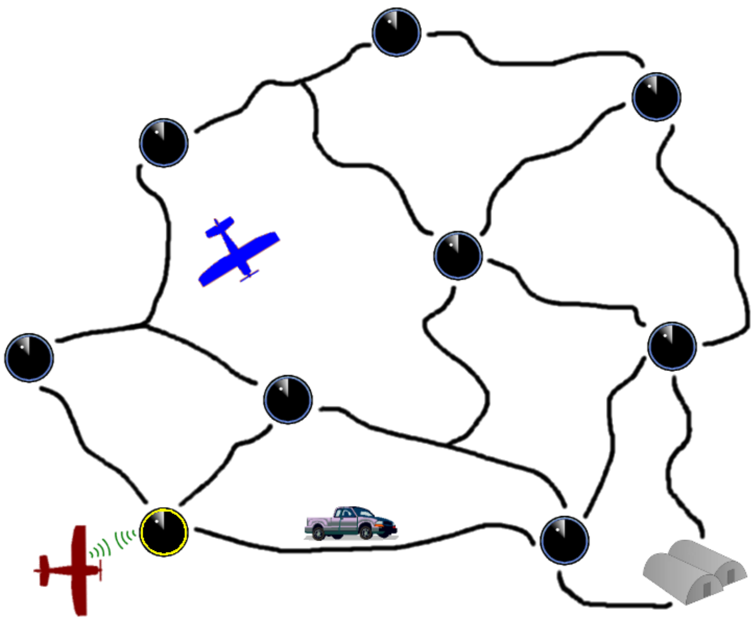


Figure 1.1: Cooperative surveillance and pursuit problem using mobile and stationary agents.

unattended ground sensors to track an intruder along a road network (Figure 1.1) or a mobile unmanned aerial vehicle capable of deploying a smaller unmanned aerial vehicle to collect information about an object of interest (Figure 1.2).

1.2 Problem statement

The problem for the agents can be stated as follows: **given models for the objects of interest, find paths for the agents such that the objects are monitored**. From this generalized problem statement, several questions of interest arise:

- Which paths accomplish the task?
- What properties do these paths have?
- How can these paths be computed?

These broad questions of interest are complex, hence to reduce their scope and inherent difficulty, the following constraints are applied:

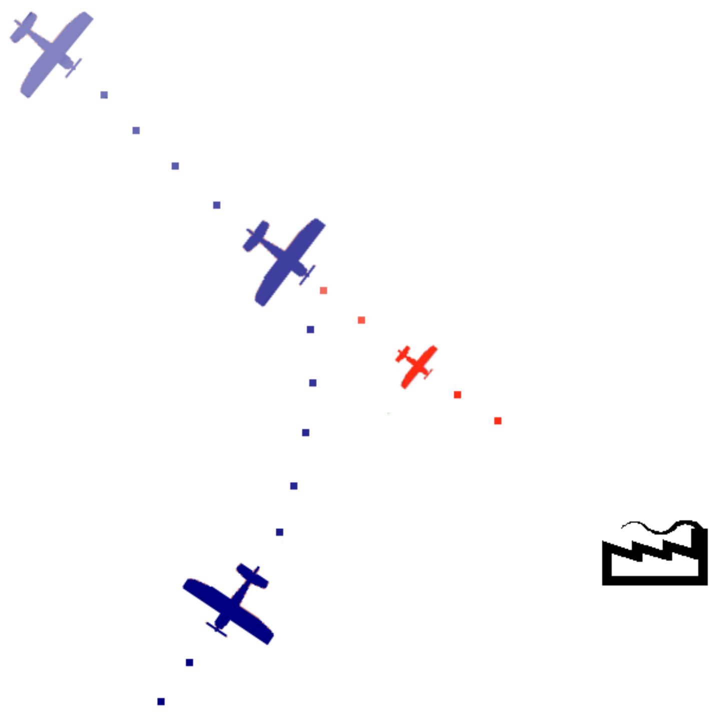


Figure 1.2: Cooperative reconnaissance problem using marsupial agents.

- The agents' movements are modeled using unicycle dynamics.
- The agents travel at a constant velocity.
- The agents possess accurate models of the objects of interest.
- Mobile objects move according to Markov models.

Even with these assumptions, there are many different possible configurations for the heterogeneous agents. Hence the problem is divided into three subproblems, each treating a different configuration for the autonomous systems where a configuration constrains the types of agents considered. In the first case, only mobile agents are considered and thus the problem is one of path planning. In the second case, both mobile and stationary agents are considered hence the conjunction of the first case and a sensor placement problem is studied. A question that arises during the study of problems with stationary and mobile agents is how could the system be improved if the mobile agents could pick-up and relocate

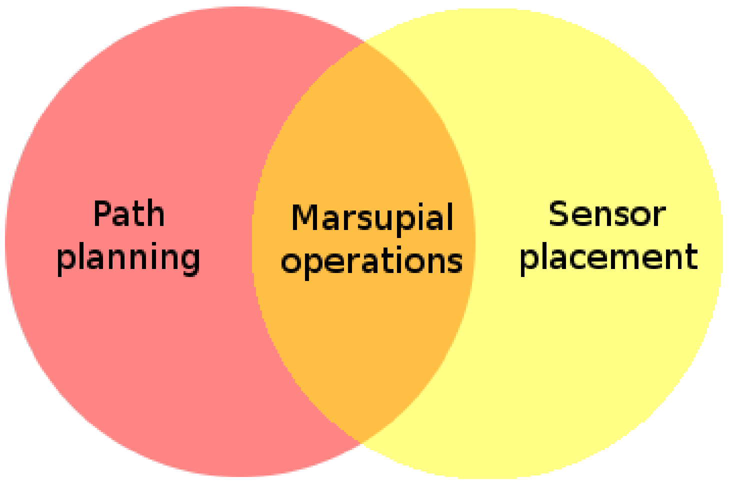


Figure 1.3: Cooperative monitoring problems.

the stationary agents. Hence, the last case studied is one where mobile agents are capable of deploying other agents (i.e., their relationship is marsupial) which is also a path planning problem but includes other factors such as releasing and retrieving agents. The following subquestions corresponding to different agent architectures are treated:

1. Path planning: How can mobile agents cooperate to achieve monitoring tasks?
2. Sensor placement: How can mobile and stationary agents cooperate to achieve monitoring tasks?
3. Marsupial operations: How can mobile marsupial agents cooperate to achieve monitoring tasks?

1.3 Generalized problem formulation

The systems considered in this dissertation are agents that can move and gather knowledge, in addition how the agents move affects how the agents gather knowledge. Thus, the dissertations concerns itself with the link between kinematics and epistemology. The following states and functions are introduced to build a general formulation for the problems considered in this dissertation. Let $x(t)$ be the kinematic states of the agents and $e(t)$ be the knowledge states of the agents. Let $f : x(t) \to \mathbb{R}$ be a function that evaluates the kinematic states and $g : e(t) \to \mathbb{R}$ a function that evaluates the knowledge states over the course of the task. Then the problems considered can be formulated as follows:

$$\min_{x(t),\forall t} \left(f\left(x(t)\right)\right) \text{ subject to } g\left(e(t)\right) = 0, \tag{1.1}$$

i.e., an objective function is optimized with respect to the kinematic states subject to constraints on the knowledge states.

For example, in Section 3.2 a mobile agent is tasked with satisfying revisit constraints to monitor stationary objects while minimizing fuel expenditures; for this formulation $e(t)$ contains the revisit constraint states (these states are called slack times and are introduced in Section 3.1), $f(\cdot)$ would account for the fuel costs, and $g(\cdot)$ accounts for the satisfaction of the revisit constraints. Another example is the problem studied in Section 5.2 where two marsupial agents are tasked with collecting a given amount of information about an object while minimizing their likelihood of detection; for this problem $e(t)$ contains the amount of information gathered, $f(\cdot)$ accounts for the likelihood of being detected, and $g(\cdot)$ checks whether the desired amount of information has been collected. In some problems, there may not be an objective function in which case we want to find paths for the agents that satisfy the knowledge requirements for the mission; e.g., the problem considered in Section 3.1 where a mobile agent is tasked with satisfying revisit constraints to monitor stationary objects.

1.4 Original contributions

The original contributions of this dissertation are:

- Almost periodic solutions to the persistent visitation problem are shown to exist for a single mobile agent with or without fuel consumption. Properties of tours that solve the problem are given. Heuristics to compute a path for a single agent without fuel consumption are provided and an algorithm to generate a cost minimal path for the agent when accounting for fuel is given. The cooperative surveillance and pursuit problem is formulated and a heuristic to solve the problem is provided. The problems are all demonstrated to be intractable.

- Methods for optimal sensor placement in the cooperative surveillance and pursuit problem using Fisher information, probability of misclassification, or the penalization of poor detections for mobile objects moving according to an ergodic Markov chain are given. An approach to compute the optimal revisit deadlines for the placed stationary agents using the Markov model of the mobile object is also presented.
- Necessary conditions for the paths of carrier and passenger agents at release and retrieval for general marsupial monitoring problems are presented. For a problem involving a single carrier and single passenger collecting information about a single object of interest, the agents are demonstrated to move rectilinearly.

The demonstration that the paths which solve the persistent visitation problem are periodic is significant because it shows that periodicity occurs without any a priori assumptions about periodicity unlike much of the literature. This periodicity also reduces the search space for solutions which is helpful for intractable problems. The algorithm provided for the computation of cost minimal paths is one valuable use of periodicity.

The heuristic provided for the cooperative surveillance and pursuit problem shows that pursuit using mobile agents not equipped with sensors is possible when cooperating with stationary agents equipped with sensors. The approaches given for the optimal configuration of the stationary agents provide potential mission designers with a variety of tools to best configure the agents.

The necessary conditions for the paths of passenger and carrier agents provide a solid theoretical foundation on which to study problems using marsupial agents. The proof of rectilinear motion for the passenger agent and carrier agent helps reduce the search space for the optimal paths of the agents.

1.5 Organization

The remainder of the dissertation is organized as follows. A literature survey relevant to path planning, sensor placement, and marsupial operations is provided in Chapter 2. Chapter 2 also contains a short review of theoretical background frequently discussed in the document, namely complexity classes and intractability.

Persistent monitoring of multiple stationary objects of interest using a single mobile agent is studied in Chapter 3. Properties of paths for the mobile agent, heuristics to compute paths for the mobile agent when fuel is neglected, and algorithms to generate paths for the mobile agent when fuel is considered are presented. A pursuit problem involving multiple mobile agents tracking a mobile object of interest along a graph is also studied; properties of the problem are shown and heuristics for the mobile agents' decision making process are given. The content of Chapter 3 is based on material presented in [1], [2], and [3].

Approaches for the configuration of stationary agents for monitoring tasks are presented in Chapter 4. Several methods to place the stationary agents and select their revisit deadlines are given, their properties shown, and their merits discussed via simulation results. Chapter 4 stems from material published in [4].

Analytical and numerical results for the problem of monitoring objects of interest using marsupial agents are given in Chapter 5. These include necessary conditions of optimal paths for both agents and properties of optimal paths for a pair of marsupial agents monitoring a single object of interest. Chapter 5 originates from material in [5] and [6].

A summary of the dissertation and concluding remarks are provided in Chapter 6.

CHAPTER 2

Literature survey

As discussed above, the problems being investigated touch upon multiple existing areas of study. Literature relevant to path planning, sensor placement, and marsupial operations for autonomous systems is now reviewed.

2.1 Path planning for surveillance

The problem of persistently monitoring a given area with one or more unmanned aerial vehicles has been studied extensively. In [7], the authors investigate the generation of optimal paths and cycles such that all the points in a given area are covered by the sensor footprint. In [8] and [9], the authors investigate how to optimally divide a mission area to be monitored by multiple unmanned aerial vehicles and how to minimize the time between visitations to the same region to improve overall surveillance. In [10] and [11], the authors investigate patrol algorithms and their relation to the frequency of visitation of each cell in a partitioned mission area.

In [12], the authors present controllers that minimize the accumulation of uncertainty in the mission space by varying the agent's velocity along a predetermined path. In [13] the goal is also to minimize an uncertainty metric in the mission area, but the controllers derived can control the movement of the mobile agents. Path planning for monitoring has also been studied for a variety of other vehicles, such as ground vehicles, boats, and spacecraft. In [14], the authors study science-optimal attitude control for a spacecraft in the presence of failures using stochastic dynamic programming. Surveillance using pan-tilt-zoom cameras is investigated in [15] where the authors provide methods to synthesize control protocols for the cameras to achieve their surveillance tasks. Monitoring missions can be modeled a variety of ways, monitoring tasks using ground vehicles are often modeled as vehicle routing problems.

2.1.1 Vehicle routing problems

The vehicle routing problem and methods to solve it are introduced in [16]. In VRP, a vehicle is tasked with performing pickup and delivery operations from a depot to a set of customers in a given area; the goal is to find a path for the vehicle that minimizes a given metric whether it be time, distance, or cost. In [17], the period vehicle routing problem (PVRP) is presented where planning is done over a multiple day horizon and each location must be visited at least once while some require periodic visits (e.g., every other day but no faster than once a day).

In [18], the vehicle routing problem with time windows (VRPTW) is presented; it is a variant of VRP where each location to be visited has a time window in which the visit must occur. In [19], the periodic vehicle routing problem with time windows, which is the combination of VRPTW and PVRP, is presented. Energy efficient paths for unmanned aircraft are studied in [20], graphs that account for wind conditions are constructed and minimum energy paths through this graph are computed using an A^* heuristic approach.

In [21], the authors generate minimum length Hamiltonian cycles such that the distance between each target node and the cycle is within a certain threshold. The optimization criteria for vehicle routing problems are typically the distance traveled or fuel consumed, however other criteria of interest have included consistency of the schedule of visits from day to day [22] or minimizing the emissions by the vehicles [23].

2.1.2 Traveling salesman problems

When the pickup and delivery aspects of VRP are removed, the problem is reduced to the Traveling-Salesman Problem (TSP). In TSP the goal is to find the shortest tour for a salesman starting from a given location, visiting each of a specified group of locations, and then returning to the departure point. For TSP, exact algorithms suffer from computational complexity that increases as a function of the number of visiting locations. Instead, heuristics provide good solutions within reasonable time, but they do not guarantee an optimal solution. A review of existing TSP heuristics is provided in [24].

A periodic version of the TSP is studied in [25] and a heuristic to compute solutions is provided. In [26], the authors study stochastic and dynamic variations of TSP where the target locations are generated stochastically; the authors then provide algorithms to compute paths for the vehicles which optimize criteria such as length of the path or the time between the generation of the target and its observation by the vehicle.

2.1.3 Patrolling

The patrolling problem consists of continuously visiting, with one or more agents, locations of interest in an area such that the time between visits to the same location is minimized.

In [27], the authors investigate heuristics to solve the patrolling problem given certain conditions. In [28], the author provides an analysis of a variety of aspects of the patrolling problem when multiple agents are involved. In [29], the authors investigate a patrolling problem where incidents occur with a known distribution in time and space; the goal of the agent is to minimize the expected wait time between the occurrence of an incident and its detection. The problem of patrolling multiple targets cooperatively is treated in [30] where the authors provide algorithms to compute trajectories for vehicles that minimize the weighted refresh time of the targets being visited. A survey of patrolling algorithms for multiple robots is provided in [31] for a variety of problems, robot types, and constraints.

In [32], a method for a patrolling agent to combat intrusion is given. The authors define the problem using a graph, where each node is a potential target and has an intrusion time indicating how much time it will take an intruder to break into the node. The method derived results in the patrolling agent visiting the nodes perpetually where the time between consecutive visits for a given node is less than its intrusion time. This work is extended to multiple patrolling agents in [33]. A problem involving unmanned aerial vehicles patrolling a set of unattended ground sensors, similar to the scenario presented in Figure 1.1, is studied in [34] where the authors use iterative learning control to select the paths of the vehicles. The authors allow the unmanned aerial vehicles to communicate with the unattended ground sensors at a distance which allows for more flexibility in the paths compared to approaches where agents must visit nodes on a graph.

While the current literature examines many implementations of persistent monitoring scenarios, no approach is constrained by observation rates at a set of locations and holds no assumptions on the periodicity of solutions. Sections 3.1 and 3.2 address this issue.

2.1.4 Target tracking

In [35], several unmanned aerial vehicles are tasked with monitoring an area with targets; the vehicles coordinate among themselves to allocate targets by maximizing the expected time over target for the entire group of vehicles. The operating area is then divided using Voronoi partitions and within each area the vehicles select time-optimal paths by constructing trajectories that satisfy the dynamic constraints of the vehicles and pass through a set of waypoints. Target tracking for unmanned aerial vehicles is also studied in [36] where the vehicles are tasked with tracking a target in an area with obstacles and threats. The authors provide a path planning algorithm for the vehicles to avoid obstacles with highest priority, maximize visibility of the target with lower priority, and minimize threat to the vehicles with lowest priority; these constraints are satisfied through the use of a probability threat exposure map that accounts for threats and obstacles in the area.

Most of the approaches discussed thus far either compute paths before the mission is executed or compute actions during the mission in turns; in some circumstances decisions may not be made a priori or in turns in which case real-time approaches are needed. One such approach is provided in [37] where the authors study real-time algorithms for path planning in the presence of obstacles.

In [38], the problem of tracking an intruder using unattended ground sensors and unmanned aerial vehicles (similar to the scenario shown in Figure 1.1) is discussed and approaches to finding paths for the unmanned aerial vehicles that maximize the number of interceptions are presented. This problem is studied further in [39] and [40] for a single unmanned aerial vehicle and an intruder that cannot retrace its steps; the authors provide sufficient conditions for the unmanned aerial vehicle to intercept the target as well as the corresponding optimal control policy for the unmanned aerial vehicle. In [41], the authors provide a Bayesian histogram filter to estimate the position of the intruder in the road network. Optimal strategies for the unmanned aerial vehicle and intruder are studied in [42] where the authors model the problem as a pursuit evasion game on a Manhattan grid and the intruder cannot retrace its steps.

Improving performance in surveillance and interception missions can also be achieved by increasing the endurance of the vehicles thus yielding more opportunities for interception, one such method is the use of energy optimal paths for aircraft equipped with solar cells [43]. The problem of estimating a vehicle's position in a graph given its velocity distribution and a previous detection at a known location and time is treated in [44]. The authors use a histogram filter to predict the vehicle's potential locations in a graph. In [45], the authors develop control techniques for a single or multiple unmanned aerial vehicles to monitor a stationary or moving target at a given standoff distance instead of intercepting the target in a decentralized manner. These techniques rely on Lyapunov vector fields to control airspeed and heading rate and are proven to be optimal and stable under certain conditions and robust to wind perturbations and uncertainty in the target motion. The approach used in Section 3.3 of this dissertation coordinates multiple mobile agents in a centralized fashion while handling uncertainty in the target motion.

In many path planning problems, computing actions which remain optimal over the entire mission duration is not feasible and thus actions which are optimal within a finite horizon are computed instead. In [46], the authors describe a nonlinear model predictive control approach for autonomous aircraft by which the dynamics are discretized in a finite horizon dictated by the plant and the goal is to select discrete parameters which achieve the given goal. The authors demonstrate this technique for an autonomous parafoil and an autonomous glider tracking a desired heading, pitch, roll, or yaw angle. Communication is an important aspect of patrolling missions, the problem of integrating sensing and communication together for information gathering tasks is treated in [47] where the authors provide a path planning algorithm for unmanned vehicles to acquire information about certain objects. Their approach utilizes an unscented Kalman filter over a finite horizon to optimize an information metric which accounts for the quality of communication links between the agents. Finite horizon approaches are also used for path planning without information gathering, in [48] a finite horizon suboptimal controller for nonlinear input-affine systems is described and demonstrated to work in a scenario involving an aircraft landing. These studies show that control while only considering a finite horizon is effective, thus a finite horizon approach for decision making is used in Section 3.3 of this dissertation.

Pursuit-evasion games occur on a graph between defenders and an intruder; the defenders win the game if the intruder is caught, otherwise the intruder wins. The defenders and intruder move in turns and can only travel to adjacent nodes. The problem was initially studied and conditions on the number of defenders necessary to guarantee capture were derived in [49], [50], and [51]. Many variations of the problem exist; such as the addition of constraints on the topology of the graph, the velocities of the defenders or intruders, the amount of information held by one team about the graph or the other team; a survey of pursuit-evasion research relevant to mobile robotics is presented in [52]. Security games introduce targets which the defenders must protect from intruder attacks, in addition the intruder can observe the defenders. In [53], the authors study and provide algorithms to solve security games where the intruder can perform actions during defender movements.

While the current literature examines many variations of patrolling problems and pursuit problems, no method treats a framework where the mobile agents fully rely on static sensors placed on an arbitrary road network to track and intercept intruders and provides a path planning algorithm for the mobile agents. Section 3.3 of this dissertation concentrates on this subject. In addition, the lack of assumptions about the topology of the network enables the handling of cases where the environment does not permit favorable sensor placement. However, when the environment is favorable and a portion of the agents used are stationary, the question of path planning for the agents turns into a question of sensor placement for the stationary agents. Literature relevant to sensor placement is now reviewed.

2.2 Sensor placement

Many methods for sensor placement exist and vary widely depending on the problem considered. In this dissertation, the methods are grouped into two general categories: graph theoretic approaches and optimization approaches. An overview of these methods is now given.

2.2.1 Graph theoretic approaches

In chemical plant flow networks, several important parameters need to be monitored to observe and control the process taking place. These parameters can be directly or indirectly measured by sensors placed in the plant. Often multiple sensor configurations to measure these parameters exist which has led to many investigations into optimal sensor placement for chemical plants. Conditions for observability in process networks are given in [54] as well as algorithms to classify observations. In [55], the authors use linear algebraic methods to construct sensor configurations which observe all the required parameters.

In [56], the problem of diagnosing faults in graph-based systems is treated and the author provides a sensor placement scheme to detect a single fault. In [57], the authors investigate sensor placement approaches for observing a target's location; the target is assumed to reside in an n-dimensional space modeled as a grid and sensors are placed on vertices of the grid. Bounds on the number of sensors necessary to be able to observe the target's location are provided and methods to determine the placement of the sensors are provided. Sensor placement is also of interest to the computer vision community where cameras are to be placed such that objects are observed; in [58] the performance of the cameras are based on quality of the resulting image of the object. The authors then define a graph on which the cameras can be placed and use a genetic algorithm to find a sequence of camera positions which observe all the objects of interest.

2.2.2 Optimization approaches

Optimal sensor placement that satisfies constraints on observing certain targets while minimizing a cost function is studied in [59]; the authors survey existing methods in process networks and provide a method to apply existing search algorithms to the optimal sensor placement problem being treated.

In [60], the authors develop a genetic algorithm for non-redundant sensor placement for linear processes and demonstrate its effectiveness. The sensors can be placed along chords of a spanning tree; cost, accuracy, or reliability of the resulting sensor network are then used to assess the fitness of a candidate solution. Genetic algorithms have also been used for other sensor placement problems such as fault diagnosis in structures; in [61] the authors use genetic algorithms and simulated annealing to place fault detecting sensors. Sensor placement problems are also of interest for water networks; in [62] the authors investigate the optimal placement of sensors to detect contaminants in a water network. The problem is formulated as a mixed-integer program and the authors show the effectiveness of the approach with respect to deviations in the probabilities of contamination within the network.

A genetic algorithm for sensor placement along general graphs is presented in [63], this approach is then applied to sensor placement in an industrial plant. The algorithm was tested on an example steam metering network and was able to find a solution within 0.05% of the optimum. In [64], the authors investigate the optimal placement of sensors for target tracking; Fisher information is used to assess the quality of the placement and a method to compute the optimal placement is provided. The authors also treat the case of mobile sensors and provide an algorithm which can compute new locations for the sensors in a decentralized fashion. Sensor placement for the localization of a single target is investigated in [65]; the authors study the optimality of the sensors' placement with regards to different sensor measurements such as range measurements and bearing measurements. The metric used to assess the quality of the sensor placement is the determinant of the Fisher Information matrix. Sensor placement for localization is also studied in [66] where the performance of the sensors is based on a lower bound of the resulting accuracy of the localization; the authors use a coordinate descent approach to solve the problem and demonstrate that it produces better results faster compared to simulated annealing in a majority of scenarios.

The problem of placing sensors to monitor a large space structure is treated in [67], the sensors are to be placed such that certain structural properties of the large space structure can be observed while minimizing the amount of sensors placed. The approach limits the redundancy of the observations received by the sensors by maximizing independent information. In [68], the authors use stochastic programming to place sensors in water networks such that uncertain demand can be handled. Another technique for sensor placement is effective independence which computes the various possible sensor locations' contribution to the overall goal. There are many other design criteria of interest such as sensor network lifetime or quality of collected information for which other placement techniques exist. In [69], a dynamic vehicle routing problem is studied where an adversary places the demand in a given area and strives to maximize the time between the placement of the demand and when it is eventually serviced by the vehicle. The authors model the problem as a complete information zero-sum game and provide the optimal routing policy for the vehicle and the optimal sensor placement and timing for the adversary.

A general sensor placement problem is studied in [70] where the sensor placement is optimized with respect to a performance metric, the authors focus their study on a convex relaxation of the problem which alleviates the intractability of the original problem for a large number of sensors. The authors show the effectiveness of their approach and give bounds on the optimality of the solutions to the relaxed problem. In [71], a target tracking problem with a large amount of sensors and multiple targets is studied; the authors demonstrate the feasibility of solving the problem with convex optimization followed by a local search algorithm.

When designing wireless sensor networks, multiple candidate sensor network configurations often exist with varying benefits and costs and as such optimal sensor placement is also studied in the wireless sensor networks literature. Surveys of recent research in the field of wireless sensor networks are provided in [72] and [73]. When deploying large networks computational complexity becomes an issue; computationally efficient methods to deploy wireless sensor networks inside buildings are investigated in [74].

Target tracking using a wireless sensor network is studied in [75]. The sensors are placed throughout the area and are enabled when the target is close and otherwise disabled to conserve energy; the authors show that their approach consumes less energy than methods for these problems and also has less error in its predictions of the target's location. Another wireless sensor network problem for target tracking is studied in [76] where the authors provide methods for multi-point surveillance and demonstrate their effectiveness with regards to tracking probability.

Network lifetime is a criterion often accounted for when designing wireless sensor networks. In [77], the authors optimize the network lifetime scaled by the number of sensors and in [78] where the network lifetime is maximized using the artificial bee colony algorithm and particle swarm optimization. Sensor placement for wireless networks can also account for the topography of the environment, such as in [79] where the authors account for line of sight coverage.

Communication is a large concern due to the networked nature of the wireless sensors, in [80] the authors investigate sensor placement configurations which maximize information acquired while minimizing the resulting communication cost of the placement. Their method includes an initial learning stage where probabilistic data on information acquisition by the sensors and communication links are gathered, this data is then used in conjunction with a model to predict future sensor quality and communication cost which in turn enables the optimal placement of the sensors. Communication can also be thought of as energy dissipation which is another area of concern for wireless sensor networks, a variety of methods exist to mitigate energy dissipation in the network; [81] contains a survey of literature addressing energy in wireless sensor networks. These topics include power management, reducing data to minimize the size of transmissions, or data sampling in energy-conservative ways. Energy management can also occur after the sensors have been placed, a heuristic that computes which sensors should be put in a sleep mode to conserve energy while the remaining sensors are used to track a number of targets is given in [82].

While sensor placement has been widely studied in the literature, sensor placement and timing for the tracking mobile objects using stationary agents that emit alarms has not been investigated. Chapter 4 of this dissertation provides methods for the selection of the locations and revisit rates of the stationary agents for a tracking scenario. A variety of methods to place sensors or stationary agents exist for a multitude of problems, however the problem of deploying mobile agents requires a different treatment. Literature relevant to marsupial agents is now reviewed.

2.3 Marsupial operations

Many mission scenarios with marsupial architectures exist; these include plans by the U.S. Air Force to release smaller passenger aircraft to fly below cloud cover or trees to gather intelligence [83], submarines that before firing torpedoes deploy a number of small decoy vehicles that emit the same signal that a torpedo firing does to minimize the likelihood of the submarine being identified and located (e.g., the mobile submarine simulator [84]), and exploration or surveillance tasks using carrier ground vehicles accompanied by aerial passenger vehicles capable of scouting the terrain ahead quickly [85].

The literature on marsupial operations focuses mostly on novel architectures for marsupial vehicles to accomplish certain tasks, while a minority of the literature focuses on how the vehicles can optimally accomplish their task(s) once the architecture has been finalized. These aspects of the marsupial operations literature are now reviewed.

2.3.1 Marsupial architectures

In [86], a variety of marsupial architectures are presented and a survey of the state of the art at the time in marsupial robotics is presented. The different marsupial architectures are divided based on the relationship the carrier vehicle has with the passenger vehicle. Several marsupial robot architectures in urban search and rescue are described in [87], [88], [89], and [90]. Tethered marsupial vehicles are studied in [91] for a variety of missions including search and rescue and the authors provide methods for constructing these marsupial vehicles such that the tether functions as desired. The problem of energy allocation and distribution within a group of marsupial vehicles is treated in [92].

In [93], the authors investigate a marsupial architecture involving a carrier underwater vehicle and passenger underwater vehicle which mimic fish; an algorithm for the passenger vehicle to track a light source while avoiding obstacles is presented and its effectiveness shown in a test where the carrier vehicle deploys the passenger vehicle and the passenger must subsequently follow the carrier vehicle. A marsupial architecture with a large amount of passenger vehicles per carrier vehicle is studied in [94]. The authors detail the hardware of the carrier and passenger vehicles and present a vision system for the docking of the vehicles. A method to direct the carrier vehicle and passenger vehicles such that a majority of the vehicles are kept fueled is provided and the effectiveness of the method is illustrated in a variety of environments. In [95] and [96], the authors describe a docking approach for the passenger vehicles based on vision. Their approach does not require communications between the passenger and carrier vehicles and is faster than docking using tele-operation without a degradation in success rate. A software suite which is capable of simulating a variety of marsupial robot architectures is presented in [97].

2.3.2 Path planning for marsupial systems

Exploration using marsupial robots is studied in [98], the goal is for the robots to reach the targets of interest while minimizing the cost of the paths traveled. The paths for the robots are computed using a temporal symbolic planning approach. In [99], the problem of path planning for the carrier vehicle is treated. Multiple passenger vehicles are present throughout a space and the carrier vehicle selects a path such that it is capable of refueling all the passenger vehicles before their fuel runs out; the computation of the path is achieved using temporal symbolic planning.

The question of how to dock marsupial vehicles is also studied in [100], the authors describe several existing approaches to docking and introduce their own method based on minimizing energy consumed during the docking phase. Optimal dispersion of passenger vehicles by a carrier vehicle for a search task is studied in [101]. The goal of the passenger vehicles is to search the area as quickly as possible; the dispersion approach involves dividing the area, the carrier vehicle deploying the passenger vehicles within their respective areas, and the passenger vehicles searching once they are deployed or receive a certain signal from the carrier vehicle.

A significant portion of the literature relevant to marsupial operations focuses on novel mission architectures and the hardware necessary for the marsupial vehicles to operate, dock, and undock; Chapter 5 of this dissertation instead focuses on the path planning aspect of the marsupial agents for constrained optimization problems and allows for the enforcement of docking and undocking constraints.

2.4 Intractability

Assuming a problem is well-posed and possesses a solution; the solution must then be computed. The speed of the computation of the solution will depend on the complexity of the problem, i.e., the more complex a problem is the longer the computation of the solution will take. Problems for which the computation of the solution take an impractical large amount of time are known as intractable problems. In the following subsections, complexity is formally defined, complexity classes of problems are defined, and methods of overcoming complexity are discussed. In this section, only complexity with respect to time will be treated (a review of the complexity of an algorithm with respect to space is contained in [102]).

2.4.1 Complexity

The complexity of an algorithm quantifies the amount of operations the algorithm must perform based on its input before completion. It is usually denoted in terms of *big O notation* where O stands for order; this notation yields an approximation of the order of magnitude of the amount of computations the algorithm will require. For example, an algorithm that reads an entire word would be $O = n$ for a word with n letters, i.e., the amount of computations required for this algorithm will be of the order n.

2.4.2 Complexity class

In the previous subsection, the complexity of an algorithm was defined. This is related to the complexity of a problem in the following manner, the complexity of a problem defines a lower bound on the complexity of algorithms devised to solve the problem. Before complexity classes for problems can be defined, a model for computation is needed. The Turing machine is a model for computation in which a tape is read letter by letter and the machine decides how to process the letter (keep or erase), how to alter its state, and which letter to read next; [103] contains a more detailed description of a Turing machine. A Turing machine which reads only and always read to the right can be thought of as a deterministic finite state automaton. Similarly, if the transitions in state are non-deterministic than the Turing machine is a non-deterministic Turing machine.

Formal definitions of complexity are given in [104] and [105], what follows is a brief overview. Problems can be classified based on their complexity, as was discussed earlier problems are distinguished by the algorithms used to compute their solution. Problems can be first classified based on the type of order the lower bound complexity algorithm which solves them is:

- deterministic Polynomial time (P)
- Non-deterministic Polynomial time (NP)

For example, a problem which belongs to the deterministic polynomial-time complexity class can be solved in polynomial time by a deterministic Turing machine while a problem which belongs to NP can be solved in polynomial time only by a non-deterministic Turing machine. The NP class of problems are problems whose solution can be verified in polynomial-time, which implies a non-deterministic Turing machine to solve the problem can be constructed by building all the possible solutions and then verifying each with a deterministic Turing machine.

Within these classes of problems, there are varying levels of difficulty. To define these varying levels of difficulty, the satisfiability problem must first be defined: given a set of clauses, is the conjunction of the given clauses satisfiable? Reducibility must also be defined, informally problem X is said reducible to problem Y if a mapping exists from inputs of problem X to inputs of problem Y such that a solution to problem Y with the mapped inputs implies a solution exists to problem X with the original inputs. Problems for which the satisfiability problem can be reduced to are considered NP-hard. Problems which are NP-hard and also in NP are NP-complete. The NP-complete class can be considered the hardest problems to solve in NP while the NP-hard class can be considered problems which are as difficult to solve as the hardest problems in NP.

2.4.3 Overcoming complexity

The intractability of a problem is a fundamental property of the problem, that is no efficient algorithm to solve the problem can be devised. However, heuristics, algorithms for which no guarantees about the solution are made, can be devised which in practice may yield good solutions frequently. For example if dealing with an optimization problem where one desires the best solution; a heuristic may provide a solution that is usually within a certain range of optimal. One such heuristic is the k-opt heuristic, which takes an initial candidate solution and searches the solution space which can be reached by making k switches to the elements of the solutions for a better solution. This process can be iterated a given number of times or till the solution has converged. The traveling salesman problem is demonstrated to be in NP-complete in [105]; an implementation of k-opt for TSP is presented in [106] and the effectiveness of the method is shown.

2.5 Research directions

The literature survey above shows that much work has been performed on monitoring using unmanned systems and related topics. However, several questions remain unanswered. Most of the approaches in the patrolling literature constrain the solutions to be periodic though the problem statement does not necessarily require it; deadlines between consecutive visits to locations lead to increased flexibility in the solution and often satisfy the problem requirements as well. In the pursuit literature, there exist approaches to pursuing evaders in graphs however the case of agents fully relying on stationary sensors to pursue the evaders along arbitrary graphs has not been studied as extensively. While there has been much work on the hardware issues stemming from agents using a marsupial architecture, little work has been done on optimal path planning for marsupial agents performing information collection missions . The questions below pertain to the main questions listed in Section 1 and treat the stated gaps in the literature.

- Which persistent paths for the agents guarantee proper monitoring of the area using a revisit deadline framework? (Path planning)
  - How can these paths be computed?
  - How difficult is the computation of these paths?
  - When do solutions exist?
- How can mobile agents and stationary agents cooperate to track a mobile object of interest?
  - How can the paths for the mobile agents be computed? (Path planning)
  - How should the stationary agents be configured to best track the mobile object? (Sensor placement)
- How can marsupial agents cooperate when collecting information about objects of interest? (Marsupial operations)

- What are the optimal paths for the agents?

- What properties do these paths have?

CHAPTER 3

Paths for mobile agents

In this chapter, path planning for mobile agents performing monitoring tasks is treated. The results from this chapter are used to address the path planning question stated in Section 1.2: *how can mobile agents cooperate to achieve monitoring tasks?*

First, the problem of a single vehicle without fuel constraints monitoring a number of objects of interest is studied. Second, the agent consumes fuel and fuel depots that enable the agent to refuel for a given cost are considered. The problem is then extended to multiple mobile agents monitoring stationary agents and pursuing a mobile object. The paths that solve these problems are presented, properties of these paths are shown, and methods to compute these paths are discussed.

3.1 Monitoring stationary objects with no fuel constraints

In this section, a vehicle that travels without consuming fuel is monitoring multiple stationary objects. Paths that monitor the objects successfully are given and their properties are discussed. Heuristics to compute the vehicle's path are presented and their effectiveness illustrated.

3.1.1 Model

The mobile agent is assumed to be traveling at a constant velocity, v, in a two dimensional space. There are n objects of interest in the space where each object of interest has Cartesian coordinates (ξ_i,ζ_i) and a finite revisit deadline $r_i \in \mathbb{R}$. The finite revisit deadline indicates that the time between consecutive visits to object of interest i must be less than or equal to r_i where $r_i > 0$. Let $d_{i,j}$ be the distance between nodes i and j.

3.1.1.1 Revisit deadlines

In persistent intelligence, surveillance, and reconnaissance scenarios, an object of interest's properties can change over time and the mission designer is to track these changes. These properties may be observable by the agent directly or indirectly, in which case an inference system is required. We assume that the mission designer has an internal model of how the properties of a given object of interest are expected to change and rates at which they are expected to change. Based on this knowledge, a revisit deadline can be derived which leads to tracking of the state of these properties over time. Methods for this derivation are not discussed in this dissertation: results from sampling in signal processing to reconstruct a signal can be used such as the Nyquist-Shannon sampling theorem [107].

3.1.1.2 Path planning

The mobile agent's kinematic model is based on the unicycle vehicle model [26]. The objects of interest are assumed to be far from one another (isolated) and to be poorly visible by the mobile agent. Thus, according to Proposition 1 and its corollary in [108], the optimal flight path of the mobile agent consists of straight lines between the objects of interest. The path planning problem can thus be treated much like a traveling salesman problem.

3.1.1.3 State space model

The situation is modeled as follows:

$$y(k+1) = f(y(k), u(k)), \tag{3.1}$$

$$\dot{x}(t) = g(t, y(k)), \tag{3.2}$$

where $y(k) \in \{1, 2, ..., n\} \times \mathbb{R}$ and $x(t) \in \mathbb{R}^n$. Specifically,

$$y(k) = \begin{bmatrix} p(k) & \tau(k) \end{bmatrix}^T, \tag{3.3}$$

where $p(k) \in \{1, 2, ..., n\}$ indicates which object of interest the mobile agent is visiting upon completion of step k, where a step is the act of the mobile agent traveling from one object of interest to the next, and $\tau(k) \in \mathbb{R}$ indicates the total time elapsed upon completion of step k; and

$$x(t) = \begin{bmatrix} x_1(t) & \cdots & x_n(t) \end{bmatrix}^T, \tag{3.4}$$

where $x_i(t), 1 \leq i \leq n$ is the slack time of object of interest i, which indicates how much longer the mobile agent can wait before a visit to object of interest i is overdue.

3.1.1.4 Initial conditions

Without loss of generality, we assume that the mobile agent starts at the first object of interest, i.e.,

$$\begin{aligned} p(0) &= 1, \\ \tau(0) &= 0, \\ x_1(0) &= r_1, \\ x_2(0) &= r_2, \\ &\vdots \\ x_n(0) &= r_n. \end{aligned} \tag{3.5}$$

3.1.1.5 Dynamics

The dynamics of the visitation schedule p and the total time elapsed τ are modeled as

$$p(k+1) := u(k), \tag{3.6}$$

$$u(k) \neq p(k), \tag{3.7}$$

$$\tau(k+1) := \tau(k) + \frac{d_{u(k),p(k)}}{v}, \tag{3.8}$$

where $d_{u(k),p(k)}$ is the distance from the agent's current location to its next destination and v is the agent's velocity. The dynamics of the slack time for object of interest i is given as,

$$\dot{x}_i(t) = -1 + \sum_{j=1}^{k} \delta_{ip(j)} \cdot (r_i - x_i(\tau(j))) \cdot \delta(t - \tau(j)), t \leq \tau(k), 1 \leq i \leq n, \tag{3.9}$$

where $\delta(t - \tau(j))$ is the continuous Dirac delta function and $\delta_{ip(j)}$ is the discrete Kronecker delta function. In (3.9), the slack time of each object decreases with time and is reset to the revisit deadline when the object is visited.

3.1.2 Problem formulation

Based on the model above, we formulate the persistent visitation problem as follows: the mobile agent is to find a sequence $u(k)$, $k \in \mathbb{N}$ such that under equations (3.5), (3.6), (3.7), (3.8), (3.9), $\forall k, \forall i \in \{1,2,...,n\}, \forall t \geq 0, x_i(t) \geq 0$.

3.1.3 Solutions & periodicity

3.1.3.1 Solutions

A *solution* to the persistent visitation problem is defined as an infinite sequence of visitations such that no visitation is ever overdue.

Definition 3.1.1. *A cycle is defined as a sequence of visitations starting from one object of interest and ending at that same object of interest such that all other objects of interest have been visited.*

Thus a solution consists of an infinite sequence of cycles.

3.1.3.2 Existence of almost periodic solutions

Remark 3.1.2. *Viewing a solution as an infinite sequence of cycles, the initial slack times of these cycles belong to the compact set* $[0,r_1] \times [0,r_2] \times ... \times [0,r_n]$.

Theorem 3.1.3. *If a solution exists, then an almost periodic solution also exists.*

Proof. If a solution exists, then an infinite sequence of cycles exists. Each cycle has a set of initial slack times, hence if a solution exists, there exists an infinite sequence of initial slack times of cycles. The initial slack times of cycles belong to a compact set (Remark 3.1.2), thus the sequence of initial slack times of cycles is bounded. The Bolzano-Weierstrass Theorem states that *every bounded sequence has a convergent subsequence* [109]. Hence a convergent subsequence can be extracted from the infinite sequence of initial slack times of cycles. Let T_i be the i^{th} cycle in the solution and $\tau_{end}(T_i)$ be the time at the end of the cycle T_i, thus

$$\forall \epsilon \in \mathbb{R} > 0, \exists N(\epsilon) \in \mathbb{N} \text{ such that } \forall i \geq N(\epsilon), \|x(\tau_{end}(T_i)) - x(\tau_{end}(T_{i+1}))\| < \epsilon. \quad (3.10)$$

Thus a solution can be constructed consisting of infinite repetitions of the cycle T_i where $i \geq N$. Therefore, an almost periodic solution exists. □

3.1.4 Complexity

In this subsection, a proof that the persistent visitation problem belongs to the NP-complete class of problems is provided.

3.1.4.1 NP

A proof that the persistent visitation problem belongs to the NP class is now provided. The proof is in the form of providing a deterministic polynomial-time Turing machine verifier for the persistent visitation problem.

Lemma 3.1.4. *The persistent visitation problem belongs to the NP class of problems.*

Proof. A language L is in NP if there exists a deterministic polynomial-time Turing machine V_L such that given any w, there exists a string c such that $x \in L$ exactly when $V_L(w, c)$ accepts [102]. In this work c is an encoding of a candidate solution path and w is an encoding of the weighted digraph G, representing the objects of interest, their transit times, and the set of revisit deadlines. Solution paths are viewed as sequences of edges, where the alphabet Σ is the set of directed edges. The machine V_L is modeled as a deterministic finite automaton (DFA), D. The states of the machine, Q, consist of accepting states, F, that represent all possible combinations of positive slack times for the objects of interest and a single fail state, R, that represents states with negative slack times. Thus $Q = F \cup R$. The initial state $q_0 \in F$ indicates when all the slack times are equal to their respective revisit deadlines, $q_0 = [r_1\ r_2\ ...\ r_n]$. Let Δ contain all the transitions between the states in Q. This DFA $D = (Q, \Sigma, \Delta, q_0, F)$ reads a sequence of edges and makes the appropriate transitions in slack time states. D rejects a sequence if a negative slack time occurs during the processing of the sequence and accepts it otherwise. D can be generated using Algorithm 1. D is equivalent to a read-only right moving Turing machine with a polynomial time bound; for each input x the bound is $|x|$. Therefore, the persistent visitation problem is NP. □

3.1.4.2 NP-complete

A proof that the persistent visitation problem belongs to the NP-complete class of problems is now provided. We introduce the finite horizon persistent visitation problem and formulate the problem as follows: given a finite horizon $h \in \mathbb{R}$, the mobile agent is to find a finite sequence $u(k)$, such that $\tau(|u|-1) \leq h$, $\tau(|u|) > h$, and under equations (3.5), (3.6), (3.7), (3.8), (3.9), $\forall k, \forall i \in \{1, 2, ..., n\}, h \geq t \geq 0, x_i(t) \geq 0$.

We make use of the metric traveling salesman problem which is proven to be NP-complete in [110]. The metric traveling salesman problem is formulated as follows: given a finite set $C = \{c_1, c_2, ..., c_n\}$ of cities, a distance metric $d(c_i, c_j) \in \mathbb{Z}^+$ for each pair of cities $c_i, c_j \in C$, and a bound $B \in \mathbb{Z}^+$, is there a tour of all the cities in C having total length no more than B?

Lemma 3.1.5. *The finite horizon persistent visitation problem is NP-complete.*

Proof. A language L is NP-complete if $L \in$ NP and another NP-complete problem is reducible to L [102]. The metric traveling salesman problem can be reduced to the finite horizon persistent visitation problem as follows: $r_i = B$, $h = B$, $v = 1$, and $d_{i,j} = d(c_i, c_j)$. The first $(|u|-1)$ cities of the solution to such a finite horizon persistent visitation problem consist of a cycle visiting all cities such that the total distance elapsed is less than or equal to B. This cycle may contain multiple visits to the same city, however since the distance function used is a metric it satisfies the triangle inequality and thus any repeat visits in the cycle can be removed without increasing the distance traveled. Thus if a sequence of visits satisfying this finite horizon persistent visitation problem exists then a tour satisfying the metric traveling salesman problem exists. Hence the finite horizon persistent visitation problem is NP-hard. In addition, the finite horizon persistent visitation problem belongs to the NP class of problems (Lemma 3.1.4), therefore the finite horizon persistent visitation problem is NP-complete. □

Theorem 3.1.6. *The persistent visitation problem is NP-complete.*

Proof. The persistent visitation problems belongs to the NP class of problems (Lemma 3.1.4) and the finite horizon persistent visitation problem is NP-complete (Lemma 3.1.5). A solution to the persistent visitation problem is a solution to the finite horizon persistent visitation problem, thus the finite horizon persistent visitation problem is reducible to the persistent visitation problem. Hence the persistent visitation problem is NP-complete. □

3.1.5 Heuristics

The persistent visitation problem is NP-complete, thus efficient algorithms to solve the problem do not exist. While an algorithm to compute cycles that solve the persistent visitation problem does exist [32], it quickly becomes computationally prohibitive to compute a solution for large instances of the problem. We thus present heuristics as an alternative that can be used to find solutions to the persistent visitation problem when searching for cycles is not a feasible option. While computationally advantageous, heuristics may not be complete, i.e., they are not guaranteed to find a solution even if one exists. The effectiveness of the heuristics is shown through numerical examples where plots of the slack times of the stationary objects as a function of time are given. All plots presented in this section use dimensionless time (scaled by the minimum revisit deadline). These plots are provided for multiple problem configurations where a problem configuration is comprised of the mobile agent's velocity as well as the locations and revisit deadlines of the objects of interest. The configurations of all the examples used in this section can be viewed in Table 3.1.

Algorithm 3.1: Generate solution verifier for a persistent visitation problem

```
**Data**: Edges: $\{e_{1,2}, e_{1,3}, ..., e_{1,n}, e_{2,1}, ..., e_{n,n-1}\}$
Revisit deadlines: $\{r_1, r_2, ..., r_n\}$
Transit times from node $i$ to node $j$: $t_{i,j}$
1  $\Sigma \leftarrow \{e_{1,2}, e_{1,3}, ..., e_{1,n}, e_{2,1}, ..., e_{n,n-1}\}$
2  $\begin{bmatrix} q_0 \\ \alpha \end{bmatrix} \leftarrow \begin{bmatrix} r_1 & r_2 & ... & r_n \\ 1 & 1 & ... & 1 \end{bmatrix}$
3  $\hat{F} \leftarrow \{q_0\}$
4  $F \leftarrow \emptyset$
5  $\Delta \leftarrow \emptyset$
6  for $q \in \hat{F}$ do
7      for $i \leftarrow 1 : n$ do
8          if $q(i) = r_i$ then
9              for $j \leftarrow 1 : n\ ,\ j \neq i$ do
10                 $q' \leftarrow q - t_{i,j} \cdot \alpha$
11                 if $q'(1) < 0 \vee q'(2) < 0 \vee ... \vee q'(n) < 0$ then
12                     $\delta' \leftarrow (q, e_{i,j}, R)$
13                     $\Delta \leftarrow \Delta \cup \{\delta'\}$
14                 else
15                     $q'(j) \leftarrow r_j$
16                     if $\{q'\} \cap F = \emptyset \wedge \{q'\} \cap \hat{F} = \emptyset$ then
17                         $\delta' \leftarrow (q, e_{i,j}, q')$
18                         $\Delta \leftarrow \Delta \cup \{\delta'\}$
19                         $\hat{F} \leftarrow \hat{F} \cup \{q'\}$
20     $\hat{F} \leftarrow \hat{F} \setminus \{q\}$
21     $F \leftarrow F \cup \{q\}$
22 $D \leftarrow \{\{F \cup R\}, \Sigma, \Delta, q_0, F\}$
**Result**: Persistent visitation solution verifying DFA: $D$
```

Config. Data	Example A	Example B	Example C
v	6.89411	15.48131	41.69983
(ξ_1,ζ_1,r_1)	(50.389,64.681,18.465)	(32.757,67.126,26.319)	(16.565,60.198,15.778)
(ξ_2,ζ_2,r_2)	(13.872,47.557,21.748)	(83.35,76.885,10.035)	(65.408,68.921,44.889)
(ξ_3,ζ_3,r_3)	(78.811,78.03,40.111)	(86.198,98.987,30.865)	(45.054,8.3821,13.739)
(ξ_4,ζ_4,r_4)		(88.428,58.803,9.2851)	(91.334,15.238,49.549)

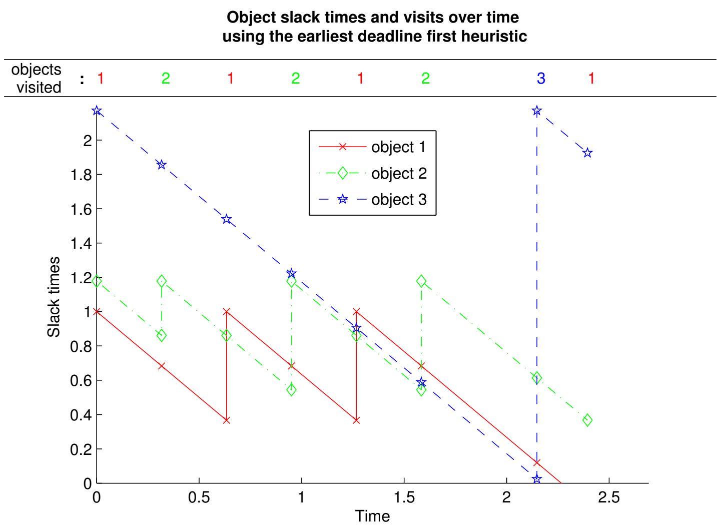
Table 3.1: Configurations of examples used

3.1.5.1 Earliest deadline first

We start by studying a heuristic that does not search ahead, i.e., a heuristic that selects a destination based on the current state. The earliest deadline first heuristic selects the object of interest with the minimum slack time as its next destination:

$$u(k) := \underset{1 \leq i \leq n}{argmin}\{x_i(k)\}. \tag{3.11}$$

In classic scheduling problems, the earliest deadline first heuristic is complete (proven in [111]); however this is not true in the persistent visitation problem. A counterexample is shown in Figure 3.1; where earliest deadline first fails after six steps while the two step maximum minimum slack time heuristic (presented in Section 3.1.5.3) is able to satisfy the revisit deadlines on the same problem configuration.



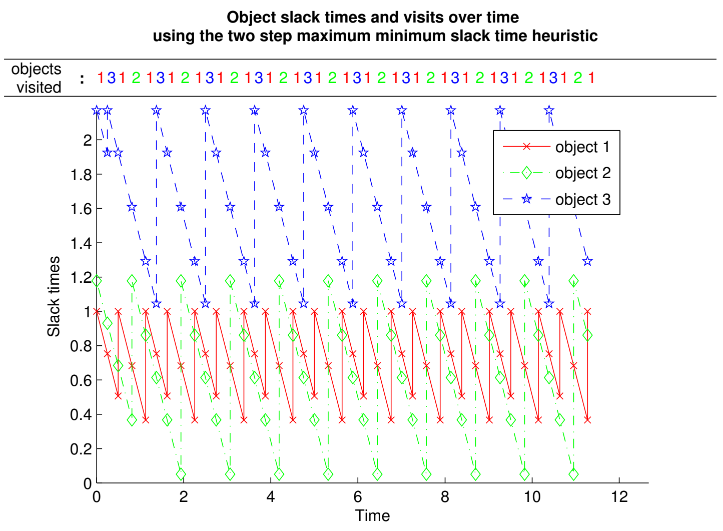


Figure 3.1: Failure of earliest deadline first heuristic and success of two step maximum minimum slack time heuristic on example A.

3.1.5.2 One step heuristics

In this section, we present heuristics that search one step ahead. We introduce the following intermediate variable, A, representing the time spent in transit during the step:

$$A(k,i) = \frac{d_{p(k),i}}{v}. \tag{3.12}$$

Figure 3.2 illustrates that the one step heuristics presented are incomplete; in this example earliest deadline first could solve the problem but the one step heuristics failed.

Maximum minimum slack time: The maximum minimum slack time heuristic selects as its next destination the object of interest that maximizes the minimum slack time:

$$u(k) := \underset{1 \leq i \leq n, i \neq p(k)}{argmax} \{ \underset{1 \leq j \leq n}{min} \{ x_j(k) - A(k,i) \} \}. \tag{3.13}$$

Maximum sum of slack times: The maximum sum of slack times heuristic selects as its next destination the object of interest that maximizes the sum of the slack times:

$$u(k) := \underset{1 \leq i \leq n, i \neq p(k)}{argmax} \{ \sum_{j=1}^{n} (x_j(k) - A(k,i)) \}. \tag{3.14}$$

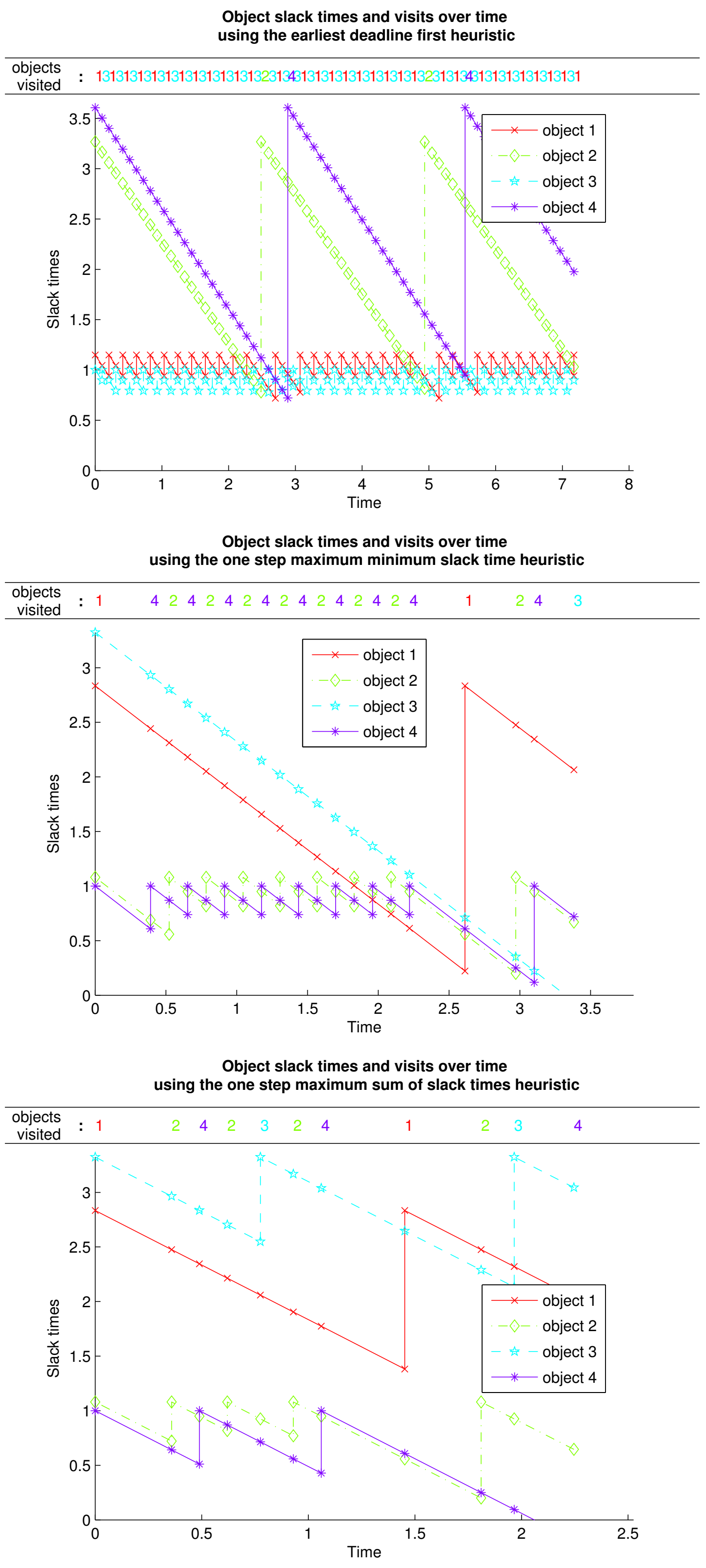


Figure 3.2: Earliest deadline first success and failure of one step heuristics on example B.

3.1.5.3 Two step heuristics

In this section, we present heuristics that search two steps ahead. We introduce the following intermediate variable, B, representing the time spent in transit during the second step:

$$B(i,l) = \frac{d_{i,l}}{v}. \tag{3.15}$$

The two step heuristics presented are shown to be incomplete in Figure 3.3, where they fail on a problem instance which earliest deadline first could solve (illustrated in Figure 3.2).

Maximum minimum slack time: This heuristic searches for the path that maximizes the minimum slack time across the two steps and selects the first step in the resulting path as its next destination:

$$u(k) := \underset{1 \le i,l \le n, i \ne p(k), l \ne i}{argmax} \left\{ \underset{1 \le j \le n}{min} \left(\{x_j(k) - A(k,i)\}, \{r_i - B(i,l)\}, \{x_j(k) - A(k,i) - B(i,l), j \ne i\} \right) \right\}. \tag{3.16}$$

Maximum sum of slack times: This heuristic searches for the path that has the maximum sum of slack times at the end of the two steps; it selects the first step in that path as its next destination:

$$u(k) := \underset{1 \le i,l \le n, i \ne p(k), l \ne i}{argmax} \left\{ r_i - B(i,l) + \sum_{j=1, j \ne i}^{n} \left(x_j(k) - A(k,i) - B(i,l) \right) \right\}. \tag{3.17}$$

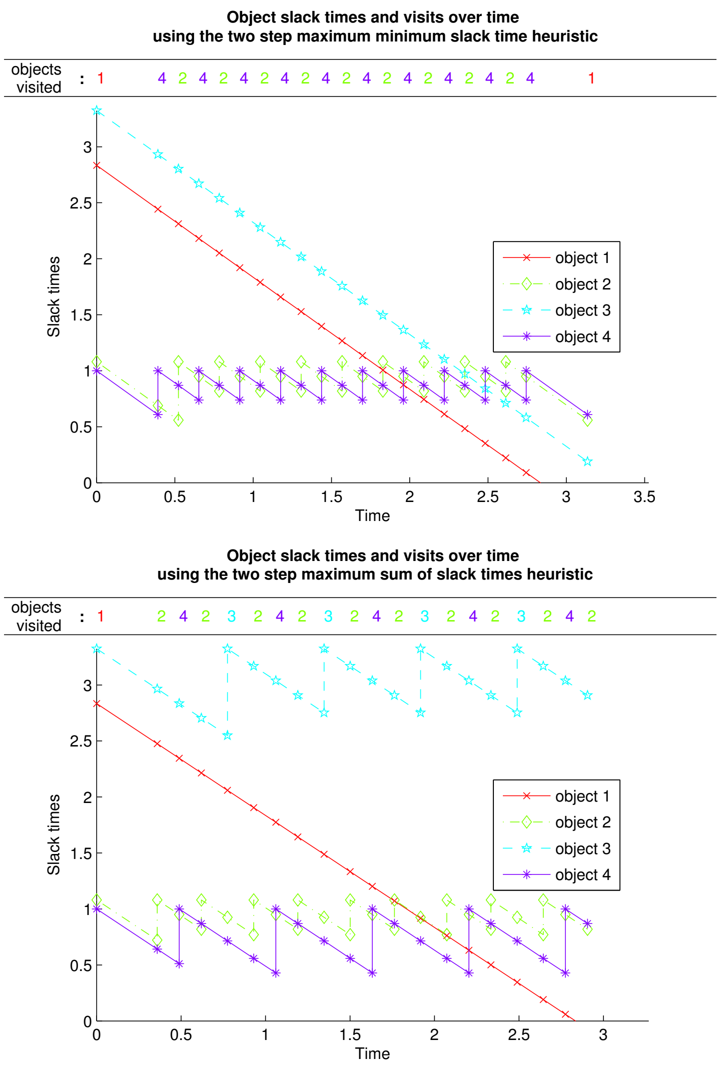


Figure 3.3: Failure of two step heuristics on example B.

3.1.5.4 K-step heuristics

The maximum minimum slack time and maximum sum of slack times heuristics can be extended to search depths greater than two. However, increasing the depth of the heuristics' search increases the computation time. The heuristics presented above are all incomplete. While this remains true for earliest deadline first, the search depth of the maximum minimum slack time and maximum sum of slack time heuristics affects their completeness.

In [32], the authors prove that if there exists a cycle that solves the persistent visitation problem then there exists a cycle with an upper bound on its temporal length that can solve the problem. Once the heuristics are searching a number of steps equivalent to that upper bound in time, the algorithm in [32] can solve the problem with similar computational complexity. This upper bound on temporal length is $max(r_i)$, thus k-step search ahead heuristics are complete if $k > \frac{v \cdot max(r_i)}{min(d_{i,j})}$.

3.1.6 Simulations

The different heuristics presented highlight the fact that multiple solutions exist to the same problem (illustrated in Figure 3.4). The distinct goals of the heuristics lead to different characteristics in their solutions. The earliest deadline first heuristic goes to the object of interest with the minimum slack time; this can be seen by observing the minimum slack time and the next object visited. The maximum minimum slack time heuristic raises the minimum slack time which in this example leads to a solution similar to that of the earliest deadline first heuristic. The maximum sum of slack times heuristic's goal is to raise the sum of the slack times and thus this heuristic visits the objects more often than the other heuristics.

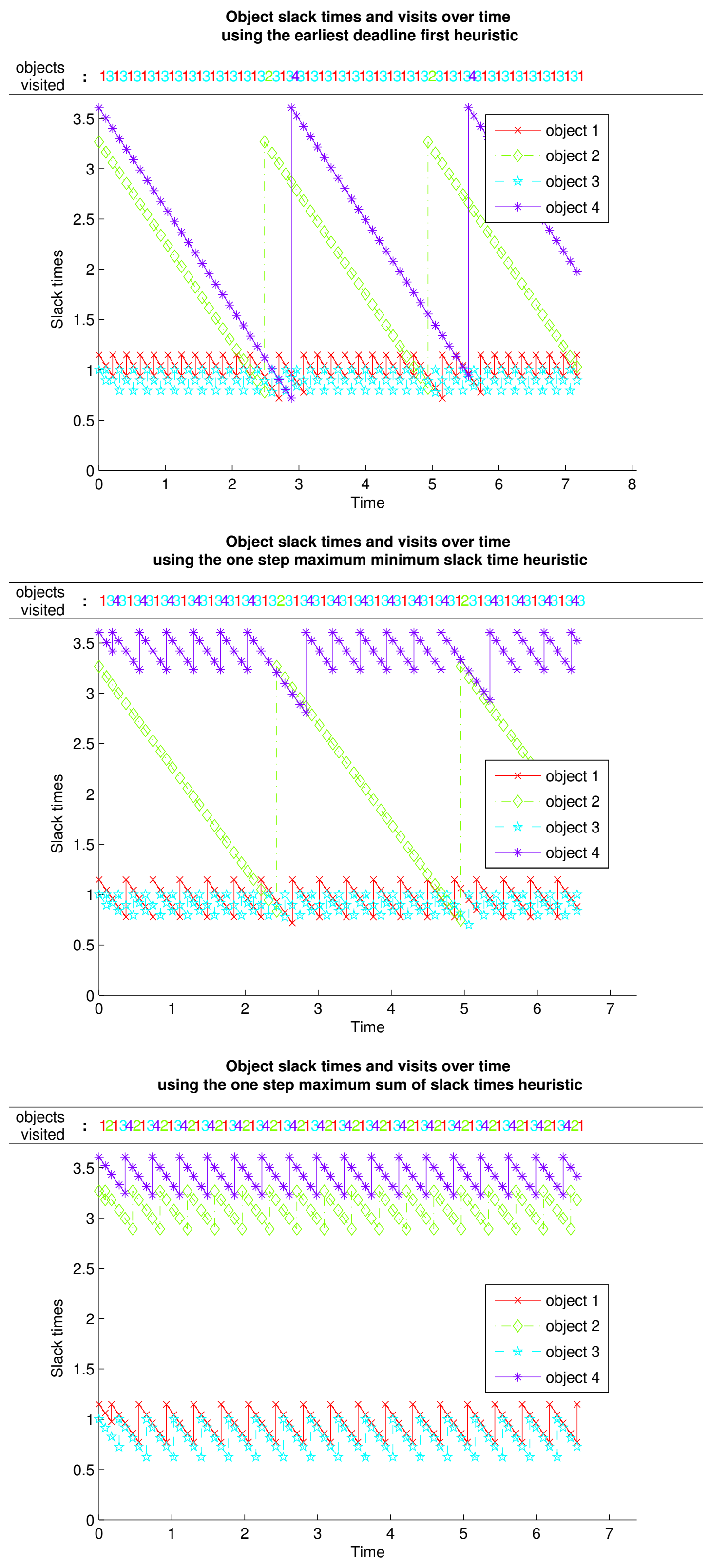


Figure 3.4: Performance of zero and one step heuristics on example C.

The solutions in Figure 3.4 illustrate that the slack times settle into a periodic regime as predicted in Theorem 3.1.3. This behavior is not unique to the problem configuration used for Figure 3.4 but has been observed in all solutions obtained from the heuristics presented. However problem instances with aperiodic solutions were also found.

3.2 Monitoring stationary objects with fuel considerations

In the previous section, a single vehicle that does not consume fuel is tasked with monitoring a number of stationary objects. In this section, the vehicle now consumes fuel at a finite rate. Fuel depots are present in the area, where the vehicle can refuel at a cost. Different fuel depots may have different refueling costs, hence the problem is now to find a path for the vehicle that still monitors the stationary objects but also minimizes the expenditures on fuel over the duration of the mission. Properties of paths are discussed later in this section and an algorithm to compute the optimal path is presented.

3.2.1 Model

In this subsection, the modeling for this problem is discussed. The model used when treating this problem is a variant of the model used when treating the problem for a single vehicle without fuel constraints (see Section 3.1.1). The vehicle has a finite fuel capacity, F, and consumes fuel at a finite constant rate of $\dot{f}_c$ where $\dot{f}_c > 0$; the vehicle endurance is thus $r_f = (F/\dot{f}_c)$. There are q fuel depots, each with Cartesian coordinates (ξ_i, ζ_i), and a finite constant positive fuel cost $c_i \in \mathbb{R}, n < i \leq n+q$. The fuel depots store an infinite amount of fuel. The vehicle can select how much fuel to purchase when visiting a fuel depot. Refueling is assumed to have a negligible duration in comparison to the time spent in transit.

3.2.1.1 State space model

A fuel state, $f(t) \in \mathbb{R}$, which is the amount of fuel the vehicle is carrying at time $t \in \mathbb{R}$, is added. An extra state, the mission state $m(t) \in \mathbb{R}^{n+1}$, is created that contain all the continuous states, i.e., the slack times and fuel state:

$$m(t) = \begin{bmatrix} x_1(t) & x_2(t) & \cdots & x_n(t) & f(t) \end{bmatrix}^T . \tag{3.18}$$

The input for this problem also changes because the amount of fuel purchased needs to be accounted for. The input is now $u(k) \in \{1, 2, \cdots, n+q\} \times \mathbb{R}$; it contains which customer or depot the vehicle visits next and how much fuel the vehicle purchases if the next visit is a depot visit (k again indicates the step at which this input is applied). A second index, $l \in \{1, 2\}$, is introduced to indicate the specific input: $l = 1$ indicates that $u(k, l) \in \{1, 2, \cdots, n+q\}$ corresponds to the visiting location and $l = 2$ indicates that $u(k, l) \in \mathbb{R}$ corresponds to the fuel amount being purchased. Fuel can only be purchased at fuel depots thus:

$$u(k, 1) \leq n \Rightarrow u(k, 2) = 0. \tag{3.19}$$

3.2.1.2 Initial conditions

The initial conditions for the state space model introduced in the previous subsection are as follows:

$$\begin{aligned} p(0) &= u(1,1), \\ \tau(0) &= 0, \\ x_1(0) &= r_1, \\ x_2(0) &= r_2, \\ x_n(0) &= r_n, \\ f(0) &= f_0. \end{aligned} \tag{3.20}$$

The amount of fuel the vehicle carries initially, f_0, is an additional input to the model.

3.2.2 Dynamics

The dynamics of the total time elapsed, $\tau(k)$, and the slack times, $x_i(t)$, remain the same for this problem (see (3.8) and (3.9) respectively). The dynamics of the visitation schedule, p, are:

$$p(k+1) := u(k,1), \tag{3.21}$$

$$u(k,1) \neq p(k). \tag{3.22}$$

The dynamics of the fuel carried by the vehicle are:

$$\dot{f}(t) = -\dot{f}_c + \sum_{j=1}^{k} u(j,2) \cdot \delta(t - \tau(j)), t \leq \tau(k). \tag{3.23}$$

3.2.3 Periodicity

In Section 3.1, where the problem without fuel considerations is treated, almost periodic paths are demonstrated to exist. In this subsection, we show that this property also holds when considering fuel.

A path is defined as an infinite sequence of visitations such that no stationary object visitation is ever overdue and the vehicle's fuel is always bounded by zero and the fuel tank size.

Definition 3.2.1. *A tour is defined as a sequence of visitations starting from one stationary object and ending at that same stationary object such that all other stationary objects have been visited.*

A path can thus be expressed as an infinite sequence of tours. The following remark and lemmas are used to prove the existence of almost periodic paths.

Remark 3.2.2. *Viewing a path as an infinite sequence of tours, the initial mission state of these tours belongs to the compact set* $[0, r_1] \times [0, r_2] \times \cdots \times [0, r_n] \times [0, r_f]$.

Lemma 3.2.3. *If a path exists, then a path with an almost periodic mission state exists.*

Proof. If a path exists, then an infinite sequence of tours exists. Each tour has an initial mission state; hence if a path exists, there exists an infinite sequence of initial mission states of tours. The initial mission state of tours belongs to a compact set (Remark 3.2.2), thus the sequence of initial mission states of tours is bounded. The Bolzano-Weierstrass Theorem states that *every bounded sequence has a convergent subsequence*. Hence a convergent subsequence can be extracted from the infinite sequence of initial mission states of tours. Let τ_i be the i^{th} tour in the path, thus

$$\forall \epsilon \in \mathbb{R}^{n+1} > 0, \exists N(\epsilon) \in \mathbb{N} : \forall i \geq N(\epsilon), \|m_{final_{\tau_i}} - m_{final_{\tau_{i+1}}}\| < \epsilon. \tag{3.24}$$

Thus a path can be constructed consisting of infinite repetitions of the tour where . Therefore, a path with an almost periodic mission state exists. □

Lemma 3.2.4. *If no depot or stationary object is equidistant to two other depots, then there exists a unique sequence of visitations describing the mission state on* $[t_0, t_1]$ *where at least one visit to a stationary object occurs in* $[t_0, t_1]$.

Proof. Impulses in a slack time in the mission state are uniquely described by visits to the stationary object the slack time represents. Thus visits to stationary objects can be extracted from the impulses slack times in the mission state; this is called labeling. Thus all slack time impulses can be labeled directly. Impulses in the amount of fuel in the mission state are not uniquely described by visits to a certain fuel depot as multiple depots exist thus they cannot be uniquely labeled directly. Because no stationary object is equidistant to two depots, the time in transit from any stationary object to a depot is unique. Thus the time between an impulse in a slack time and an adjacent impulse in the amount of fuel uniquely determines which depot visit caused the impulse in the amount of fuel. Thus impulses in the amount of fuel adjacent to an impulse in a slack time can be labeled. No depot is equidistant to two other depots, thus the time in transit from a depot to another is unique. Therefore, an impulse in the amount of fuel adjacent to a labeled impulse in the amount of fuel can be labeled. Hence all impulses occurring in the elements of the mission state on a given time interval can be labeled if at least one slack time impulse occurs within the time interval. □

Theorem 3.2.5. *If a path exists and no depot or stationary object is equidistant to two other depots, then an almost periodic path exists.*

Proof. If a path exists, then according to Lemma 3.2.3, a path with an almost periodic mission state exists. Let $P(t)$ be the mission state during a period. In $P(t)$ at least one visit to a stationary object occurs since for a slack time to return to a higher value an impulse must occur, and no depot or stationary object is equidistant to two other depots, thus according to

Lemma 3.2.4 $P(t)$ is described by a unique sequence of visitations. Therefore, the sequence of visitations of the path with an almost periodic mission state is almost periodic. □

Depots can be moved by infinitesimal amounts to circumvent the issue of equidistant depots or stationary objects and guarantee the existence of tours. So far the existence of tours that keep mission states within their bounds has been shown; now the quantity of such tours is discussed.

Theorem 3.2.6. *There exists a finite number of tours that start at stationary objects that if infinitely repeated maintain the mission states within their bounds.*

Proof. The time in transit from one location to another is finite, thus a finite sequence of visitations takes finite time to complete. The duration of a tour that starts at a stationary object is constrained by the revisit deadline of the starting stationary object. There are a finite number of combinations of visits that can be achieved within a finite time interval. Thus the number of tours starting from a given stationary object that can solve the problem is finite. There are a finite number of stationary objects, hence there exist a finite number of tours that start at stationary objects and solve the persistent visitation problem if infinitely repeated. □

3.2.4 Problem formulation

The goal of the agent in the persistent visitation problem with fuel constraints is to find the tour that minimizes the cost per unit time while satisfying the fuel and slack time constraints. Thus, the tour needs to satisfy continuity constraints for the slack times and the fuel. The time in transit between consecutive visits to a stationary object must be less than or equal to that stationary object's revisit deadline. The fuel consumed in transit between consecutive visits to fuel depots must be less than or equal to the vehicle fuel capacity. In addition, infinite repetitions of this tour must form a solution. Thus, the tour must satisfy closure properties. The time in transit between the last visit and the first visit to a stationary object must be less than that stationary object's revisit deadline and the fuel consumed between the last depot visit and the first depot visit must be less than the vehicle fuel capacity.

Let $T(k)$, $k \in \mathbb{N}$ be a tour of length L. Let O be the matrix of visitation indices, O has $n+1$ rows where the first n rows correspond to visits to stationary objects and the last row corresponds to visits to fuel depots. $O_i(k)$ is the index of the k^{th} visit to stationary object i in the tour and $O_{n+1}(k)$ is the index of the k^{th} visit to a fuel depot in the tour. Let l_i be the length of O_i, l_{n+1} be the length of O_{n+1}. For example, if $O(2)$ contains $\{2,5,7\}$ then stationary object 2 was visited at steps 2, 5, and 7, equivalently $T(2) = T(5) = T(7) = 2$. If there are four stationary objects, then $O(5) = \{3,4,6\}$ means that fuel depots were visited at steps 3, 4, and 6, equivalently $T(3) > 4$, $T(4) > 4$, and $T(6) > 4$ (since there are four stationary objects the fuel depots are represented by integers higher than 4). Let $h(i, j)$ be the distance between location i and location j.

Based on the model above, we formulate the problem as follows: the vehicle is to find a tour T such that the total fuel cost per unit time is minimized and the following conditions on continuity and closure of slack times and fuel hold:

$$\sum_{j=O_i(k)}^{O_i(k+1)-1} \frac{h(T(j),T(j+1))}{v} \leq r_i, 1 \leq i \leq n, 1 \leq k \leq l_i - 1, \tag{3.25}$$

$$\sum_{j=O_{n=1}(k)}^{O_{n+1}(k+1)-1} \frac{h(T(j),T(j+1))}{v} \leq r_f, 1 \leq k \leq l_i - 1, \tag{3.26}$$

$$\sum_{j=O_i(l_i)}^{L-1} \frac{h(T(j),T(j+1))}{v} + \sum_{j=1}^{O_i(1)-1} \frac{h(T(j),T(j+1))}{v} \leq r_i, 1 \leq i \leq n, \tag{3.27}$$

$$\sum_{O_{n+1}(l_{n+1})}^{L-1} \frac{h(T(j),T(j+1))}{v} + \sum_{j=1}^{O_{n+1}(1)-1} \frac{h(T(j),T(j+1))}{v} \leq r_f. \tag{3.28}$$

The first two equations are the continuity constraints and the last two equations are the closure constraints. The slack time continuity and closure constraints are equivalent to the intrusion time constraints used in [32].

3.2.5 Solution

A multiple step approach to finding the tour that solves the stated problem is now presented. The first step consists of Algorithm 2, which finds all tours that satisfy the slack time constraints and can satisfy the fuel constraint without solving for the amounts of fuel to be purchased; it uses the functions presented in Algorithm 3 and Algorithm 4 for constraint verification. The next step in the approach is to solve a constrained minimization problem for each tour to calculate the amounts of fuel to be purchased that satisfy the fuel constraint and minimize the total cost spent on fuel during the tour. The final step of the approach selects the tour with the minimum total cost divided by the time length, thus choosing the tour with the minimum cost per unit time.

3.2.5.1 Algorithm to find tours

This algorithm finds all tours that can solve the problem and start from stationary objects. The algorithm uses three main variables in its computation: T is the sequence of visitations of the current tour, O is the matrix of visitation indices, and Π is the set of valid tours.

The recursive algorithm used is presented in Algorithm 2. The algorithm iterates through the possible locations to be visited, appends a visit to the current tour, and checks whether the current sequence satisfies the continuity and closure conditions. Specific steps proceed as follows; lines 2-14 consist of the loop for potential next visits. Line 3 forces tours to start at stationary objects and does not allow consecutive visits to the same location. Lines 6-9 add an index to the matrix of visitation indices for the new visit. Line 8 checks the continuity conditions of the mission state, step 9 checks whether the tour has completed, and line 10 checks the closure conditions of the mission state if the tour has completed. In lines 10-12, if the continuity conditions of the mission state are satisfied, the tour has completed, and the closure conditions of the mission state for the completed tour are satisfied then the tour is added to the set of valid tours; while if the tour satisfies the continuity conditions and is incomplete, the search on that tour is continued as indicated in

Algorithm 3.2: The $findTours()$ algorithm to find all valid tours starting from stationary objects.

```
Data: $T, O, \Pi$
1  $L \leftarrow length(T)$
2  for $l \leftarrow 2$ to $m$ do
3      if $(L \neq 0 \vee i \leq n) \wedge (L = 0 \vee i \neq T(end))$ then
4          $T' \leftarrow T.add(i)$
5          $O' \leftarrow O$
6          if $i \leq n$ then
7              $O'(i) \leftarrow O'(i).add(L+1)$
8          else
9              $O'(n+1) \leftarrow O'(n+1).add(L+1)$
10         if $missionStateContinuity(T', O') \wedge T'(1) = T'(end)$
11             $\wedge missionStateClosure(T', O')$ then
12             add $T'$ to $\Pi$
13         else if $missionStateContinuity(T', O') \wedge T'(1) \neq T'(end)$ then
14             $\Pi = findTours(T', O', \Pi)$
Result: $\Pi$
```

lines 13-14; otherwise the tour is rejected.

Algorithm 2 calls two other functions to check the continuity and closure conditions of the mission state. These functions implement the restrictions in (3.25)-(3.28) and are presented in Algorithm 3 and Algorithm 4 respectively.

The mission state continuity function, Algorithm 3, first verifies the continuity conditions for the fuel and then verifies the continuity conditions for all the slack times. From one iteration in Algorithm 2 to the next, the length of the sequence can be changed by zero or one thus there is no need to verify that the continuity constraints in (3.25) and (3.26) are satisfied; instead this function verifies the last two visits to stationary objects and depots. The fuel continuity verification is achieved by checking that the time between the last two depot visits and the time between the last depot visit and the current step are less than the vehicle endurance. The slack time continuity verification is achieved by checking that the slack time for stationary object i between the last two visits to stationary object i and the time elapsed from the last visit to stationary object i and the current step are less than the

Algorithm 3.3: The *missionStateContinuity*() function to verify continuity of mission states.

```
Data: $T, O$
$L \leftarrow length(T)$, $K \leftarrow length(O(n+1))$
if $K = 0$ then
    $v_1, v_2 \leftarrow 1$
else if $K = 1$ then
    $v_1 \leftarrow 1, v_2 \leftarrow O(n+1)$
else
    $v_1, v_2 \leftarrow$ last two entries in $O(n+1)$
if $\sum_{i=v_1}^{v_2-1} \frac{locations(T(i)).distance(locations(T(i+1)))}{v} > r_f$ then
    return $false$
if $\sum_{i=v_2}^{L-1} \frac{locations(T(i)).distance(locations(T(i+1)))}{v} > r_f$ then
    return $false$
for $i \leftarrow 1$ to $n$ do
    if $length(O(i)) = 0$ then
        $v_1, v_2 \leftarrow 1$
    else if $length(O(i)) = 1$ then
        $v_1 \leftarrow 1, v_2 \leftarrow O(i)$
    else
        $v_1, v_2 \leftarrow$ last two entries in $O(i)$
    if $\sum_{i=v_1}^{v_2-1} \frac{locations(T(i)).distance(locations(T(i+1)))}{v} > r_i$ then
        return $false$
    if $\sum_{i=v_2}^{L-1} \frac{locations(T(i)).distance(locations(T(i+1)))}{v} > r_i$ then
        return $false$
return $true$
```

Algorithm 3.4: The *missionStateClosure*() function to verify closure of mission states.

```
Data: $T, O$
1  $L \leftarrow length(T)$
2  $v_1 \leftarrow$ last entry in $O(n+1)$, $v_2 \leftarrow$ first entry in $O(n+1)$
3  $t = \sum_{i=v_1}^{L-1} \frac{locations(T(i)).distance(locations(T(i+1)))}{v} + \sum_{i=1}^{v_2-1} \frac{locations(T(i)).distance(locations(T(i+1)))}{v}$
4  if $t > \frac{F}{f_c}$ then
5      return $false$
6  for $i \leftarrow 1$ to $n$ do
7      $v_1 \leftarrow$ last entry in $O(i)$, $v_2 \leftarrow$ first entry in $O(i)$
8      $t = \sum_{i=v_1}^{L-1} \frac{locations(T(i)).distance(locations(T(i+1)))}{v} + \sum_{i=1}^{v_2-1} \frac{locations(T(i)).distance(locations(T(i+1)))}{v}$
9      if $t > r_i$ then
10         return $false$
11 return $true$
```

revisit deadline for stationary object i.

Similarly, the mission state closure function, Algorithm 4, checks that the time elapsed between the last fuel depot visit and the first is less than the vehicle endurance and that the time elapsed between the last visit to stationary object i and the first visit to stationary object i is less than the revisit deadline for stationary object i.

3.2.5.2 Tour fuel cost minimization

For each tour, a constrained minimization problem is solved to calculate the fuel to be purchased at each depot such that the total cost of fuel is minimized. Let T be a tour; the sequence of fuel depot visits, D, can be extracted from T: $D = \{d_1, d_2, \cdots, d_m\}$ where d_i is the i^{th} depot visited. We define $\Delta t_i, 1 \leq i \leq m-1$ as the time spent traveling between d_i and d_{i+1}; Δt_m is the time in transit between d_m and d_1. Let Δf_i be the amount of fuel purchased by the vehicle when visiting d_i.

During the tour, the fuel the vehicle is carrying must always be bounded by zero and the vehicle fuel capacity. Thus the amount of fuel purchased at a given step must be large enough for the vehicle to reach the next depot but small enough as to not surpass the fuel capacity. This requirement results in the following constraints on the amount of fuel purchased:

$$\dot{f}_c \cdot \Delta t_1 \leq \Delta f_1 \leq F, \tag{3.29}$$

$$\dot{f}_c \cdot (\Delta t_1 + \Delta t_2) \leq \Delta f_1 + \Delta f_2 \leq F + \dot{f}_c \cdot \Delta t_1, \tag{3.30}$$

$$\dot{f}_c \cdot \sum_{i=1}^{m} \Delta t_i \leq \sum_{i=1}^{m} \Delta f_i \leq F + \dot{f}_c \cdot \sum_{i=1}^{m-1} \Delta t_i. \tag{3.31}$$

These constraints can be restated in the following matrix inequality form:

$$\begin{bmatrix} -1 & 0 & \cdots & 0 \\ -1 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} \cdot \begin{bmatrix} \Delta f_1 \\ \Delta f_2 \\ \vdots \\ \Delta f_m \end{bmatrix} \leq \begin{bmatrix} -\dot{f}_c \cdot \Delta t_1 \\ -\dot{f}_c \cdot (\Delta t_1 + \Delta t_2) \\ \vdots \\ -\dot{f}_c \cdot \sum_{i=1}^{m} \Delta t_i \\ F \\ F + \dot{f}_c \cdot \Delta t_1 \\ \vdots \\ F + \dot{f}_c \cdot \sum_{i=1}^{m-1} \Delta t_i \end{bmatrix}. \tag{3.32}$$

In addition, the amount of fuel purchased must always be greater than or equal to zero and less than or equal to the fuel capacity:

$$0 \leq \Delta f_i \leq F, 1 \leq i \leq m. \tag{3.33}$$

The goal of the vehicle is to minimize the cost per unit time. Since the duration of the tour is already, it can be ignored, thus the minimization is expressed as follows:

$$\min_{\Delta f_i, 1 \leq i \leq m} \left(\sum_{j=1}^{m} \Delta f_j \cdot c_{d_j} \right). \tag{3.34}$$

Equations (3.32), (3.33), and (3.34) form a constrained minimization problem which can be solved using linear programming methods.

3.2.5.3 Minimum cost per unit time tour

The third part of the algorithm selects the tour that results in the path with minimum cost per unit time:

$$T_{best} := \min_{T \in \Pi} \left(\frac{\sum_{j=1}^{m(T)} \Delta f_j(T) \cdot c_{d_j(T)}}{\tau(L(T))} \right). \tag{3.35}$$

The initial conditions for the path generated by infinite repetitions of a tour are as stated in (3.20) where $f_0 = f(\tau(L(T_{best})))$.

3.2.6 Algorithm correctness, completeness, and complexity

The tour finding algorithm verifies that slack time and fuel continuity are satisfied whenever a visitation is added; in addition slack time and fuel closure are verified before a tour is admitted, and thus the tour finding portion of the algorithm is correct. The tour fuel cost minimization portion of the algorithm ensures that the amount of fuel purchased at each depot is such that the amount of fuel the vehicle is carrying is always positive and less than or equal to the fuel capacity. Hence the tour fuel cost minimization portion of the algorithm is correct. Therefore, the algorithm is correct.

The tour finding algorithm searches for all possible combinations of visits within the search depth set by the revisit deadline of the starting stationary object and returns tours that satisfy the slack time and fuel constraints. As such, the algorithm is complete by exhaustion. Because of this search methodology, the algorithm's performance at worst is of the order $(n+q-1)^{n+1}$. However, the slack time and fuel constraints mean that in reality the algorithm rejects a sequence of visitations that violates a constraint before searching the maximum depth.

The algorithm can be extended to allow tours to start at depots as well as stationary objects; however a finite number of tours would no longer be guaranteed, thus additional constraints to guarantee termination might be required. In addition, if this extension were implemented the algorithm would no longer be complete since there would be no limit to the length of solutions.

3.3 Pursuing mobile objects

In the previous sections, a single agent (with or without fuel considerations) is tasked with monitoring a number of stationary objects. This problem is now extended by considering multiple mobile agents and a number of stationary agents that cooperate to pursue a mobile object along a road network.

In this problem, the goal of the mobile agents is to intercept and capture an image of a mobile object along a road network modeled as a graph. The mobile agents do not possess sensors capable of detecting the mobile object, hence they rely on stationary agents placed at certain locations in the graph. These stationary agents are capable of detecting the presence of the mobile object and communicate this information to nearby mobile agents to assist in the interception task. The restricted capability of the mobile agents is motivated by small unmanned aerial systems that possess limited on board processing resources and electro-optical or infrared sensors incapable of ascertaining the presence of an intruder [112].

The mobile agents visit the stationary agents using a revisit deadline framework similar to the one presented in Sections 3.1 and 3.2. A visualization of this scenario is shown in Figure 1.1. The method for the pursuit of the mobile object is presented, the problem is formulated, a heuristic to select actions for the mobile agents is provided, and the effectiveness of the heuristic is shown through simulation.

3.3.1 Model of agents

There are n stationary agents placed in a planar area along a road network with Cartesian coordinates (ξ_i, ζ_i), $1 \leq i \leq n$. The stationary agents and the roads the mobile object can use to travel between them are modeled as a graph $G(N, E)$ where $e_{i,j} \in E$ if there exists a road connecting stationary agents i and j. Let $d_{i,j}$ be the length of road $e_{i,j}$. The road network and stationary agents placement is assumed to remain the same over the duration of the mission.

In addition, m mobile agents are patrolling the area, each with constant velocity v, finite fuel capacity F, and fuel consumption rate $\dot{f}_c$. There is a single base that is capable of refueling the mobile agents located at (ξ_{n+1}, ζ_{n+1}). The stationary agents are assumed to be distant from one another, thus the mobile agents' limited turn radius is not taken into account and the mobile agents are modeled as point masses moving in straight lines [108]. If straight line travel cannot be assumed then curvature must be accounted for; this can be done using results from [113] where the authors describe methods to convert paths on a graph with straight line edges to paths on a graph with edges that satisfy the turning constraints of the mobile agents.

The mobile agents are equipped with a long-range communication device, which enables communication with a central authority, and a short-range communication device, which enables the mobile agents to query the status of a stationary agent directly below. The mobile agents, stationary agents, and central authority are assumed to possess synchronized clocks; this can be achieved by calibration before the mission and periodic stationary agent clock synchronization during mobile agent visits.

Once a mobile agent obtains information from a stationary agent, it is immediately relayed to the central authority. The central authority decides which nodes the mobile agents are to visit next once their respective destinations are reached. There is no benefit from allowing multiple mobile agents to visit the same stationary agents simultaneously since a detection is shared immediately and only a single mobile agent is required to capture an image; hence a stationary agent can only be visited by a single mobile agent at a time.

3.3.1.1 Revisit deadlines

Each stationary agent has a revisit deadline, $r_i > 0, 1 \leq i \leq n$, set by the mission designer, which under ideal conditions is the maximum time that can elapse between two visits to the stationary agent by the mobile agents. The revisit deadlines are used as a method to keep the defenders' knowledge of intrusions up to date. The revisit deadlines are also used to indicate the relative importance of the various stationary agents such that the mobile agents can prioritize their actions accordingly.

3.3.1.2 Initial conditions

The initial time is set to zero, the initial set of arrival times is initialized to the empty set, without loss of generality the mobile agents are assumed to start at the base with full fuel capacity, and the slack time for each stationary agent is initialized to its respective revisit deadline, i.e.,

$$
\begin{aligned}
\tau(0) &= 0, \\
\Omega(0) &= \emptyset, \\
p_1(0) &= n+1, \\
({\xi_1}'(0), {\zeta_1}'(0)) &= (\xi_{n+1}, \zeta_{n+1}), \\
p_2(0) &= n+1, \\
({\xi_2}'(0), {\zeta_2}'(0)) &= (\xi_{n+1}, \zeta_{n+1}), \\
&\vdots \\
p_m(0) &= n+1, \\
({\xi_m}'(0), {\zeta_m}'(0)) &= (\xi_{n+1}, \zeta_{n+1}), \qquad (3.36)\\
f_1(0) &= F, \\
f_2(0) &= F, \\
&\vdots \\
f_m(0) &= F, \\
x_1(0) &= r_1, \\
x_2(0) &= r_2, \\
&\vdots \\
x_n(0) &= r_n.
\end{aligned}
$$

3.3.1.3 Dynamics

Let $q_j(i,k)$ be the Euclidean distance between mobile agent j and stationary agent i at step k. When a mobile agent arrives at its destination, its next destination is set by the input and the travel time to that destination is added to the set of arrival times:

$$\text{if } q_j(p_j(k),k) = 0, \quad \begin{cases} p_j(k+1) & := u_j(k) \\ \Omega(k) & := \{\frac{q_j(u_j(k),k)}{v}\} \cup \Omega(k), \end{cases}$$
$$\text{else} \qquad p_j(k+1) \quad := p_j(k). \tag{3.37}$$

The time of the next step is set by the minimum mobile agent arrival time in $\Omega(k)$; that time is then removed from $\Omega(k)$:

$$\tau(k+1) := \min(\Omega(k)), \tag{3.38}$$

$$\Omega(k+1) := \Omega(k) \setminus \{\tau(k+1)\}. \tag{3.39}$$

The amount of fuel onboard a mobile agent decreases as dictated by the fuel consumption rate and is reset to full capacity whenever a visit to the base occurs:

$$\begin{aligned} \dot{f}_j(t) = -\dot{f}_c + \sum_{l=1}^{k-1} \dot{f}_c \cdot (1 - \delta_{p_j(l)p_j(l+1)} \cdot \delta_{(n+1)p_j(l)}) \cdot H(t-\tau(l)) \cdot H(\tau(l+1)-t) \\ + \sum_{l=1}^{k} \delta_{(n+1)p_j(l)} \cdot (F - f_j(\tau(l))) \cdot \delta(t-\tau(l)),\ t \leq \tau(k)\ ,\ 1 \leq j \leq m \end{aligned} \tag{3.40}$$

where δ_{ij} is the discrete Kronecker delta function, $H(t-\tau(l))$ is the Heaviside step function, and $\delta(t-\tau(j))$ is the continuous Dirac delta function. The slack time of a given stationary agent decreases linearly in time and is reset to its revisit deadline whenever it is visited by a mobile agent:

$$\begin{aligned} \dot{x}_i(t) = -1 + \sum_{l=1}^{k-1} \sum_{j=1}^{m} (1 - \delta_{p_j(l)p_j(l+1)} \cdot \delta_{ip_j(l)}) \cdot H(t-\tau(l)) \cdot H(\tau(l+1)-t) \\ + \sum_{l=1}^{k} \sum_{j=1}^{m} \delta_{ip_j(l)} \cdot (r_i - x_i(\tau(l))) \cdot \delta(t-\tau(l)),\ t \leq \tau(k)\ ,\ 1 \leq j \leq m \end{aligned} \tag{3.41}$$

3.3.2 Mobile object model

A single mobile object is traveling on the road network at a time; but there may be multiple mobile objects over the course of the mission. The mobile objects move inside their adversary's territory and suspect they are under observation. Thus, the mobile objects move stochastically to reduce the predictability of their actions. However, the mobile objects do not know how they are being observed and cannot perceive the mobile agents flying above [40]. If the mobile objects can detect the mobile agents in the area, then a different mobile object model is needed, e.g., using results from the pursuit-evasion literature or Stackelberg games.

The mobile object moves from one node to the next in the graph according to a first order Markov process, i.e., the next location it visits only depends on its current location. The central authority is assumed to possess the Markov model for the mobile object's movement; this model could have been obtained through intelligence or deduced from prior observation. The mobile object enters the graph at a random node according to the initial distribution of the Markov model. Since the mobile object's movement is memoryless, the mobile agents and central authority only use the most recent mobile object detection for their computations and do not keep track of the trail of stationary agents' detections left by the mobile object.

The distance traveled by the mobile object at time t is modeled as the process X_t:

$$\begin{aligned} X_0 &= 0, \\ X_t - X_\rho &= \mathcal{N}(\mu_v \cdot (t-\rho), \sigma_v^2 \cdot (t-\rho)^2), t > \rho > 0, \end{aligned} \tag{3.42}$$

where $\mu_v > 0, \sigma_v > 0$. The fuel consumption of the mobile object's vehicle is assumed to be negligible.

3.3.3 Mobile object interception

To direct the mobile agents such that interception is likely to occur, the probability of the mobile object passing by a stationary agent that a mobile agent is loitering above must be calculated.

3.3.3.1 Probability of mobile object passing a stationary agent in a given time window

Given a path s of stationary agents where $s(i)$ indicates the i^{th} stationary agent visited and the fact that the mobile object passed $s(1)$ at time 0, the probability of the mobile object passing $s(2)$ between times t_i and t_f (where $t_i > 0$ and $t_f > t_i$) is:

$$P_{int}(s(2), s, t_i, t_f) = \int_{t_i}^{t_f} P[X_\rho = d_{s(1),s(2)}]d\rho. \tag{3.43}$$

The probability of the mobile object passing $s(3)$ between times t_i and t_f is:

$$P_{int}(s(3), s, t_i, t_f) = \int_{t_i}^{t_f} P[X_\rho = d_{s(1),s(2)} + d_{s(2),s(3)}]d\rho. \tag{3.44}$$

The probability of the mobile object passing $s(b+1)$ between times t_i and t_f is:

$$P_{int}(s(b+1), s, t_i, t_f) = \int_{t_i}^{t_f} P[X_\rho = \sum_{f=1}^{b} d_{s(f),s(f+1)}]d\rho. \tag{3.45}$$

Generalizing, the probability of the mobile object passing stationary agent j (while traveling along the path s) between times t_i and t_f is:

$$P_{int}(j, s, t_i, t_f) = \sum_{b=1}^{|s|-1} \delta_{s(b+1),j} \int_{t_i}^{t_f} P[X_\rho = \sum_{f=1}^{b} d_{s(f),s(f+1)}]d\rho. \tag{3.46}$$

The probability that the mobile object passes stationary agent j between times t_i and t_f before reaching the base is:

$$P_{int}(j, s, t_i, t_f) = \sum_{b=1}^{|s|-1} (1 - \mathbf{1}_{s(1:b)}(n+1))\delta_{s(b+1),j} \int_{t_i}^{t_f} P[X_\rho = \sum_{f=1}^{b} d_{s(f),s(f+1)}]d\rho, \tag{3.47}$$

where $\mathbf{1}_{s(1:b)}(n+1)$ is the indicator function.

A set of paths S is introduced, where $S(a)$ indicates the a^{th} path within the set and $S(a, f)$ indicates the f^{th} node visited in the a^{th} path within the set. $P(S(a))$ is the probability of the a^{th} path occurring in the set where $\sum_{a=1}^{|S|} P(S(a)) = 1$; $P(S(a))$ is computed using the Markov model for the mobile object's motion along the road network. Thus the probability that the mobile object passes stationary agent j between times t_i and t_f before reaching the base given a set of paths S is:

$$P_{int}(j, S, t_i, t_f) = \\ \sum_{a=1}^{|S|} P(S(a)) \cdot \sum_{b=1}^{|S(a)|-1} (1 - \mathbf{1}_{S(a,1:b)}(n+1)) \cdot \delta_{S(a,b+1),j} \cdot \int_{t_i}^{t_f} P[X_\rho = \sum_{f=1}^{b} d_{S(a,f),S(a,f+1)}]d\rho. \tag{3.48}$$

3.3.3.2 Probability of mobile object interception

Let $g_i(j, k)$ be the Euclidean distance between the positions of mobile agent i at steps j and k. Given the probability of a mobile object passing a stationary agent during a certain time window, the probability of mobile object interception can be calculated by accounting for the mobile agent locations:

$$P_{capture}(S, y(l), y(l+1)) = \sum_{j=1}^{m} \alpha_j \cdot P_{int}(p_j(l+1), S, \tau(l), \tau(l+1)), \tag{3.49}$$

$$\text{where } \alpha_j = \begin{cases} 1 & \text{if } g_j(l, l+1)=0, \\ 0 & \text{otherwise.} \end{cases}$$

This equation consists of a sum over all the mobile agents, where j indicates the mobile agent index. α_j is only 1 when mobile agent j is loitering over a stationary agent during the step, in which case the probability of a mobile object passing the stationary agent where the mobile agent is located during the time window of the step is added.

3.3.4 Problem formulation

Using the models for the agents and mobile objects presented in the previous subsection, the problem is now formulated. Given the mobile agent states at the current step, stationary agents' revisit deadlines, and results from stationary agents' queries, the mobile agents are to find paths that satisfy the revisit deadlines and maximize the probability of intercepting the mobile object. This revisit deadline satisfaction version of the problem does not allow for missing any revisit deadline and hence limits the mobile agents' ability to pursue the mobile object.

Thus, a revisit deadline optimization version problem is formulated to give the mobile agents flexibility in meeting revisit deadlines and pursuing the mobile object. Instead of satisfying the revisit deadlines, the amount by which revisit deadlines are missed is minimized. Let $t_{mission}$ be the duration of the mobile agents' base defense mission. Let S_l be the set of possible mobile object paths given the most recent mobile object detection at step l. Let $\lfloor h \rfloor(u,t)$ indicate the smallest step in sequence u of mobile agent actions where $\tau(\lfloor h \rfloor(u,t)) \geq t$. Let $\tilde{x}_i(t)$ and $\underset{\sim}{x}_i(t)$ respectively be the slack time left and the slack time overdue for stationary agent i at time t,

$$\tilde{x}_i(t) = \frac{(|x_i(t)| + x_i(t))}{2}, \tag{3.50}$$

$$\underset{\sim}{x}_i(t) = \frac{(|x_i(t)| - x_i(t))}{2}. \tag{3.51}$$

Let β be the cost of not capturing the mobile object and let γ be the cost associated with missing a deadline, where $\beta > 0$ and $\gamma > 0$. Let $C(l)$ be the cost of the mobile agent actions taken at step l,

$$\begin{aligned} C(l) =& \beta \cdot (\tau(l+1) - \tau(l)) \cdot \Big(1 - P_{capture}(S_l, y(l), y(l+1)\Big) \\ &+ \frac{\gamma}{n} \cdot \sum_{i=1}^{n} (\tau(l+1) - \tau(l) - \tilde{x}_i(\tau(l))) \cdot \frac{\underset{\sim}{x_i}(\tau(l+1)) + \underset{\sim}{x_i}(\tau(l))}{2}. \end{aligned} \tag{3.52}$$

The first component of the cost penalizes the mobile agents for not intercepting the mobile object over the course of the step. The second component of the cost penalizes revisit deadlines that are overdue by adding the integral of the slack time missed over the course of the step.

Based on the cost function above, the revisit deadline optimization version of the problem is formulated as follows: the central authority is to find a sequence $u_j(k), 1 \leq j \leq m, k \in \mathbb{N}$ such that under (3.36)- (3.41), $\sum_{k=1}^{\lfloor h \rfloor (u, t_{mission})} C(k)$ is minimized and $0 \leq t \leq t_{mission}, 1 \leq j \leq m, f_j(t) \geq 0$.

3.3.4.1 Problem complexity

Proposition 3.3.1. *The revisit deadline satisfaction version problem of cooperative surveillance and pursuit is NP-hard.*

Proof. If no mobile objects are present in the road network then an optimal solution is one where the slack times are kept positive. The problem of keeping slack times positive for a single mobile agent is the persistent visitation problem and is proven to be NP-complete in Section 3.1. The persistent visitation problem can be reduced to the problem of cooperative surveillance and pursuit by selecting one mobile agent to patrol the same graph with the same revisit deadlines without mobile objects present. Thus the problem of cooperative surveillance and pursuit is NP-hard. □

Corollary 3.3.2. *The revisit deadline optimization version problem of cooperative surveil-*

Algorithm 3.5: System for the cooperative surveillance and pursuit problem.

```
Data: $G(N,E)$,$r$,$d$,$m$,$v$,$F$,$\dot{f}_c$,$t_{mission}$,$t_{search}$,$t_{stale}$
1  $k \leftarrow 0; t \leftarrow 0$
2  $(y(k), f(t), x(t)) \leftarrow$ (Eq. 3.36)
3  $t_D \leftarrow \emptyset; n_D \leftarrow \emptyset; T_U \leftarrow \emptyset; N_U \leftarrow \emptyset$
4  while $\tau(k) < t_{mission}$ do
5      $(S_k, P(S_k)) \leftarrow paths(n_D, 1, t_D, t_D, T_U, N_U, \tau(k) + t_{search}, \emptyset, \emptyset)$
6      $u(k) \leftarrow mobileAgentsAction(y(k), f(\cdot), x(\cdot), t_{search}, S_k, P(S_k))$
7      $(y(k+1), f(\cdot), x(\cdot)) \leftarrow$ (Eq. 3.37 – 3.41), $y(k), u(k)$
8      $k \leftarrow k+1; t \leftarrow \tau(k+1)$
9      for $(detection) \in queries(k)$ do
10         if $(detection).t > t_D$ then
11             $(t_D, n_D) \leftarrow (detection)$
12     if $t_D + t_{stale} < \tau(k)$ then
13         $t_D \leftarrow \emptyset; n_D \leftarrow \emptyset; T_U \leftarrow \emptyset; N_U \leftarrow \emptyset$
14         for $(t_U, n_U) \in (T_U, N_U)$ do
15             if $t_U \leq (\tau(k) - t_{stale})$ then
16                 $(T_U, N_U) \leftarrow (T_U, N_U) \setminus (t_U, n_U)$
17     for $(\neg detection) \in queries(k)$ do
18         $(T_U, N_U) \leftarrow (T_U, N_U) \cup (\neg detection)$
Result: $u_j(\cdot)$
```

lance and pursuit is NP-hard.

3.3.5 Mobile agent path selection

The problem is NP-hard, hence a heuristic algorithm is used to select the paths of the mobile agents. This algorithm searches ahead for a sequence of mobile agent actions that minimizes the cost function within a certain time window and executes the first set of mobile agent actions in the sequence.

3.3.5.1 System structure

The algorithm used to simulate the defenders' system for a cooperative surveillance and pursuit problem is provided in Algorithm 5. The states are first initialized (lines 1-3); then at each step the possible mobile object paths within t_{search} are computed (line 5),

Algorithm 3.6: The *paths*() algorithm to find possible mobile object paths.

```
Data: $s, p, t_{min}, t_{max}, T_U, N_U, t_f, S, P(S)$
1  $\Gamma \leftarrow adjacentNodes(s(|s|), G)); l \leftarrow |\Gamma|$
2  while $l > 0$ do
3      $T'_{max}(l) \leftarrow t_{max} + \frac{d_{s(|s|),\Gamma(l)}}{min(v_{int})}$
4      for $(t_U, n_U) \in (T_U, N_U)$ do
5          if $\Gamma(l) = n_U$ then
6              if $T'_{max}(l) \leq t_U$ then
7                  $\Gamma \leftarrow \Gamma \setminus \{\Gamma(l)\}$
8                  $T'_{max} \leftarrow T'_{max} \setminus \{T'_{max}(l)\}$
9              break
10     $l \leftarrow (l - 1)$
11 for $l \leftarrow 1$ to $|\Gamma|$ do
12     $s' \leftarrow [s\ \Gamma(l)]; p' \leftarrow p \cdot \frac{1}{|\Gamma|}$
13     $t'_{min} \leftarrow t_{min} + \frac{d_{s(|s|),\Gamma(l)}}{max(v_{int})}$
14     if $t'_{min} \geq t_f$ then
15         $S \leftarrow S \cup \{s'\}; P(S = s') \leftarrow p'$
16     else
17         $(S, P(S)) \leftarrow paths(s', p', t'_{min}, T'_{max}(l), T_U, N_U, t_f, S, P(S))$
Result: $S, P(S)$
```

followed by the computation of the minimal cost action for the mobile agents within the same time horizon using the possible mobile object paths (line 6). In addition, at each step k, new detections are added from the stationary agent queries (lines 9-11), stale queries are rejected (lines 12-16), and stationary agent queries without detections are added (lines 17-18). A stationary agent query is characterized by the status of the stationary agent, the time of the detection if one occurred (otherwise the time of the query), and the index of the queried stationary agent. The resulting data from querying the stationary agents is used in the mobile object path generation process in the next time step. This process of finding mobile object paths, selecting mobile agent actions, and gathering stationary agents queries is repeated until the mission completion time is reached (line 4).

3.3.5.2 Mobile object path generation

Possible mobile object paths are generated using the information from stationary agents queries. The central authority stores the time and stationary agent index of the most recent mobile object detection. It also stores the times and stationary agent indices of recent stationary agents queries without detections. This information is used to generate the potential mobile object paths at each step using a recursive breadth first search methodology. If a detection is too stale then it is ignored. In simulations, the threshold for a detection to be considered stale was selected to be half the mean mobile object travel time between the two stationary agents most distant from one another.

A recursive algorithm is used to compute the possible mobile object paths (Algorithm 6); this algorithm is used by the heuristic to assist in its decision making process. The input of the algorithm is the current candidate path for the mobile object (which at the start of the algorithm's execution is the most recent detection). At each iteration in the search, the nodes adjacent to the last node of the current working path, s, are obtained (line 1). Possible travel times to these adjacent nodes are then computed using the mobile object's velocity probability distribution. If any of the visits to the adjacent nodes violates the constraints set by recent stationary agent queries without detections then they are removed (lines 2-10). Valid adjacent nodes are then appended to the current working path (lines 11-12). If the minimum travel time of the new path is larger than the search depth then the path is admitted to the set of possible paths (lines 14-15), otherwise the search continues for that path (lines 16-17). When all the candidate paths reach the search depth, the algorithm terminates and returns the possible mobile object paths. These possible mobile object paths are then used to select the actions of the mobile agents.

3.3.5.3 Selection of mobile agents' actions

The following algorithm is used by the heuristic to make its decisions. The algorithm starts by searching for all possible sequences of actions for the team of UAVs within the search horizon t_{search}. For each sequence of actions available to the team of mobile agents, $\tilde{u}$, it then assesses the cost using the following equation:

$$\sum_{l=k}^{\lfloor h \rfloor(\tilde{u},t_{search})} C(l), \tag{3.53}$$

where k is the current step. The cost calculations use the mobile object paths generated earlier.

The algorithm starts by using the current state of the mobile agents to compute feasible actions; the action is then applied and the corresponding mobile agent states and costs are computed. These actions, states, and costs are then added to sets of candidate sequences of actions, candidate states of the mobile agents after the application of the corresponding sequence of actions, and candidate costs after the application of the corresponding sequence of actions. The algorithm then proceeds to iterate over the set of candidate sequences of actions for the mobile agents.

The procedure for computing potential sequences of actions and their costs is described in Algorithm 7. Several intermediate variables are used: $\tilde{U}$ is the working set of sequences of mobile agent actions, W is the corresponding set of states after the sequences of actions in $\tilde{U}$ have been applied, and Θ is the set of costs for the sequences of actions. u is the current minimal cost sequence of actions and ϕ is the current minimal cost. These variables are initialized (lines 1-2); the algorithm then proceeds to loop over the working set of sequences of mobile agent actions until none remain (line 3). At each iteration, the first elements in the working sets of states, sequences of actions, and costs are obtained and removed from their parent sets (lines 4-5). The set of possible actions, Z, given the current working state $w \in W$ is then computed (lines 6-14). For each possible action for the team of mobile agents, $z \in Z$, the next state, λ, is computed (line 15). If λ results in positive fuel for all the mobile agents and has reached the search depth then its cost is computed (lines 16-17); if that cost is smaller than the current minimal cost then the current minimal cost sequence of actions is set to the sequence of actions that resulted in λ (lines 18-19). If λ results in positive fuel for all the mobile agents without reaching the search depth then the working state, sequence of actions, and cost are added to the corresponding parent working sets (lines 20-24). The algorithm terminates when the set of candidate sequences of mobile agent actions is empty (i.e., all feasible actions in the search horizon have been considered) and returns the minimal cost sequence of actions.

The heuristic directs the mobile agents to take the first step in the minimal cost sequence of actions; this step results in the mobile agents visiting or loitering above certain stationary agents, thereby obtaining new information from the stationary agents and possibly intercepting the mobile object. The possibility of interception exists when a mobile agent is loitering a stationary agent; interception occurs when the stationary agent which the mobile agent is loitering detects the mobile object.

3.3.5.4 Algorithm completeness

The search depth for the mobile agent actions affects the existence of solutions; if the search depth is less than the endurance of the mobile agents, $t_{search} < (F/\dot{f}_c)$, then there is no guarantee that the generated paths lead to the satisfaction of the mobile agents' fuel constraints. If $t_{search} \geq t_{mission}$ then the heuristic is complete, i.e., the heuristic finds a solution if one exists.

3.3.5.5 Algorithm complexity

The topology of the road network, the number of mobile agents, the mobile agents' velocity, the mean mobile object velocity, and t_{search} affect the algorithm complexity. Let $\bar{d}$ be the mean distance between any two stationary agents. By inspection, the time complexity of the heuristic per step is

$$O\left(|E|^{\frac{t_{search}\cdot\mu_v}{\bar{d}}} + \left(\frac{n!}{m!(n-m)!}\right)^{\frac{t_{search}\cdot v}{\bar{d}}}\right). \tag{3.54}$$

Algorithm 3.7: The *mobileAgentsAction*() heuristic to find the minimal cost action for the mobile agents within a given search horizon.

```
Data: $y(k)$,$f(\cdot)$,$x(\cdot)$,$S$,$P(S)$
$W \leftarrow \{(y(k), f(\cdot), x(\cdot))\}$
$\tilde{U} \leftarrow \emptyset; \Theta \leftarrow \emptyset; u \leftarrow \emptyset; \phi \leftarrow \infty$
while $|W| > 0$ do
    $w \leftarrow W(1); \tilde{u} \leftarrow \tilde{U}(1); \theta \leftarrow \Theta(1)$
    $W \leftarrow W \setminus \{w\}; \tilde{U} \leftarrow \tilde{U} \setminus \{\tilde{u}\}; \Theta \leftarrow \Theta \setminus \{\theta\}$
    $Z \leftarrow \emptyset$
    if $q_1(w.p_1) = 0$ then
        $Z \leftarrow N \cup \{n+1\}$
    for $l \leftarrow 2$ to $m$ do
        if $q_l(w.p_l) = 0$ then
            $Z \leftarrow Z \times \{N \cup \{n+1\}\}$
        else
            $Z \leftarrow Z \times \emptyset$
    for $z \in Z$ do
        $\lambda \leftarrow$ (Eq. 3.37 – 3.41), $w, z$
        if $(\forall j, \lambda.f_j(\lambda.\tau) \geq 0) \wedge (\lambda.\tau \geq \tau(k) + t_{search})$ then
            $\theta' \leftarrow \theta +$ (Eq. 3.52), $S, P(S), w, \lambda$
            if $\theta' < \phi$ then
                $u \leftarrow [\tilde{u}\ z]; \phi \leftarrow \theta'$
        else if $\forall j, \lambda.f_j(\kappa.\tau) \geq 0$ then
            $\theta' \leftarrow \theta +$ (Eq. 3.52), $S, P(S), w, \lambda$
            if $\theta' < \phi$ then
                $W \leftarrow W \cup \{\lambda\}; \tilde{U} \leftarrow \tilde{U} \cup [\tilde{u}\ z]$
                $\Theta \leftarrow \Theta \cup \{\theta'\}$
Result: $u$
```

The complexity increases polynomially with respect to the number of stationary agents and number of roads, increases exponentially with respect to the vehicle velocities and the search depth, and decreases exponentially with respect to the mean distance between any two stationary agents.

3.3.6 Local search heuristic

For comparison, a local search heuristic is also implemented to solve this problem (Algorithm 8). At each step, the algorithm randomly generates a feasible sequence of actions of a given depth (line 3) and computes possible intruder paths that can occur in the same time window (line 4). The k-neighborhood of the initial sequence of actions is then computed (line 6) for k = 2, and the possible intruder paths are extended due to the increased time window (line 7). For each sequence of actions, the corresponding cost is computed using the intruder paths (line 8), and the minimum cost sequence of actions is extracted (line 9). This procedure of generating a k-neighborhood and selecting the optimal sequence is repeated until convergence of the minimal cost sequence of actions or a number of iterations, $search_{iter}$, is reached (lines 10–21). The overall procedure can also be repeated for multiple initial random feasible sequences of actions to further increase the search space. $search_{init}$ indicates the number of these initial guesses. For the search to continue until convergence, $search_{iter}$ is indicated as zero. The simulations shown in this paper use $search_{init} = 5$ and $search_{iter} = 0$.

3.3.7 Simulations

To illustrate the performance of the *mobileAgentsAction*() heuristic, several simulations with varying configurations are shown. The local search heuristic, described in Section 3.3.6, is used as a baseline comparison to the *mobileAgentsAction*() heuristic developed in Section 3.3.5.

Four scenarios are considered: scenario A2 occurs in area A with a mobile agent to

Algorithm 3.8: The *localSearch*() heuristic to find a local minimal cost action for the UAVs.

```
Data: $y(k)$,$f(\cdot)$,$x(\cdot)$,$S$,$P(S)$, $search_{init}$, $search_{iter}$
1  $\tilde{U} \leftarrow \emptyset; u \leftarrow \emptyset; \phi \leftarrow \infty$
2  for $l \leftarrow 1$ to $search_{init}$ do
3      $\kappa \leftarrow$ Generate a feasible sequence of actions of a given depth
4      $S, P(S) \leftarrow$ Compute possible intruder paths for the duration of the sequence of
       actions
5      $i \leftarrow -1$
6      $\tilde{U} \leftarrow$ Compute the k-neighborhood of the sequence of actions
7      $S, P(S) \leftarrow$ Extend possible intruder paths to the maximum end time of the
       k-neighborhood
8      $\tilde{U}.cost \leftarrow$ Compute costs for all actions
9      $\kappa_2 \leftarrow \min_{\tilde{u} \in \tilde{U}}(\tilde{u}.cost)$
10     while $\kappa_2 \neq \kappa \wedge i < search_{iter}$ do
11         $\kappa \leftarrow \kappa_2$
12         $\tilde{U} \leftarrow$ Compute the k-neighborhood of the sequence of actions
13         $S, P(S) \leftarrow$ Extend possible intruder paths to the maximum end time of the
           k-neighborhood
14         $\tilde{U}.cost \leftarrow$ Compute costs for all actions
15         $\kappa_2 \leftarrow \min_{\tilde{u} \in \tilde{U}}(\tilde{u}.cost)$
16         if $search_{iter} > 0 \wedge i < 0$ then
17             $i \leftarrow i + 2$
18         else if $search_{iter} > 0$ then
19             $i \leftarrow i + 1$
20     if $\kappa_2.cost < \phi$ then
21         $u \leftarrow \kappa_2$
Result: $u$
```

mobile object velocity ratio of 2, scenario A3 occurs in area A with a mobile agent to mobile object velocity ratio of 3, scenario B1 occurs in area B with a mobile agent to mobile object velocity ratio of 1, and scenario B2 occurs in area B with a mobile agent to mobile object velocity ratio of 2. Two mobile agents are operating in area A while three mobile agents are operating in area B; mobile agent fuel consumption is not accounted for in these simulations. Visualizations of areas A and B are shown in Figure 3.5 and Figure 3.6. Detailed parameters for the scenarios are given in Tables 3.2 and 3.3; these tables contain the locations of the base and stationary agents, revisit deadlines for the stationary agents, and velocity for the mobile agents and mobile objects.

The metric used to assess the performance of the approaches is the capture index. The capture index is the number of mobile objects whose image was captured subtracted by the number of mobile objects that reached the base divided by the total number of mobile objects over the course of the mission. Three cost configurations, indicated by (β, γ), are considered: $(1, t_{mission})$, $(t_{mission}, 1)$, $(t_{mission}^2, 1)$ where the mission time, $t_{mission}$, is 30 in the simulations. Missing revisit deadlines is weighted heavily in the first cost configuration while not capturing an image of the mobile object is weighted heavily in the last cost configuration. Eighty simulations were run per search depth per scenario per cost configuration; Figures 3.7-3.10 show the average intruder capture index (represented on the ordinate) for these 80 simulations for the three different cost configurations (represented by the three different line styles) as a function of search depth for Scenarios A and B (represented on the abscissa).

In the left subfigures of Figures 3.7-3.10, where *mobileAgentsAction*() is used, the capture index increases as a function of search depth for all cost functions. The capture index for *localSearch*() (right subfigures of Figures 3.7-3.10) does not increase significantly with larger search depth. While *mobileAgentsAction*() and *localSearch*() perform similarly for short search depths, *mobileAgentsAction*() performs better than *localSearch*() for larger search depths.

Table 3.2: Base and stationary agents parameters for scenarios A and B.

	Scenarios A			Scenarios B		
	ξ_i	ζ_i	r_i	ξ_i	ζ_i	r_i
Base	6	6		6	6	
Stationary agent 1	94.66	70.33	11.4	150.03	56.21	8.1
Stationary agent 2	117.05	109.94	12.4	162.18	124.23	10.7
Stationary agent 3	57.17	151.44	10.5	37.41	72.06	3.0
Stationary agent 4	70.00	100.00	2.6	117.54	131.08	8.4
Stationary agent 5	10.79	106.16	10.8	19.65	110.68	9.5
Stationary agent 6	186.80	25.98	8.3	44.29	4.66	7.1
Stationary agent 7	93.88	2.38	5.6	148.70	107.66	12.6
Stationary agent 8	40.00	50.00	7.0			
Stationary agent 9	140.00	42.00	11.0			

For scenarios A2, A3, and B2 (Figures 3.7, 3.8, and 3.10), weighting mobile object capture heavily when using *mobileAgentsAction*() resulted in more captures for large search depths while only a marginal difference between cost configurations is seen for small search depths. In scenario B1 (Figure 3.9), weighting missed revisit deadlines more heavily for *mobileAgentsAction*() resulted in more captures for search depths less than 0.9% of the mission completion time. The performance of *localSearch*() is not significantly affected by different cost configurations.

In scenario A, an optimal search depth is seen for both velocity advantages at 1.05% of the mission completion time for *mobileAgentsAction*(). An optimal search depth is not seen in scenario B, however it may occur at a search depth greater than performed in the simulations. As expected, the mobile object capture indices in scenario A3 are larger than in scenario A2, i.e., the mobile agents perform better when they are faster. This trait is also seen when comparing scenario B2 to scenario B1. The mobile agents do not perform well when they do not have a velocity advantage (Figure 3.9) even when they are more numerous. When the mobile agents are faster than the mobile object and the topography is advantageous, as is the case in scenario B, they can perform very well (Figure 3.10).

The performance of *mobileAgentsAction*() largely depends on the problem instance as

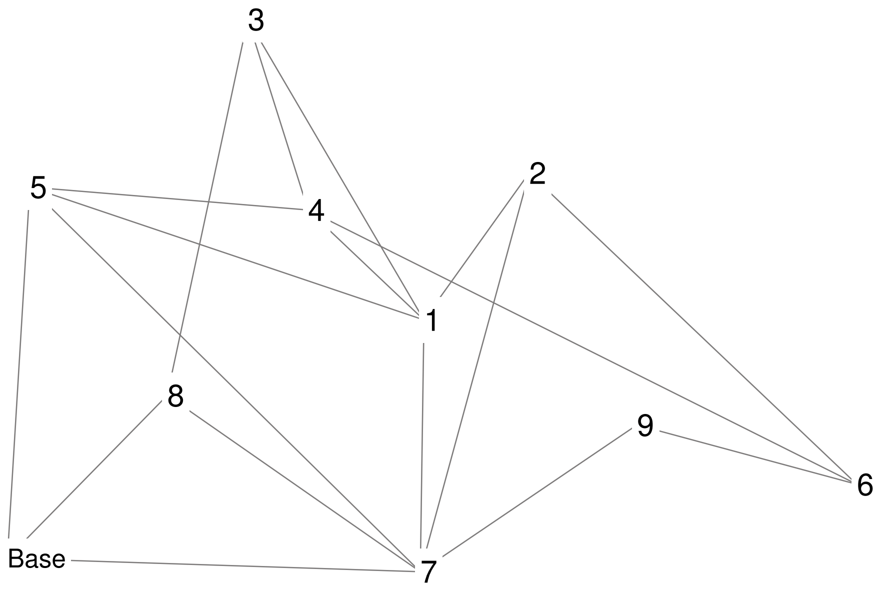


Figure 3.5: Visualization of base, stationary agents, and roads for scenario A.

Table 3.3: Vehicle parameters for scenarios A and B.

	Scenario A2		Scenario A3		Scenario B1		Scenario B2	
	μ_v	σ_v	μ_v	σ_v	μ_v	σ_v	μ_v	σ_v
Mobile object	37.5	1.57	25	1.57	25	2.33	25	2.33
Mobile agents	75	0	75	0	25	0	50	0

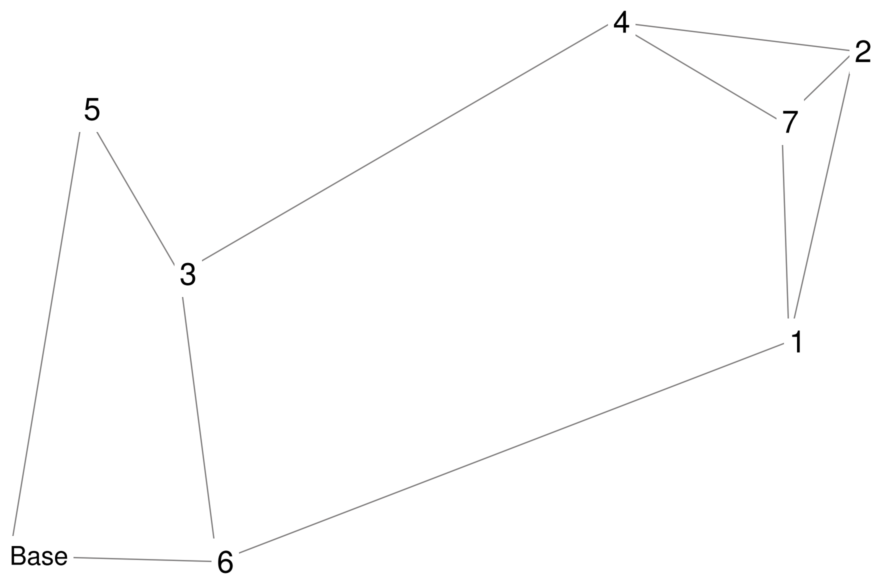
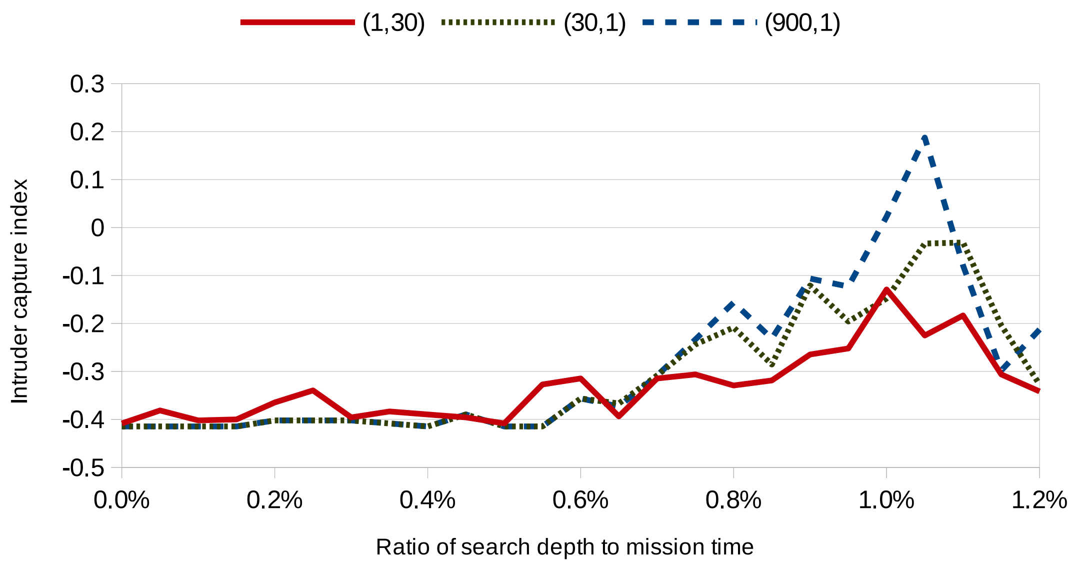


Figure 3.6: Visualization of base, stationary agents, and roads for scenario B.



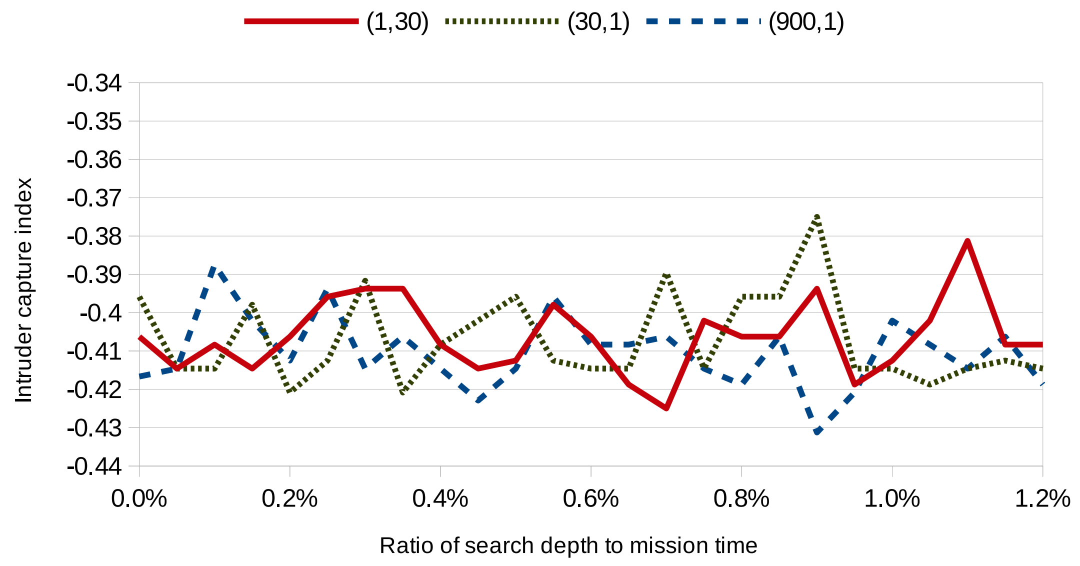
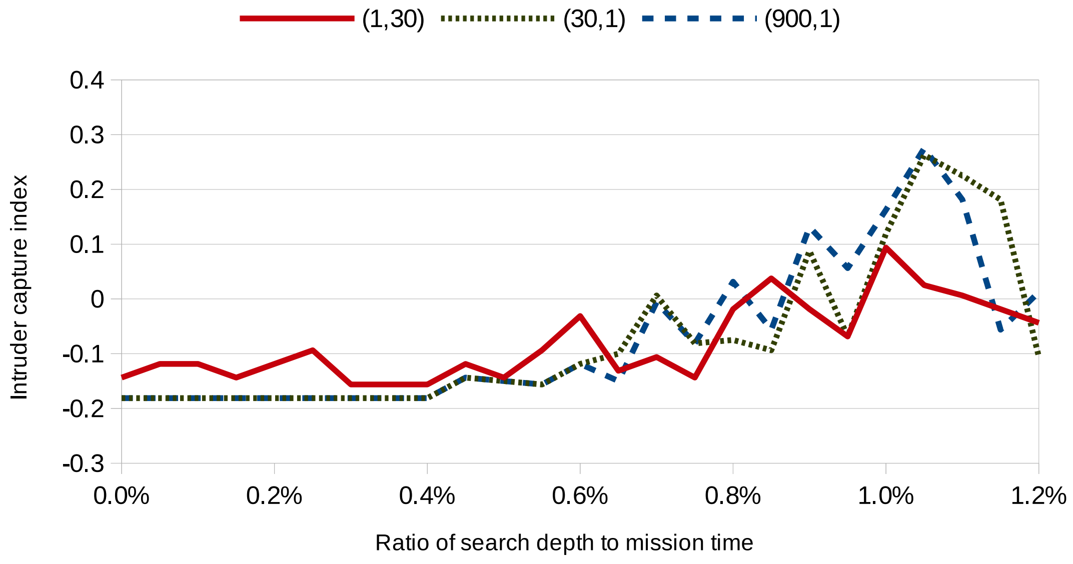


Figure 3.7: Capture index for scenario A2 using *mobileAgentsAction*() (top) and *localSearch*() (bottom).



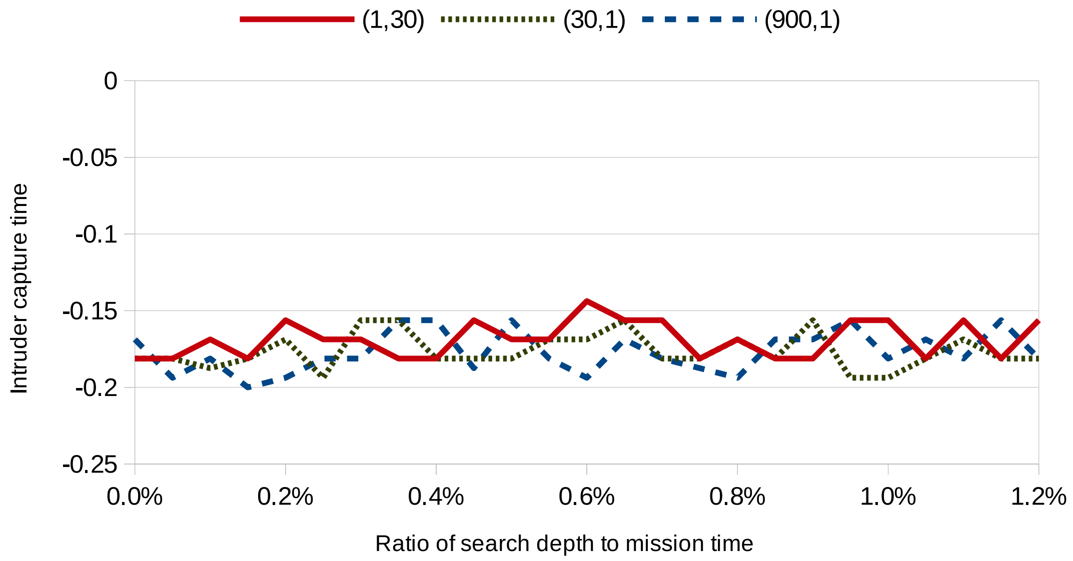
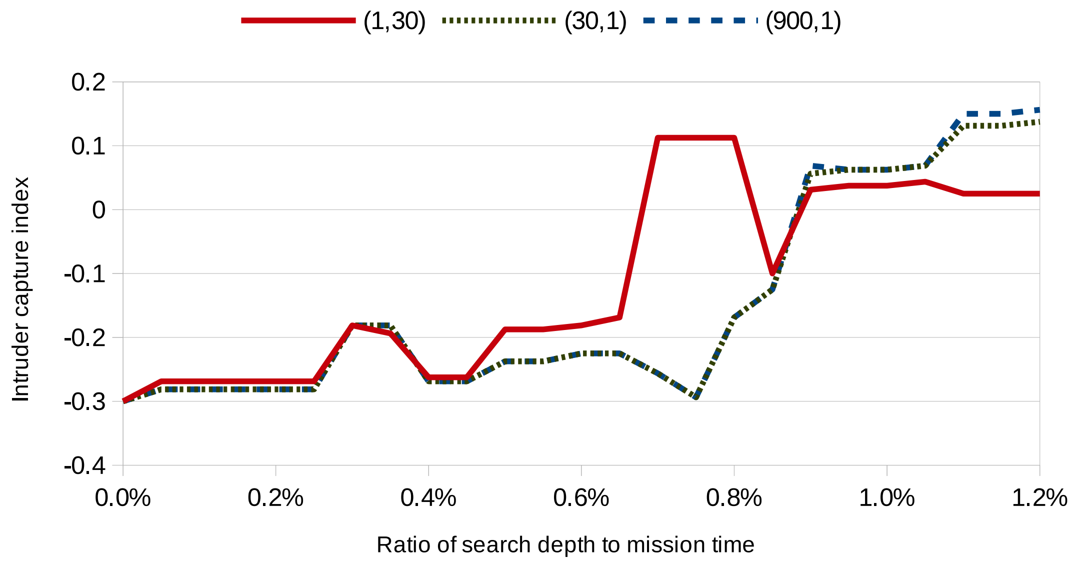


Figure 3.8: Capture index for scenario A3 using *mobileAgentsAction*() (top) and *localSearch*() (bottom).



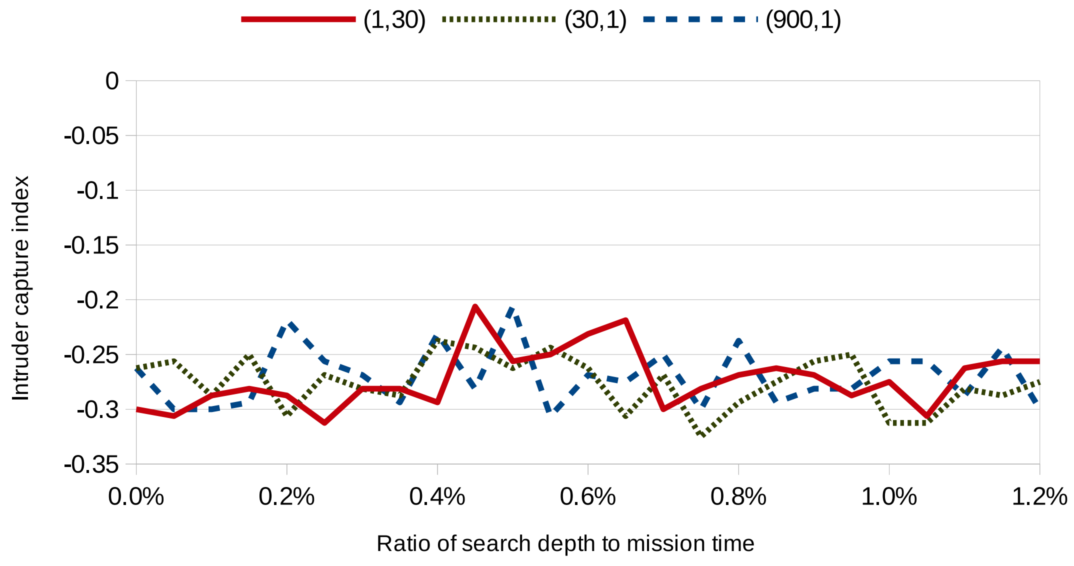
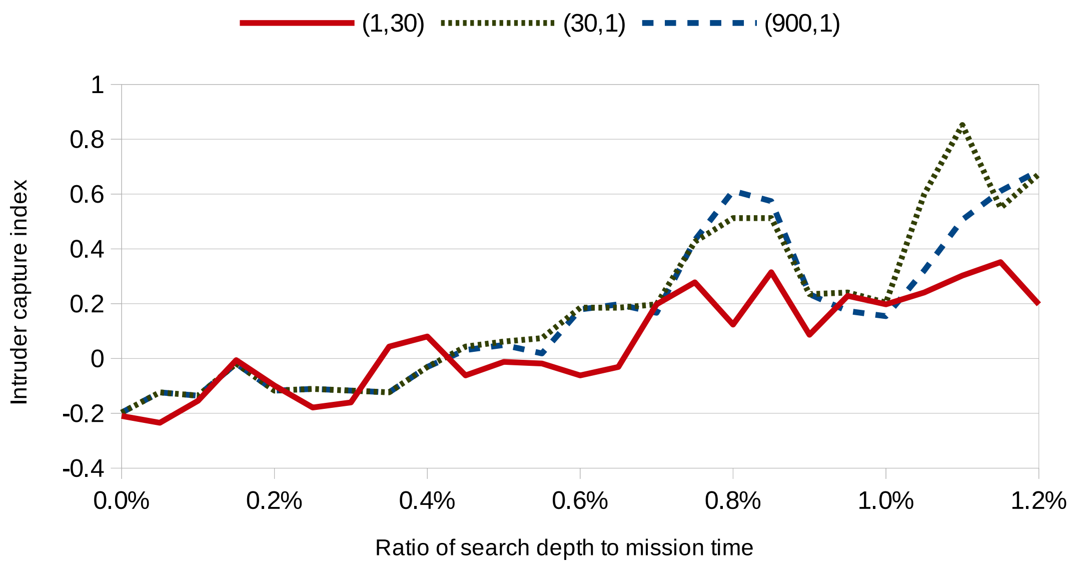


Figure 3.9: Capture index for scenario B1 using *mobileAgentsAction*() (top) and *localSearch*() (bottom).



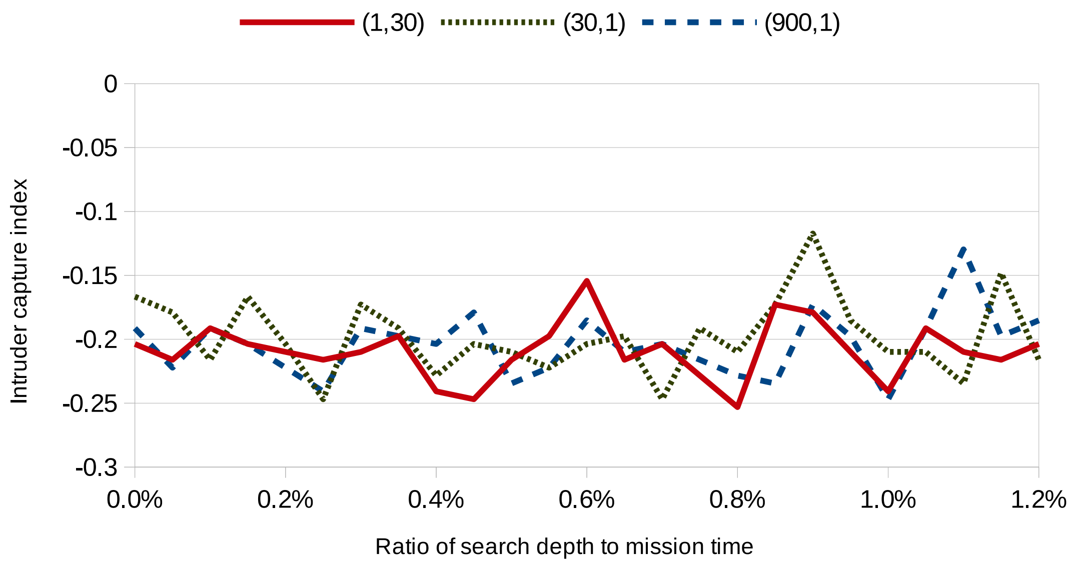


Figure 3.10: Capture index for scenario B2 using *mobileAgentsAction*() (top) and *localSearch*() (bottom).

can be seen from the differences in results from the scenarios presented. Problem instances with bottlenecks in the road network topology lead to better performance since the mobile agents' pursuit and interception of the mobile object is simplified. Topologies with highly connected nodes close together lead to poorer performance since the mobile agents cannot adequately pursue the mobile object. The effect of the topology on the performance of the heuristic highlights the importance of the selection of stationary agents' locations.

3.4 Summary

In this chapter, path planning for mobile agents monitoring stationary objects or pursuing mobile agents is studied. Using a problem formulation with revisit deadlines, almost periodic paths for a single vehicle monitoring stationary objects are demonstrated to exist. The problems are shown to be intractable and heuristics as well as algorithms are provided to compute the paths for the agents. The complexity and completeness of the heuristics and algorithms are discussed; the effectiveness of the heuristics is shown using numerical results.

CHAPTER 4

Configurations for stationary agents

In the previous chapter, path planning for mobile agents was studied and in Section 3.3 the mobile agents interacted with stationary agents. In this chapter, optimal configurations for stationary agents performing monitoring tasks are investigated. The results from this chapter are used to address the sensor placement question stated in Section 1.2: *how can mobile and stationary agents cooperate to achieve monitoring tasks?*

In this chapter, the problem of monitoring a single mobile object on an arbitrary graph using stationary agents is studied. The stationary agents detect the presence of the object by measuring a certain property of the object. The stationary agents are to be configured for optimal monitoring, i.e., their locations and revisit deadlines (the maximum time that can elapse between consecutive visits by a mobile agent) are selected to optimize monitoring.

The models used for the mobile object and the stationary agents are presented. The problem is then explicitly formulated and a variety of methods for the placement of stationary agents observing the object are given. A method to select revisit deadlines for the stationary agents is also provided. Properties of the approaches are shown and their characteristics illustrated through numerical examples.

4.1 Model

Let $G = (N, E, L)$ be a weighted digraph representing the area in which the mobile object can move, where N is the set of nodes and $n = |N|$, E is the set of directed edges in the graph, and L is the set of weights associated with the set E (e.g., the lengths of the edges). Location in this chapter means an edge or a node. The models used for the agents and object are now introduced.

4.1.1 Revisit deadlines

The stationary agents must be visited repeatedly for the mobile agents to maintain recent knowledge of the detections in the graph. Instead of using revisit periods for each stationary agent a revisit deadline is used, where a revisit deadline is the maximum time that can elapse between consecutive visits to a stationary agent. Revisit deadlines lead to increased flexibility in the mission planning process due to the larger number of paths available. However, that increased flexibility can come at the cost of additional required computation time. Methods to compute paths for mobile agents to meet revisit deadlines are given in Chapter 3.

4.1.2 Mobile object motion in graph

Let p be the probability of the mobile object's motion in the graph, where $p_{i,j} = p(E_{i,j})$ is the likelihood of the mobile object traveling along the edge from node i to node j and $\sum_{j=1}^{n} p_{i,j} = 1$. If there is no directed edge from node i to node j then $p_{i,j} = 0$. The probabilities can be computed from existing knowledge about the mobile object or derived from a red team exercise simulating the mobile object's actions towards a given objective, e.g., using game theoretic approaches.

The probability distribution of the mobile object's location among the nodes can be derived from the model of the mobile object's movements in the network. Because the mobile object's movements are modeled as a finite order ergodic Markov process, the probability distribution of the mobile object's location is obtained by computing the stationary distribution of the Markov process. Let $\mathcal{P}$ be the matrix containing the probabilities of the movements of the mobile object along the graph,

$$\mathcal{P} = \begin{bmatrix} 0 & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & 0 & \cdots & p_{2,n} \\ \vdots & & \ddots & \vdots \\ p_{n,1} & p_{n,2} & \cdots & 0 \end{bmatrix}, \tag{4.1}$$

then,

$$\pi \cdot \mathcal{P} = \pi, \tag{4.2}$$

where π contains the stationary probabilities for the individual nodes in the graph.

A stationary agent is capable of detecting a mobile object (e.g., a security camera searching for a specific object) and registers the time of the detection. A wide variety of measurements can be used by stationary agents to perform their detections. In addition, the stationary agent's sensors can be perfect or noisy in which case some detections may be false or the stationary agent may miss detections. In this work, each stationary agent possesses a sensor and a classifier; each stationary agent measures a continuous variable and makes a classification decision based on the measurement value. The sensors carried by the stationary agents may be identical or different, in which case their noise characteristics may be dependent or independent.

The stationary agents measure a distinguishing property (e.g., weight) of the mobile object (e.g., ground vehicle) to perform their classification decision. Let w be the value of the property of the mobile object, $q_j \sim \mathcal{N}(\mu_j, \sigma_j^2)$ be the noise from stationary agent j's measurement, and z indicate the stationary agent's measurement:

$$z_j = w + q_j. \tag{4.3}$$

The sensors are assumed to be calibrated hence $\mu_j = 0, \forall j$, thus the probability density function for the noise of stationary agent *j*'s sensor is:

$$f_{q_j}(q_j) = \frac{1}{\sigma_j\sqrt{2\pi}} e^{\frac{-q_j^2}{2\sigma_j^2}}, 1 \leq j \leq g. \tag{4.4}$$

When stationary agents are colocated, their measurements can be fused to generate a better estimate of the mobile object's property. This estimation is done differently depending on what is known about the mobile object's property, i.e., either the exact value of the parameter is known or its statistical properties are known.

4.1.2.1 Colocated stationary agents with known mobile object parameter

The mobile object's parameter w is assumed to be known, i.e., $w = \bar{w}$; and without loss of generality it is assumed that $\bar{w} > 0$. When multiple stationary agents (each with a sensor reading z_j) are placed at the same location and the mobile object parameter is known, the maximum likelihood estimate of the sensors, $\bar{z}$, is used to generate a single measurement:

$$\bar{z} = \underset{x}{\operatorname{argmax}}(p(\vec{z}|x)) = \underset{x}{\operatorname{argmax}}\left(f_{\vec{q}}(\vec{z} - \vec{x})\right), \tag{4.5}$$

where $\vec{z}$ contains the measurements for the individual colocated sensors, x is a candidate estimate, $\vec{x}$ is a vector of the same size as $\vec{z}$ containing the value of the candidate estimate at each entry, R is the covariance matrix for the multiple colocated sensors, and

$$f_{\vec{q}} = \frac{1}{\sqrt{2\pi}\det(R)} e^{-\frac{1}{2}\vec{q}^T R^{-1} \vec{q}} \tag{4.6}$$

is the probability density function for the combined noise of the sensors. The maximum likelihood estimate is

$$\begin{aligned}\bar{z} &= \underset{x}{\text{argmax}}\left(\frac{1}{\sqrt{2\pi}\det(R)}e^{-\frac{1}{2}(\vec{z}-\vec{x})^T R^{-1}(\vec{z}-\vec{x})}\right)\\ &= \underset{x}{\text{argmin}}\left(\frac{1}{2}(\vec{z}-\vec{x})^T R^{-1}(\vec{z}-\vec{x})\right)\\ &= \underset{x}{\text{argmin}}\left(\frac{1}{2}h(x)\right). \qquad (4.7)\end{aligned}$$

where $h(x) = (\vec{z}-\vec{x})^T R^{-1}(\vec{z}-\vec{x})$. For x to satisfy (4.7):

$$\frac{\partial h(x)}{\partial x} = 0 \qquad (4.8)$$

and

$$\frac{\partial^2 h(x)}{\partial x^2} = 2\cdot \mathbf{1}^T R^{-1}\mathbf{1} > 0, \qquad (4.9)$$

where **1** is a column vector of ones of the appropriate size.

Equation (4.8) results in an expression with $\bar{z}$ linear in the elements of $\vec{z}$ hence the maximum likelihood estimate is a linear function of $\vec{z}$ and can be written as

$$\bar{z} = \Lambda\cdot\vec{z}. \qquad (4.10)$$

where $\Lambda \in \mathbb{R}^{1\times|\vec{z}|}$ contains the coefficients of the linear relationship between $\bar{z}$ and $\vec{z}$ found in (4.8).

$\vec{z}$ is jointly Gaussian thus $\bar{z}$ is also Gaussian:

$$\bar{z} = \mathcal{N}(0,\sigma^2_{\bar{z}}), \qquad (4.11)$$

where

$$\sigma_{\bar{z}}^2 = \Lambda R \Lambda^T. \tag{4.12}$$

The variance of the maximum likelihood estimate of the sensors of the colocated stationary agents, $\sigma_{\bar{z}}^2$, can be used to model the colocated stationary agents as a single combined stationary agent. While the maximum likelihood estimate is used in this work, other estimators can be used for computing $\bar{z}$ such as a least squares estimator.

Independent sensors If the sensors are independent, i.e., their measurements are statistically independent, then (4.8) becomes

$$\frac{\partial h(x)}{\partial x} = -\sum_{j=1}^{|\vec{z}|} \frac{(z_j - x)}{\sigma_j^2} = 0. \tag{4.13}$$

Hence,

$$\begin{aligned}\bar{z} &= \frac{1}{\sum_{i=1}^{|\vec{z}|} \frac{1}{\sigma_i^2}} \cdot \sum_{j=1}^{|\vec{z}|} \frac{z_j}{\sigma_j^2} \\ &= \frac{\mathbf{1}^T R^{-1}}{\operatorname{tr}(R^{-1})} \cdot \vec{z}.\end{aligned} \tag{4.14}$$

Comparing (4.10) and (4.14),

$$\Lambda = \frac{\mathbf{1}^T R^{-1}}{\operatorname{tr}(R^{-1})}, \tag{4.15}$$

thus

$$\begin{aligned}\sigma_{\bar{z}}^2 &= \Lambda R \Lambda^T \\ &= \frac{\mathbf{1}^T R^{-1}}{\mathrm{tr}(R^{-1})} R \frac{(R^{-1})^T}{\mathrm{tr}(R^{-1})} \mathbf{1} \\ &= \frac{1}{\mathrm{tr}^2(R^{-1})} \mathbf{1}^T R^{-1} \mathbf{1} \\ &= \frac{1}{\mathrm{tr}(R^{-1})}. \end{aligned} \tag{4.16}$$

4.1.2.2 Colocated stationary agents with random mobile object parameter

The case of placing multiple stationary agents at the same location with a random mobile object parameter is now studied. It is assumed that the mobile object parameter w is Gaussian with mean $\bar{w}$ and variance σ_w^2. A variety of estimators exist to estimate a random parameter such as the maximum a posteriori estimator or the minimum mean square error estimator, in this work the maximum a posteriori estimator is used. The maximum a posteriori estimate of the sensors of the stationary agents, $\bar{z}$, is:

$$\begin{aligned}\bar{z} &= \underset{x}{\mathrm{argmax}}(p(\vec{z}|x) \cdot p(x)) \\ &= \underset{x}{\mathrm{argmax}}\left(\frac{1}{2\pi\sigma_w \det(R)} e^{-\frac{(\vec{z}-\vec{x})^T R^{-1}(\vec{z}-\vec{x})}{2} - \frac{(x-\bar{w})^2}{2\sigma_w^2}}\right) \\ &= \underset{x}{\mathrm{argmin}}\left(\frac{(\vec{z}-\vec{x})^T R^{-1}(\vec{z}-\vec{x})}{2} + \frac{(x-\bar{w})^2}{2\sigma_w^2}\right). \end{aligned} \tag{4.17}$$

Independent sensors If the sensors carried by the stationary agents are independent, then (4.17) becomes

$$\begin{aligned}
\bar{z} &= \underset{x}{\operatorname{argmin}}\left(\frac{(x-\bar{w})^2}{\sigma_w^2}+\sum_{j=1}^{|\vec{z}|}\frac{(z_j-x)^2}{\sigma_j^2}\right)\\
&= \underset{x}{\operatorname{argmin}}\left(x^2\left(\frac{1}{\sigma_w^2}+\sum_{j=1}^{|\vec{z}|}\frac{1}{\sigma_j^2}\right)-2x\left(\frac{\bar{w}}{\sigma_w^2}+\sum_{j=1}^{|\vec{z}|}\frac{z_j}{\sigma_j^2}\right)+\frac{\bar{w}^2}{\sigma_w^2}+\sum_{j=1}^{|\vec{z}|}\frac{z_j^2}{\sigma_j^2}\right)\\
&= \underset{x}{\operatorname{argmin}}\left(x^2\left(\frac{1+\sigma_w^2\operatorname{tr}(R^{-1})}{\sigma_w^2}\right)-2x\left(\frac{\bar{w}}{\sigma_w^2}+\mathbf{1}^T R^{-1}\vec{z}\right)+\frac{\bar{w}}{\sigma_w^2}+\mathbf{1}^T R^{-1}\vec{z^2}\right)\\
&= \underset{x}{\operatorname{argmin}}\left(\left(x-\frac{\bar{w}+\sigma_w^2\mathbf{1}^T R^{-1}\vec{z}}{1+\sigma_w^2\operatorname{tr}(R^{-1})}\right)^2+\frac{\bar{w}^2+\sigma_w^2\mathbf{1}^T R^{-1}\vec{z^2}}{1+\sigma_w^2\operatorname{tr}(R^{-1})}-\frac{\left(\bar{w}+\sigma_w^2\mathbf{1}^T R^{-1}\vec{z}\right)^2}{\left(1+\sigma_w^2\operatorname{tr}(R^{-1})\right)^2}\right)\\
&= \underset{x}{\operatorname{argmin}}\left(x-\frac{\bar{w}+\sigma_w^2\mathbf{1}^T R^{-1}z}{1+\sigma_w^2\operatorname{tr}(R^{-1})}\right)^2. \qquad (4.18)
\end{aligned}$$

Hence,

$$\bar{z}=\frac{\bar{w}+\sigma_w^2\mathbf{1}^T R^{-1}z}{1+\sigma_w^2\operatorname{tr}(R^{-1})}=\begin{bmatrix}\frac{1}{1+\sigma_w^2\operatorname{tr}(R^{-1})} & \frac{\sigma_w^2\mathbf{1}^T R^{-1}}{1+\sigma_w^2\operatorname{tr}(R^{-1})}\end{bmatrix}\begin{bmatrix}\bar{w}\\ z\end{bmatrix}=\Lambda\begin{bmatrix}\bar{w}\\ z\end{bmatrix}, \qquad (4.19)$$

where $\Lambda=\begin{bmatrix}\frac{1}{1+\sigma_w^2\operatorname{tr}(R^{-1})} & \frac{\sigma_w^2\mathbf{1}^T R^{-1}}{1+\sigma_w^2\operatorname{tr}(R^{-1})}\end{bmatrix}$. Thus the maximum a posteriori estimate is also Gaussian:

$$\bar{z}=\mathcal{N}\left(\frac{\bar{w}}{1+\sigma_w^2\operatorname{tr}(R^{-1})},\sigma_{\bar{z}}^2\right), \qquad (4.20)$$

where

$$\sigma_{\bar{z}}^2=\Lambda\tilde{R}\Lambda^T \qquad (4.21)$$

and

$$\tilde{R}=\begin{bmatrix}\sigma_w^2 & 0\\ 0 & R\end{bmatrix}. \qquad (4.22)$$

Therefore, the variance of the maximum a posteriori estimate is

$$\begin{aligned}
\sigma_{\bar{z}}^2 &= \frac{1}{(1+\sigma_w^2 tr(R^{-1}))^2} \begin{bmatrix} 1 & \sigma_w^2 \mathbf{1}^T R^{-1} \end{bmatrix} \tilde{R} \begin{bmatrix} 1 \\ \sigma_w^2 R^{-1} \mathbf{1} \end{bmatrix} \\
&= \frac{\sigma_w^2}{1+\sigma_w^2 \operatorname{tr}(R^{-1})}. \qquad (4.23)
\end{aligned}$$

4.2 Problem formulation

Let g be the number of stationary agents available, where $1 \leq g \leq n^2$. The location of the stationary agents and the revisit deadline of the stationary agents, r_i, can be selected.

The problems of stationary agent placement and stationary agent visit timing are formulated as follows:

1. Given a directed graph, a Markov model of the mobile object's movement on the graph, and a set of stationary agents; select the locations of the stationary agents such that
   (a) Fisher information is maximized, or
   (b) probability of misclassification is minimized, or
   (c) penalty incurred by poor detections resulting from the placement is minimized.
2. Given a directed graph, a Markov model of the mobile object's movement on the graph, and a set of placed stationary agents; select the revisit deadlines for mobile agents visiting such that the amount of time a vehicle visit to a sensor precedes a phenomenon visit and the time elapsed before a detection from a sensor is acquired by a visiting vehicle are minimized.

4.3 Stationary agent placement

In this section, the problem of placing the stationary agents is treated. Let U be a binary matrix of size $g \times n^2$ representing a configuration of stationary agents along the graph where a 1 in the $(i, j)^{th}$ entry indicates that stationary agent i has been placed at location j in the network. Let U_i indicate the i^{th} row of U corresponding to the location of stationary agent i. Several methods to place stationary agents are now introduced.

4.3.1 Background

Several properties of functions used for the placement of stationary agents are now introduced. Let f be a function defined over the set Ω where $f : 2^{\Omega} \to \mathbb{R}$. Let Γ and Ψ be subsets of Ω such that $\Gamma \subseteq \Psi \subseteq \Omega$.

4.3.1.1 Monotonicity

f is monotonic if

$$f(\Gamma) \leq f(\Psi). \tag{4.24}$$

If f is the function describing the utility of a sensor placement, then f being monotonic implies placing more sensors is never detrimental.

4.3.1.2 Submodularity

f is submodular if

$$\forall x \in \Omega \setminus \Psi, f(\Gamma \cup \{x\}) - f(\Gamma) \geq f(\Psi \cup \{x\}) - f(\Psi). \tag{4.25}$$

If f is the function describing the utility of a sensor placement, then f being monotonic and submodular indicates that using a greedy algorithm yields a solution which is close to optimal.

4.3.2 Fisher information

Fisher information is the expected value of the information an observable variable z gives about an unknown variable y:

$$F = E\left[\left(\frac{\partial}{\partial z}\log f(y;z)\right)^2 |z\right] \tag{4.26}$$

where $f(y;z)$ is the conditional probability density function.

It is often used as an objective function in sensor placement problems or more generally optimal experimental design since it allows designers to quantify the information acquired through a sensor placement or experimental design without having to perform experiments. A related notion to Fisher information is the Cramer-Rao lower bound which is the lower bound of the variance of the error in the measurement; under certain conditions the Cramer-Rao lower bound is the inverse of Fisher information. Thus maximizing Fisher information is equivalent to minimizing the lower bound of the variance of the estimation.

To formulate Fisher information, a number of variables and functions are introduced. Let $\bar{\mathcal{P}} = \mathcal{P} + I$ where I is the identity matrix and $\bar{p}_{i,j}$ are elements in $\bar{\mathcal{P}}$. The likelihood of a sensor j located at (i_1, i_2) detecting a phenomenon currently located at node k is

$$f\left(z_j^{(i_1,i_2)}; w^{(k)}\right) = \frac{\bar{p}_{k,i_1} \cdot \bar{p}_{i_1,i_2}}{\sigma_j\sqrt{2\pi}} \cdot e^{-\frac{(z-w)^2}{2\sigma_j^2}}; \tag{4.27}$$

if $i_1 = i_2$ then the location of the sensor is node i_1, otherwise it is a directed edge from node i_1 to node i_2.

The corresponding Fisher information for stationary agent j placed at (i_1, i_2) is then:

$$F_{i_1,i_2}(U) = -E\left[\left(\frac{\partial}{\partial w}\log f\left(z_j^{(i_1,i_2)}; w^{(k)}\right)\right)^2\right],$$
$$= -E\left[\left(\frac{\partial f\left(z_j^{(i_1,i_2)}; w^{(k)}\right)}{\partial w}\frac{1}{f\left(z_j^{(i_1,i_2)}; w^{(k)}\right)}\right)^2\right],$$
$$= -\sum_{k=1}^{n}\pi_k\int\left(\frac{\partial f\left(z_j^{(i_1,i_2)}; w^{(k)}\right)}{\partial w}\right)^2\frac{1}{f\left(z_j^{(i_1,i_2)}; w^{(k)}\right)}dw, \tag{4.28}$$

where
$$\frac{\partial f\left(z_{(j)}^{(i_1,i_2)}; w^{(k)}\right)}{\partial w} = \frac{\bar{p}_{k,i_1}\cdot\bar{p}_{i_1,i_2}}{\sigma_j\sqrt{2\pi}}\cdot\frac{-2w+2z}{2\sigma_j^2}\cdot e^{\frac{-(z-w)^2}{2\sigma_j^2}}. \tag{4.29}$$

Then (4.28) becomes,

$$F_{i_1,i_2}(U) = -\sum_{k=1}^{n}\pi_k\int\left(\frac{\bar{p}_{k,i_1}\cdot\bar{p}_{i_1,i_2}}{\sigma_j\sqrt{2\pi}}\cdot\frac{-2w+2z}{2\sigma_j^2}\cdot e^{\frac{-(z-w)^2}{2\sigma_j^2}}\right)^2\cdot\frac{\sigma_j\sqrt{2\pi}}{\bar{p}_{k,i_1}\cdot\bar{p}_{i_1,i_2}\cdot e^{\frac{-(z-w)^2}{2\sigma_j^2}}}dw$$
$$= -\sum_{k=1}^{n}\pi_k\int\frac{\bar{p}_{k,i_1}^2\cdot\bar{p}_{i_1,i_2}^2}{\sigma_j^2 2\pi}\cdot\frac{(z-w)^2}{\sigma_j^4}\cdot e^{\frac{-2(z-w)^2}{2\sigma_j^2}}\cdot e^{\frac{(z-w)^2}{2\sigma_j^2}}\frac{\sigma_j\sqrt{2\pi}}{\bar{p}_{k,i_1}\cdot\bar{p}_{i_1,i_2}}dw$$
$$= -\sum_{k=1}^{n}\pi_k\int\frac{\bar{p}_{k,i_1}\cdot\bar{p}_{i_1,i_2}}{\sigma_j^5\sqrt{2\pi}}\cdot(z-w)^2 e^{\frac{-(z-w)^2}{2\sigma_j^2}}dw. \tag{4.30}$$

Let $x = z - w$, where $dx = -dw$, then

$$\begin{aligned}
F_{i_1,i_2}(U) &= \sum_{k=1}^{n} \pi_k \int \frac{\bar{p}_{k,i_1} \cdot \bar{p}_{i_1,i_2}}{\sigma_j^5 \sqrt{2\pi}} \cdot x^2 \cdot e^{\frac{-(x)^2}{2\sigma_j^2}}\, dx \\
&= \sum_{k=1}^{n} \pi_k \cdot \frac{\bar{p}_{k,i_1} \cdot \bar{p}_{i_1,i_2}}{\sigma_j^4} \int \frac{x^2}{\sigma_j \sqrt{2\pi}} \cdot e^{\frac{-(x)^2}{2\sigma_j^2}}\, dx \\
&= \sum_{k=1}^{n} \pi_k \cdot \frac{\bar{p}_{k,i_1} \cdot \bar{p}_{i_1,i_2}}{\sigma_j^4} \cdot \sigma_j^2 \\
&= \sum_{k=1}^{n} \pi_k \cdot \frac{\bar{p}_{k,i_1} \cdot \bar{p}_{i_1,i_2}}{\sigma_j^2} \\
&= \frac{1}{\sigma_j^2} \sum_{k=1}^{n} \pi_k \cdot \bar{p}_{k,i_1} \cdot \bar{p}_{i_1,i_2}. \qquad (4.31)
\end{aligned}$$

From the definition of π,

$$\begin{aligned}
\sum_{k=1}^{n} \pi_k \cdot \bar{p}_{k,i} &= \pi_i + \sum_{k=1}^{n} \pi_k \cdot p_{k,i}, \\
&= \pi_i + \pi_i = 2\pi_i. \qquad (4.32)
\end{aligned}$$

Then,

$$F_{i_1,i_2}(U) = \frac{2\pi_{i_1} \cdot \bar{p}_{i_1,i_2}}{\sigma_j^2}. \qquad (4.33)$$

It is important to note in (4.33) that $F_{i_1,i_2}(U) \geq 0$. When no stationary agent is present at location i_1, i_2, $F_{i_1,i_2}(U) = 0$. Placing stationary agents using Fisher information is thus

$$\max_U \left(\sum_{i_1=1}^{n} \sum_{i_2=1}^{n} F_{i_1,i_2}(U) \right) \text{s.t. } U_k \cdot \mathbf{1} \in \{0,1\}, 1 \leq k \leq g, \qquad (4.34)$$

where U_k is the k^{th} row in U.

4.3.3 Probability of misclassification

The probability of misclassification quantifies the rates of false positives and false negatives for a classifier. To describe the probability of misclassification, a model for classification is first needed. Thresholding is used as the method for classification in this work and let β_j be the threshold for stationary agent j's sensor. If $z_j \geq \beta_j$ then a detection takes place, otherwise no detection occurs.

The probability of misclassification for stationary agent j placed at (i_1, i_2) detecting a mobile object currently located at node k is

$$\begin{aligned} P_{m_{i_1,i_2}}(U) &= \left(\sum_{k=1}^{n} \pi_k \bar{p}_{k,i_1} \bar{p}_{i_1,i_2}\right) \cdot P\Big(z_j = q_j + w < \beta_j\Big) + \left(1 - \sum_{k=1}^{n} \pi_k \bar{p}_{k,i_1} \bar{p}_{i_1,i_2}\right) \cdot P\Big(z_j = q_j \geq \beta_j\Big) \\ &= (2\pi_{i_1} \bar{p}_{i_1,i_2}) \cdot P\Big(q_j < \beta_j - w) + (1 - 2\pi_{i_1} \bar{p}_{i_1,i_2}\Big) \cdot P\Big(q_j \geq \beta_j\Big) \\ &= (2\pi_{i_1} \bar{p}_{i_1,i_2}) \cdot \left(\frac{1}{2} + \frac{1}{2}\operatorname{erf}\left(\frac{\beta_j - w}{\sqrt{2}\sigma_j}\right)\right) + (1 - 2\pi_{i_1} \bar{p}_{i_1,i_2}) \cdot \left(\frac{1}{2} - \frac{1}{2}\operatorname{erf}\left(\frac{\beta_j}{\sqrt{2}\sigma_j}\right)\right). \end{aligned} \tag{4.35}$$

Let ρ_{i_1,i_2} indicate the prior probability for location (i_1, i_2) where $\rho_{i_1,i_2} := 2\pi_{i_1} \bar{p}_{i_1,i_2}$. The optimal threshold is computed by minimizing the probability of misclassification (theory and examples of thresholding for classification can be found in [114]),

$$\begin{aligned} \frac{\partial P_{m_{i_1,i_2}}(U)}{\partial \beta_j} &= 0 \\ \frac{\rho_{i_1,i_2}}{\sqrt{\pi}} \cdot \exp\left(-\frac{(\beta_j - w)^2}{2\sigma_j^2}\right) &= \frac{1 - \rho_{i_1,i_2}}{\sqrt{\pi}} \cdot \exp\left(-\frac{\beta_j^2}{2\sigma_j^2}\right) \\ \log(\rho_{i_1,i_2}) - \frac{(\beta_j - w)^2}{2\sigma_j^2} &= \log(1 - \rho_{i_1,i_2}) - \frac{\beta_j^2}{2\sigma_j^2} \\ 2\beta_j w - w^2 + 2\sigma_j^2 \cdot \log\left(\frac{\rho_{i_1,i_2}}{1 - \rho_{i_1,i_2}}\right) &= 0 \\ \beta_j &= \frac{w}{2} - \frac{\sigma_j^2}{w} \cdot \log\left(\frac{\rho_{i_1,i_2}}{1 - \rho_{i_1,i_2}}\right). \end{aligned} \tag{4.36}$$

The second order condition for a minimum is indeed satisfied:

$$\begin{aligned}\frac{\partial^2 P_{m_{i_1,i_2}}(U)}{\partial\beta_j^2} &= \frac{-\rho_{i_1,i_2}}{\sqrt{\pi}\sigma_j^2}\left((\beta_j-w)\cdot\exp\left(-\frac{(\beta_j-w)^2}{2\sigma_j^2}\right)+\beta_j\cdot\exp\left(-\frac{\beta_j^2}{2\sigma_j^2}\right)+\frac{\beta_j}{\sqrt{\pi}\sigma_j^2}\cdot\exp\left(-\frac{\beta_j^2}{2\sigma_j^2}\right)\right)\\ &= \frac{w\cdot\sqrt{\rho_{i_1,i_2}(1-\rho_{i_1,i_2})}}{\sqrt{\pi}\sigma_j^2}\cdot\exp\left(-\frac{w^2}{8\sigma_j^2}-\frac{\sigma_j^2}{2w^2}\cdot\left(\log\left(\frac{\rho_{i_1,i_2}}{1-\rho_{i_1,i_2}}\right)\right)^2\right)\\ &\geq 0. \end{aligned} \tag{4.37}$$

The probability of misclassification score for an entire placement is computed by taking the sum of the probability of misclassification across all nodes and edges in the graph (the probability of misclassification for a location without a stationary agent is the prior probability for that location). Placing stationary agents using the probability of misclassification score is thus

$$\min_U\left(\sum_{i_1=1}^{n}\sum_{i_2=1}^{n}P_{m_{i_1,i_2}}(U)\right)\text{s.t. } U_k\cdot\mathbf{1}=\{0,1\}, 1\leq k\leq g. \tag{4.38}$$

4.3.4 Penalty incurred

When placing stationary agents at nodes and along edges, the quality of the ensuing detection should be accounted for (i.e., how well the mobile agent's position can be predicted from the detection or if the detection is redundant). The penalty of a placement of stationary agents is thus minimized. Penalty is defined here as the product of the quality of the detection and the probability of the stationary agent missing a detection or falsely triggering a detection. Nodes and edges yield different types of detections. A detection from a node occurs more often than a detection along a connected edge, but only gives an estimate of where the mobile object's next location is (for example, the solid highlighted area around node 3 in Figure 4.1). A detection from an edge occurs less often than a detection from its source node, however it gives the exact location of the mobile object for a certain amount of time (for example, the dashed highlighted edge from node 2 to node 4 in Figure 4.1).

The penalty function for node i, $V_{i,i}(U)$, is a weighted sum of the product of the uncov-

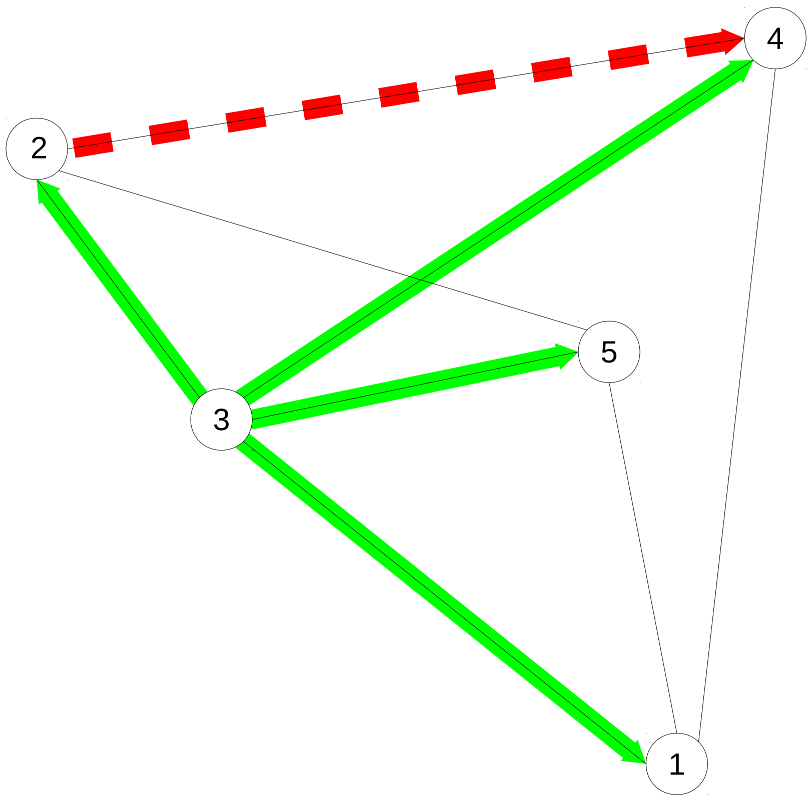


Figure 4.1: Quality of detections from nodes and edges.

ered connected edge lengths, the likelihood of the mobile object visiting the node, and the probability of missed detections and false alarms for the node. An uncovered location is a location without a stationary agent. An intermediate variable is introduced to simplify the notation in the formula for $V_{i,i}(U)$. Let ϕ_i be the number of uncovered outgoing edges from node i:

$$\phi_i = \sum_{\substack{k=1 \\ k \neq i \\ p_{i,k}>0}}^{n} (1 - U_{i,k}) . \tag{4.39}$$

Then the penalty for sensor j at node i can be found using the following equation:

$$V_{i,i}(U) = \left(P\left(q_j < \beta_j - w\right) + P\left(q_j \geq \beta_j\right)\right) \cdot \pi_i \cdot \sum_{j=1}^{n} L_{i,j} \cdot \left(p_{i,j}\right)^{\phi_i} . \tag{4.40}$$

It is important to note that the penalty of a node depends on the placement of other stationary agents in the network. The penalty function for stationary agent j placed at edge (i_1, i_2) is the length of the edge scaled by the probability of the mobile object traveling along that edge and the probability of missed detections and false alarms for the edge:

$$V_{i_1,i_2}(U) = \Big(P\big(q_j < \beta_j - w\big) + P\big(q_j \geq \beta_j\big)\Big) \cdot \pi_{i_1} \cdot p_{i_1,i_2} \cdot L_{i_1,i_2}. \tag{4.41}$$

The goal for the placement is to minimize the overall penalty, i.e.,

$$\min_U \left(\sum_{i_1=1}^{n} \sum_{i_2=1}^{n} V_{i_1,i_2}(U) \right) \text{s.t. } U_k \cdot \mathbf{1} = \{0,1\}, 1 \leq k \leq g. \tag{4.42}$$

4.4 Special cases

In this section, characteristics of placements using the aforementioned metrics under certain cases are discussed.

4.4.1 No combined stationary agents

Characteristics of the different stationary agent placement schemes when stationary agents are not colocated are described here.

4.4.1.1 Monotonicity and submodularity using Fisher information

Let Γ and Ψ be sets of alarm sensors such that $\Gamma \subseteq \Psi$. Let $\mathcal{F}$ be the total Fisher information for a given sensor placement, e.g., $\mathcal{F}(\Gamma) = \sum_{i_1=1}^{n} \sum_{i_2=1}^{n} F_{i_1,i_2,}(U,\Gamma)$ where the argument Γ indicates that the sensor parameters come from the set of sensors Γ.

Proposition 4.4.1. *Sensor placement using Fisher information when sensors are not colocated is monotonic.*

Proof. $\Gamma \subseteq \Psi$ hence $\mathcal{F}(\Psi) = \mathcal{F}(\Gamma) + \mathcal{F}(\Psi \setminus \Gamma)$. However, (4.33) indicates that $F_{i_1,i_2}(U) \geq 0$, thus $\mathcal{F}(\Psi \setminus \Gamma) = \sum_{i_1=1}^{n} \sum_{i_2=1}^{n} F_{i_1,i_2,}(U, \Psi \setminus \Gamma) \geq 0$. Hence, $\mathcal{F}(\Gamma) \leq \mathcal{F}(\Psi)$, i.e., sensor placement using Fisher information when sensors are not colocated is monotonic (as defined in (4.24)).

□

Proposition 4.4.2. *Sensor placement using Fisher information when sensors are not colocated is submodular.*

Proof. Let x be a sensor that does not belong to the sets Γ or Ψ, i.e., $\{x\} \cap \Psi = \emptyset$. Let $\rho^\star$ be a vector containing the prior probabilities in descending magnitude, σ^Γ be a vector containing the sensor standard deviations for the set of sensors Γ in ascending magnitude, and σ^Ψ be a vector containing the sensor standard deviations for the set of sensors Ψ in ascending magnitude. According to (4.34), $\mathcal{F}(\Gamma) = \sum_{i=1}^{|\Gamma|} \frac{\rho_i^\star}{\sigma_i^{\Gamma 2}}$ and $\mathcal{F}(\Psi) = \sum_{i=1}^{|\Psi|} \frac{\rho_i^\star}{\sigma_i^{\Psi 2}}$. Let σ_x be the standard deviation of sensor x, then

$$\mathcal{F}(\Gamma \cup \{x\}) - \mathcal{F}(\Gamma) = \frac{\rho^\star_{|\Gamma|+1}}{{\sigma_x}^2} \tag{4.43}$$

and

$$\mathcal{F}(\Psi \cup \{x\}) - \mathcal{F}(\Psi) = \frac{\rho^\star_{|\Psi|+1}}{{\sigma_x}^2}. \tag{4.44}$$

However, $\Gamma \subseteq \Psi$, hence $|\Gamma| \leq |\Psi|$ and $\rho^\star_{|\Gamma|+1} \geq \rho^\star_{|\Psi|+1}$. Thus, $\mathcal{F}(\Gamma \cup \{x\}) - \mathcal{F}(\Gamma) \geq \mathcal{F}(\Psi \cup \{x\}) - \mathcal{F}(\Psi)$, i.e., sensor placement using Fisher information when sensors are not colocated is submodular (as defined in (4.25)). □

4.4.1.2 Single stationary agent

Stationary agent placement using a single stationary agent is studied here.

Lemma 4.4.3. *When placing a single stationary agent, the set of optimal Fisher information placements is obtained by maximizing the prior probability.*

Proof. For a single stationary agent with variance σ^2, the placement that maximizes Fisher information is

$$\max_{i_1,i_2}\left(\frac{\rho_{i_1,i_2}}{\sigma^2}\right), \tag{4.45}$$

which is equivalent to maximizing the prior probability. □

Theorem 4.4.4. *When placing a single stationary agent, the set of optimal Fisher information placements is a subset of the optimal probability of misclassification score placements.*

Proof. The prior probability is assumed to be continuous and is now indicated by ρ, the sensor noise is q with standard deviation σ, and the sensor's classification threshold is β. The probability of misclassification score is:

$$\begin{aligned}\bar{P}_m &= \sum_{i_1=1}^{n}\sum_{i_2=1}^{n} P_{m_{i_1,i_2}}(U) \\ &= \rho \cdot P(q < \beta - \bar{w}) + (1-\rho)\cdot(1 + P(q > \beta)) \\ &= \rho \cdot \left(\frac{1}{2} + \frac{1}{2}\operatorname{erf}\left(\frac{(\beta(\rho) - \bar{w})}{\sqrt{2}\sigma}\right)\right) + (1-\rho)\cdot\left(\frac{3}{2} - \frac{1}{2}\operatorname{erf}\left(\frac{\beta(\rho)}{\sqrt{2}\sigma}\right)\right). \end{aligned} \tag{4.46}$$

Using (4.36) in (4.46),

$$\begin{aligned}\bar{P}_m = \rho \cdot &\left(\frac{1}{2} + \frac{1}{2}\operatorname{erf}\left(-\frac{w}{2\sqrt{2}\sigma} - \frac{\sigma}{\sqrt{2}w}\cdot\log\left(\frac{\rho}{(1-\rho)}\right)\right)\right) \\ &+ (1-\rho)\cdot\left(\frac{3}{2} - \frac{1}{2}\operatorname{erf}\left(\frac{w}{2\sqrt{2}\sigma} - \frac{\sigma}{\sqrt{2}w}\cdot\log\left(\frac{\rho}{(1-\rho)}\right)\right)\right), \end{aligned} \tag{4.47}$$

which yields,

$$\begin{aligned}
\frac{\partial \bar{P}_m}{\partial \rho} &= \frac{1}{2}\operatorname{erf}\left(\frac{\bar{w}}{2\sqrt{2}\sigma} - \frac{\sigma}{\sqrt{2}\bar{w}}\log\left(\frac{\rho}{1-\rho}\right)\right) - \frac{1}{2}\operatorname{erf}\left(\frac{\bar{w}}{2\sqrt{2}\sigma} + \frac{\sigma}{\sqrt{2}\bar{w}}\log\left(\frac{\rho}{1-\rho}\right)\right) \\
&\quad - \frac{\sigma}{(1-\rho)\sqrt{2\pi}\bar{w}}\cdot\exp\left(\frac{-1}{2\sigma^2}\left(\frac{\bar{w}}{2} + \frac{\sigma^2}{\bar{w}}\log\left(\frac{\rho}{1-\rho}\right)\right)^2\right) \\
&\quad + \frac{\sigma}{\rho\sqrt{2\pi}\bar{w}}\cdot\exp\left(\frac{-1}{2\sigma^2}\left(\frac{\bar{w}}{2} - \frac{\sigma^2}{\bar{w}}\log\left(\frac{\rho}{1-\rho}\right)\right)^2\right) - 1 \\
&= \frac{1}{2}\operatorname{erf}\left(\frac{\bar{w}}{2\sqrt{2}\sigma} - \frac{\sigma}{\sqrt{2}\bar{w}}\log\left(\frac{\rho}{1-\rho}\right)\right) - \frac{1}{2}\operatorname{erf}\left(\frac{\bar{w}}{2\sqrt{2}\sigma} + \frac{\sigma}{\sqrt{2}\bar{w}}\log\left(\frac{\rho}{1-\rho}\right)\right) \\
&\quad + \frac{\sigma}{\sqrt{2\pi}\bar{w}}\cdot\exp\left(-\frac{\bar{w}^2}{8\sigma^2} - \frac{\sigma^2}{2\bar{w}^2}\left(\log\left(\frac{\rho}{1-\rho}\right)\right)^2\right)\cdot\left(-\frac{1}{\sqrt{1-\rho}\sqrt{\rho}} + \frac{1}{\sqrt{\rho}\sqrt{1-\rho}}\right) - 1 \\
&= \frac{1}{2}\operatorname{erf}\left(\frac{\bar{w}}{2\sqrt{2}\sigma} - \frac{\sigma}{\sqrt{2}\bar{w}}\log\left(\frac{\rho}{1-\rho}\right)\right) - \frac{1}{2}\operatorname{erf}\left(\frac{\bar{w}}{2\sqrt{2}\sigma} + \frac{\sigma}{\sqrt{2}\bar{w}}\log\left(\frac{\rho}{1-\rho}\right)\right) - 1. \quad (4.48)
\end{aligned}$$

We can rewrite this expression as

$$\frac{\partial \bar{P}_m}{\partial \rho} = \frac{1}{2}\operatorname{erf}(\xi) - \frac{1}{2}\operatorname{erf}(\zeta) - 1, \quad (4.49)$$

where $\xi \in \mathbb{R}$ and $\zeta \in \mathbb{R}$ are the corresponding large expressions in the equation above. Note that $-1 \leq \operatorname{erf}(\cdot) \leq 1$ hence,

$$-1 \leq \frac{\operatorname{erf}(\xi) - \operatorname{erf}(\zeta)}{2} \leq 1. \quad (4.50)$$

Thus,

$$\frac{\partial \bar{P}_m}{\partial \rho} = \frac{1}{2}\operatorname{erf}(\xi) - \frac{1}{2}\operatorname{erf}(\zeta) - 1 \leq 0, \quad (4.51)$$

which means that when placing a single stationary agent maximizing the prior probability minimizes the probability of misclassification score. The optimal Fisher placements for a single stationary agent are obtained by maximizing the prior probability (Lemma 4.4.3). Therefore, for a given problem the set of optimal Fisher placements is a subset of the set of optimal misclassification score placements. □

The penalty incurred objective function depends on the length of the edges, hence similar behavior is not witnessed when placing stationary agents using penalty incurred.

4.4.2 Computational complexity

Computational aspects of placing the stationary agents are described here. There are $\frac{(n+|E|)!}{(n+|E|-g)!}$ candidate solutions, thus performing an exhaustive search quickly becomes infeasible for large problems. Instead of performing an exhaustive search, a k-opt approach can be used by switching 0s and 1s in a given candidate solution U; e.g., for $k = 2$ a stationary agent is moved from a covered location to a previously uncovered location while the rest of the placement remains the same. The number of candidate solutions for k-opt is $\frac{g!}{(g-k)!} \times \frac{(n+|E|-g)!}{(n+|E|-g-k)!}$ which can be less than the total number of candidate solutions for certain k.

To use k-opt an initial candidate solution is needed; this candidate solution can be selected or obtained by generating a random solution. A good initial candidate solution for Fisher information is the solution when stationary agents are not permitted to be colocated. The optimal solution is either this solution or a solution with a few switches to place multiple stationary agents on locations of more importance. In addition, Fisher placement when stationary agents are not permitted to be colocated is monotonic and submodular (Section 4.4.1.1), thus this initial candidate solution can be computed quickly using a greedy algorithm., e.g., place the sensor with the smallest variance on the most visited location and repeat this procedure until all sensors have been placed.

4.5 Stationary agent timing

In the previous section, the placement of stationary agents was discussed. However, it is still unclear how often mobile agents should visit the stationary agents. A method to select revisit deadlines for the placed stationary agents is now presented.

A stale stationary agent detection is less valuable than a recent stationary agent detection; as such when selecting revisit deadlines for stationary agents, they should be selected such that the time between a detection and its acquisition by a mobile agent is minimized. Let $e_i(t)$ be the age of a detection (not yet acquired by the mobile agents) at stationary agent i at time t. The maximum age of a detection is the revisit deadline of the stationary agent from which the detection came,

$$e_i(t) \leq r_i, \forall t, 1 \leq i \leq g. \tag{4.52}$$

4.5.1 Lower bound

To maintain the most accurate information possible about mobile object movements, the age of a detection at time of acquisition should be zero. This can only be guaranteed by setting all the stationary agents' revisit deadlines to zero,

$$r_i = 0 \Rightarrow e_i(t) = 0, \forall t, 1 \leq i \leq g. \tag{4.53}$$

Let m be the number of mobile agents, $r_i = 0, 1 \leq i \leq g$ if and only if $m = g$. However, it remains unclear how the revisit deadlines should be selected when $m < g$.

4.5.2 Feasibility of revisit deadlines

It is desirable that the set of revisit deadlines selected be feasible, i.e., paths for the mobile agents such that all revisit deadlines are met exist. Assessing the feasibility of a set of revisit deadlines for multiple mobile agents is NP-hard according to Proposition 3.3.1. As such, finding constraints that guarantee the existence of paths for a given set of revisit deadlines is difficult and thus feasibility of a set of revisit deadlines is not addressed when they are selected (heuristics such as the ones presented in Chapter 3 can be used by the mobile agents to satisfy the revisit deadlines or minimize the amount by which they are missed).

4.5.3 Selecting revisit deadlines

Let b_i be the expected recurrence time for the mobile object at stationary agent i, i.e., the expected time between two consecutive visits to the stationary agent by the mobile object (a method to compute expected recurrence times is given in Section 4.5.3.1). We want to minimize the expected age of a detection at stationary agent i, $e_i = b_i - r_i$,

$$\min_{r_i}(b_i - r_i), \tag{4.54}$$

and the expected amount of time by which a mobile agent visit pre-empts a mobile object visit,

$$\min_{r_i}(r_i - b_i). \tag{4.55}$$

This is equivalent to maximizing the reward and minimizing wasted resources for the mobile agents or minimizing the lead and lag time of mobile agent visits to mobile object visits. The Pareto optimal solution for the minimizations in (4.54) and (4.55) is

$$r_i = b_i. \tag{4.56}$$

4.5.3.1 Expected mobile object recurrence time

Let b_i be the expected mobile object recurrence time at node i. Intermediate variables are needed to compute b_i. $s_{j,i}$ is introduced: the expected time for the mobile object to travel from node j to node i (where $s_{j,i} = 0$ when $j = i$). Let $t_{i,j}$ be the time it takes the mobile object to move from node i to node j; then b_i can be written as follows,

$$b_i = \sum_{j=1}^{n} p_{i,j} \cdot \left(t_{i,j} + s_{j,i}\right), 1 \leq i \leq n. \tag{4.57}$$

$s_{j,i}$ must satisfy the following equation:

$$s_{j,i} = \sum_{k=1}^{n} p_{j,k} \cdot \left(t_{j,k} + s_{k,i}\right), 1 \leq i, j \leq n, j \neq i. \tag{4.58}$$

Equation (4.58) can be rewritten as a matrix equation, for example if $i = 1$ then

$$\begin{bmatrix} 1 & -p_{2,3} & \cdots & -p_{2,n} \\ -p_{3,2} & 1 & \cdots & -p_{3,n} \\ \vdots & \vdots & \ddots & \vdots \\ -p_{n,2} & p_{n,3} & \cdots & 1 \end{bmatrix} \begin{bmatrix} s_{2,1} \\ s_{3,1} \\ \vdots \\ s_{n,1} \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{n} p_{2,k} t_{2,k} \\ \sum_{k=1}^{n} p_{3,k} t_{3,k} \\ \vdots \\ \sum_{k=1}^{n} p_{n,k} t_{n,k} \end{bmatrix}. \tag{4.59}$$

Let $\mathcal{P}^o$ be a submatrix of the transition matrix where the o^{th} row and o^{th} column have been removed. The vector c contains the weighted sums of transition times in (4.58),

$$c = \begin{bmatrix} \sum_{k=1}^{n} p_{1,k} t_{1,k} \\ \sum_{k=1}^{n} p_{2,k} t_{2,k} \\ \vdots \\ \sum_{k=1}^{n} p_{n,k} t_{n,k} \end{bmatrix}, \tag{4.60}$$

and c^o is a subvector of c with the o^{th} element removed. Let s^o be a vector containing $s_{j,o}$ for $1 \leq j \leq n, j \neq o$. Equation (4.59) can then be rewritten for any i as

$$(I - \mathcal{P}^o) \cdot s^o = c^o. \tag{4.61}$$

Equation (4.61) can be used to compute s^o for $1 \leq o \leq n$ which can then be used to compute the recurrence times for nodes using (4.57).

The recurrence time for an edge (i, j), $b_{i,j}$, can be computed by using the recurrence time for the source node i and the likelihood of moving from i to j, as follows,

$$b_{i,j} = \frac{b_i}{p_{i,j}}. \tag{4.62}$$

4.6 Simulations

Three methods to place stationary agents were introduced in Section 4.3: the maximization of Fisher information (4.34), the minimization of the probability of misclassification (4.38), and the minimization of the penalty incurred by the placement (4.42); and a method to select stationary agent revisit deadlines (4.56) was introduced in Section 4.5. In this section, the resulting placements and revisit deadlines of these three placement schemes are compared. Several terms are now introduced to aid in this comparison: stationary agent placements are **identical** if the same combination of stationary agents covers the same locations in the graph, stationary agent placements are **similar** if different combinations of stationary agents cover the same locations in the graph, and stationary agent placements are **distinct** if they are neither identical nor similar.

Example stationary agent placements for two different scenarios are shown in Figures 4.2 and 4.3 on a graph with 5 nodes, 20 edges, and 3 stationary agents with independent sensors. A letter and a subscript at a given location indicate that a stationary agent has been placed at that location; the letter indicates the stationary agent placement scheme (F: Fisher, M: Misclassification, P: Penalty) and the subscript indicates which stationary agent. The data used to generate these figures is provided in Section 4.6.3. The scenarios used in Figures 4.2 and 4.3 differ only in the transition matrix of the mobile object's motion and probability distribution of the mobile object's property. In Figure 4.2, the placements using the probability of misclassification and the penalty of the placement are identical while the placements using the probability of misclassification and Fisher information are similar. In Figure 4.3, the placements using Fisher information and the probability of misclassification are similar while the placements using Fisher information and the penalty incurred by the placement are distinct (in this case minimizing the penalty resulted in placing two stationary agents at the same location).

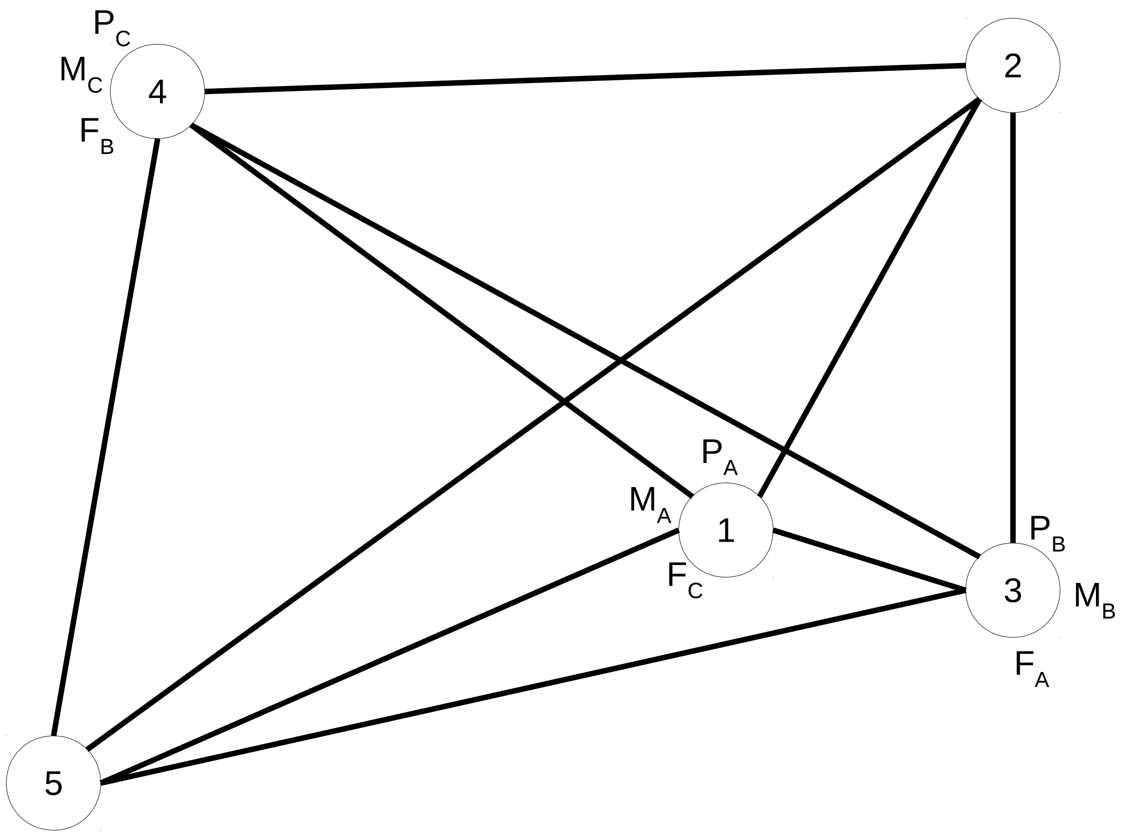


Figure 4.2: Example of identical and similar placements.

In these examples the stationary agents are placed on nodes; this feature is common for stationary agent placements because the likelihood of the mobile object visiting a node is often larger than that of visiting an edge. In addition, the stationary agents are placed on the most frequented nodes which is also common for stationary agents placements; since these nodes are weighted more heavily, they impact the optimization more significantly. These figures also show that different problem parameters (e.g., the transition matrix of the mobile object's motion or the probability distribution of the mobile object's property) result in different solutions which highlights the sensitivity of the stationary agent placements

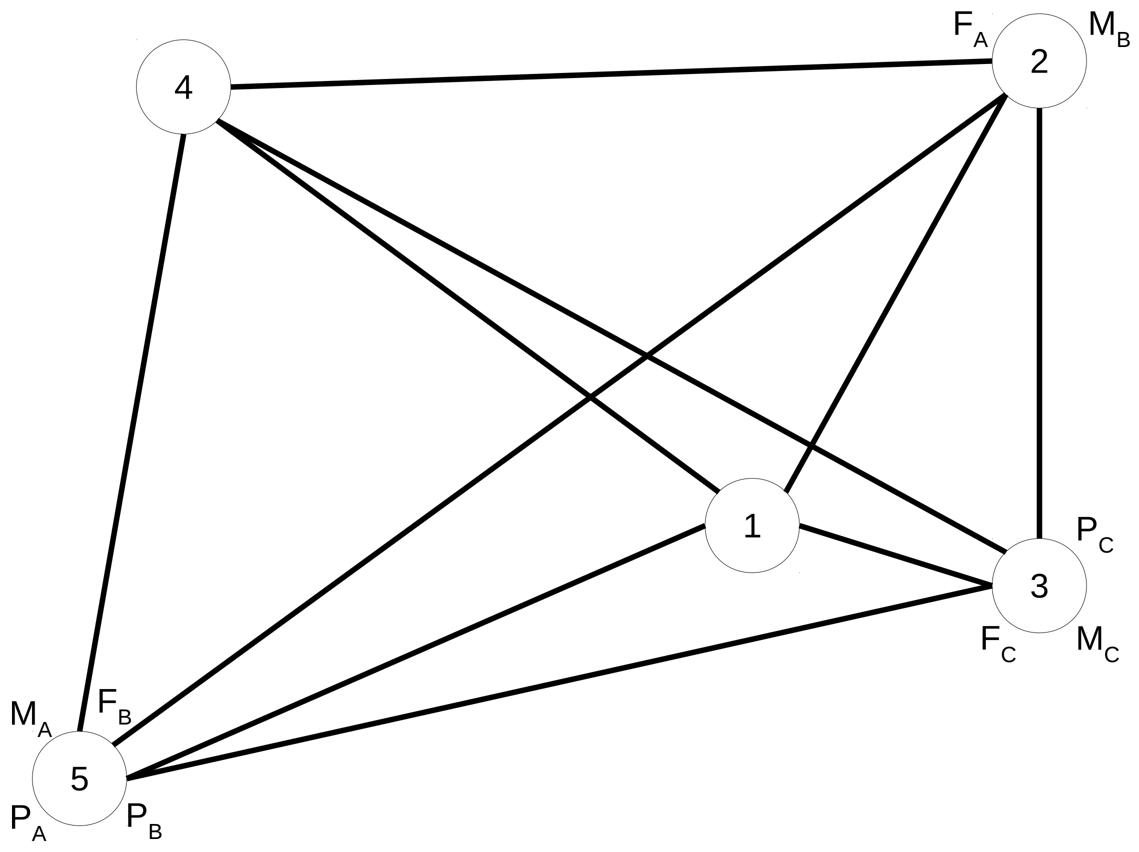


Figure 4.3: Example of similar and distinct placements.

with respect to the a priori knowledge about the mobile agent.

4.6.1 Comparison of stationary agent placements

The frequencies of occurrence of identical, similar, and distinct placements using Fisher information, probability of misclassification, and the penalty of a placement when using independent sensors are shown in Tables 4.1 and 4.2. These frequencies are computed using Monte Carlo simulations by holding the number of nodes, the number of available stationary agents, and the type of estimator constant while varying the stationary agents' sensor variances, the elements of the transition matrix of the mobile object, the lengths of the edges in the graph, and the parameters of the probability distribution of the mobile object's property. The 95% confidence intervals for the frequencies are shown in Tables 4.1

and 4.2 (indicated by the ± symbol). The elements of the transition matrix are generated using $\mathcal{U}(0,1)$ and scaled to ensure that each column sums up to one; the lengths of the edges in the graph are generated using $\mathcal{U}(0,15)$, while the sensor variances and parameters of the probability distribution of the mobile object's property are generated using $\mathcal{U}(0,100)$. In the simulations, the mobile object is assumed to travel with unit velocity. Table 4.1 contains a comparison of all three stationary agent placement schemes simultaneously while Table 4.2 contains pairwise comparisons of the stationary agent placement schemes. In the tables, 'MLE' indicates maximum likelihood estimator and 'MAP' indicates maximum a posteriori estimator.

n	g	Estimator	Simulations	Identical	Similar	Distinct
3	1	MLE	1092861	83.88%,±.07%	0%	16.12%,±.07%
3	1	MAP	1187295	89.67%,±.06%	0%	10.33%,±.06%
3	2	MLE	1561495	0%	0%	100%
3	2	MAP	1528106	11.83%,±.05%	75.72%,±.07%	12.45%,±.05%
3	3	MLE	299248	0%	0%	100%
3	3	MAP	300572	3.32%,±.06%	85.93%,±.12%	10.75%,±.11%
4	1	MLE	5896614	91.37%,±.02%	0%	8.63%,±.02%
4	1	MAP	6401624	94.49%,±.02%	0%	5.51%,±.02%
4	2	MLE	481414	0%	0%	100%
4	2	MAP	471840	12.29%,±.09%	78.85%,±.12%	8.86%,±.08%
4	3	MLE	32126	0%	0%	100%
4	3	MAP	31309	3.48%,±.20%	83.52%,±.41%	13%,±.37%

Table 4.1: Comparison of all stationary agent placements schemes.

As predicted by Theorem 4.4.4, stationary agent placements using Fisher information and probability of misclassification are identical when placing a single stationary agent as shown in Table 4.2. In addition, a majority of the three stationary agent placements are identical when placing a single stationary agent. In a minority of cases the placement using penalty incurred differs from those using Fisher information and probability of misclassification. Note that the placement using Fisher information when placing multiple stationary agents and using the maximum likelihood estimator is always distinct with respect to the probability of misclassification and penalty placements. This is because the maximization of Fisher information (4.34) subject to the use of a maximum likelihood estimator (4.16) leads to placing all the available stationary agents on the location most visited by the mobile object while the other two placement schemes do not share this feature.

4.6.2 Qualitative comparison of stationary agent placements

Several performance metrics are used to compare the quality of the stationary agent placements when using independent sensors: the percentage of false alarms emitted out of all possible false alarms emissions, the percentage of detections that are missed, and the mean detection age (i.e., the time between the detection and the next revisit deadline for the corresponding stationary agent) which quantifies the delay in the retrieval of the detection by the mobile agent. The first two quantities assess the performance of the detection and classification process while the latter quantity assesses the quality of the detection after it has occurred (accounting for the revisit deadlines). Results from Monte Carlo simulations are shown in Table 4.3. These simulations are performed in the same manner as in the previous subsection with the addition of simulating a roaming mobile object over the graph for 1000 steps. In Table 4.3, F, M, and P again indicate Fisher information, probability of misclassification, and penalty incurred, respectively. For each of the three metrics used for comparison, Table 4.3 shows the percentage of cases in which Fisher information, the probability of misclassification, and the penalty incurred obtained the minima for a given metric. The table also shows the percentage of cases where the three placements achieved the same value of a given metric. For example, in the first line of Table 4.3 the item under the column ‘Rate of false alarms’ and subcolumn ‘F<M,P’ shows that Fisher information obtained the minimum rate of false alarms (when compared to the probability of misclassification and penalty incurred placements) in 51.1% of simulations performed for placing two stationary agents on a graph with three nodes using a maximum likelihood estimator. Due to the large number of entries in Table 4.3, the 95% confidence intervals are not shown, they are all bounded between 0.195% and 3.099%.

When the property of the mobile object used for classification is known and thus the maximum likelihood estimator is used, it can be seen in Table 4.3 that stationary agent placement and timing using Fisher information achieve a minimum rate of false alarms, rate of missed detections, and mean detection age. In addition the rate at which Fisher information is the minimum is at least an order of magnitude larger than the other two placement schemes. Fisher information is the dominant stationary agent placement scheme in this case and thus should be used if false alarm rate, missed detection rate, and mean detection age are of importance.

When the property of the mobile object used for classification is not known and the maximum a posteriori estimator is used, Fisher information is no longer the dominant stationary agent placement scheme. In this case, the stationary agent placement schemes’ performance with respect to the three metrics are much closer (the differences are not in orders of magnitude). However, the penalty incurred scheme performs better than Fisher information and the probability of misclassification schemes in a majority of cases. Interestingly for both estimators, the probability of misclassification placement scheme rarely performs better than its peers using these three metrics for comparison.

4.6.3 Example data

$$L = \begin{bmatrix} 0 & 110.5 & 46.1 & 130.1 & 117.5 \\ 110.5 & 0 & 113.2 & 132.4 & 215.4 \\ 46.1 & 113.2 & 0 & 170.8 & 154.4 \\ 130.1 & 132.4 & 170.8 & 0 & 152 \\ 117.5 & 215.4 & 154.4 & 152 & 0, \end{bmatrix} \tag{4.63}$$

$$\begin{bmatrix} \sigma_A^2 & \sigma_B^2 & \sigma_C^2 \end{bmatrix} = \begin{bmatrix} 50.68 & 32.81 & 75.35 \end{bmatrix}. \tag{4.64}$$

4.6.3.1 Figure 4.2

$$\begin{bmatrix} \bar{w} & \sigma_w^2 \end{bmatrix} = \begin{bmatrix} 83.6 & 25.37 \end{bmatrix}, \tag{4.65}$$

$$\mathcal{P} = \begin{bmatrix} 0 & 0.335 & 0.39 & 0.002 & 0.273 \\ 0.328 & 0 & 0.122 & 0.408 & 0.142 \\ 0.017 & 0.057 & 0 & 0.49 & 0.436 \\ 0.346 & 0.275 & 0.364 & 0 & 0.015 \\ 0.201 & 0.124 & 0.238 & 0.437 & 0 \end{bmatrix}. \tag{4.66}$$

4.6.3.2 Figure 4.3

$$\begin{bmatrix} \bar{w} & \sigma_w^2 \end{bmatrix} = \begin{bmatrix} 2.676 & 42.53 \end{bmatrix}, \tag{4.67}$$

$$\mathcal{P} = \begin{bmatrix} 0 & 0.17 & 0.11 & 0.343 & 0.377 \\ 0.038 & 0 & 0.297 & 0.237 & 0.428 \\ 0.084 & 0.361 & 0 & 0.262 & 0.293 \\ 0.205 & 0.257 & 0.058 & 0 & 0.48 \\ 0.209 & 0.314 & 0.342 & 0.135 & 0 \end{bmatrix}. \tag{4.68}$$

4.7 Summary

In this chapter, a stationary agent placement and timing problem is formulated where the stationary agent configuration is optimized to monitor a mobile object moving on an arbitrary graph modeled as an ergodic Markov chain.

The stationary agents are placed by maximizing Fisher information, minimizing the probability of misclassification, or minimizing the penalty incurred by the placement. The stationary agents are timed (i.e., their revisit deadlines selected) by matching the revisit deadlines with the mobile object recurrence times at the stationary agent locations. The fusion of measurements from multiple colocated stationary agents on the graph is allowed to improve performance using a maximum likelihood estimator or a maximum a posteriori estimator. Properties of the stationary agent placement schemes are shown: when stationary agents are not colocated, placement using Fisher information is submodular and when placing a single stationary agent the set of optimal Fisher information placements is a subset of the optimal probability of misclassification placements.

The methods presented in this chapter can be used in the problem treated in Section 3.3 to allow for the stationary agents to optimally cooperate with mobile agents in the monitoring of a mobile object.

A question that arises once the stationary agents have been configured is whether the configuration is still optimal if the model of the object's motion changes. This question was studied in collaboration with Moritz Niendorf in [115]. Stability regions, i.e., the sets of perturbations to the problem data for which the current solution remains optimal, are derived for optimization problems where the problem data are in terms of Markov models. These stability regions are provided for problems with objective functions linear with respect to the initial distribution, the transition matrix, or the stationary distribution of the Markov model. This collaboration was continued in [116] where stability regions for objective functions linear in the product of the transition matrix and the stationary distribution, similar to the penalty incurred metric, are provided and criticality measures that assess the sensitivity of the objective function to perturbations in the problem data are given.

Table 4.2: Pairwise comparison of stationary agent placements schemes.

				Fisher-Misclassification			Fisher-Penalty			Misclassification-Penalty		
n	g	Estimator	Simulations	Identical	Similar	Distinct	Identical	Similar	Distinct	Identical	Similar	Distinct
3	1	MLE	1092861	100%	0%	0%	83.88%,±.07%	0%	16.12%,±.07%	83.88%,±.07%	0%	16.12%,±.07%
3	1	MAP	1187295	100%	0%	0%	89.67%,±.05%	0%	10.33%,±.05%	89.67%,±.05%	0%	10.33%,±.05%
3	2	MLE	1561495	0%	0%	100%	0%	0%	100%	46.1%,±.08%	35.66%,±.08%	18.24%,±.06%
3	2	MAP	1528106	12.95%,±.05%	86.51%,±.05%	0.54%,±.01%	40.67%,±.08%	46.88%,±.08%	12.45%,±.05%	58.24%,±.08%	29.81%,±.07%	11.95%,±.05%
3	3	MLE	299248	0%	0%	100%	0%	0%	100%	23.82%,±.15%	64.72%,±.17%	11.43%,±.11%
3	3	MAP	300572	3.56%,±.07%	93.76%,±.09%	2.68%,±.06%	6.93%,±.09%	82.32%,±.14%	10.75%,±.11%	39.84%,±.18%	51.92%,±.18%	8.24%,±.10%
4	1	MLE	5896614	100%	0%	0%	91.37%,±.02%	0%	8.63%,±.02%	91.37%,±.02%	0%	8.63%,±.02%
4	1	MAP	6401624	100%	0%	0%	94.49%,±.02%	0%	5.51%,±.02%	94.49%,±.02%	0%	5.51%,±.02%
4	2	MLE	481414	0%	0%	100%	0%	0%	100%	46.59%,±.14%	41.62%,±.14%	11.79%,±.09%
4	2	MAP	471840	13.32%,±.10%	85.61%,±.10%	1.07%,±.03%	44.68%,±.14%	46.46%,±.14%	8.86%,±.08%	58.54%,±.14%	33.61%,±.13%	7.85%,±.08%
4	3	MLE	32126	0%	0%	100%	0%	0%	100%	17.41%,±.41%	70.83%,±.50%	11.76%,±.35%
4	3	MAP	31309	3.87%,±.21%	90.84%,±.32%	5.29%,±.25%	13.76%,±.38%	73.28%,±.49%	12.96%,±.37%	34.48%,±.53%	57.27%,±.55%	8.24%,±.30%

Table 4.3: Comparison of quality of stationary agent configurations.

			Rate of false alarms				Rate of missed detections				Mean detection age			
n	g	Est.	F<M,P	M<F,P	P<F,M	F=M=P	F<M,P	M<F,P	P<F,M	F=M=P	F<M,P	M<F,P	P<F,M	F=M=P
3	2	MLE	51.1%	0%	3%	39.6%	24.5%	0.9%	8.5%	56.6%	89.7%	2.9%	2.9%	0%
3	2	MAP	7.8%	16.8%	7.3%	62.3%	6.1%	5.2%	7.5%	74.3%	9.1%	8.3%	9.3%	73.3%
4	2	MLE	56.3%	0.1%	1.3%	36.5%	29%	1.1%	5.8%	56.2%	84.5%	4.7%	3.3%	0%
4	2	MAP	11.2%	12.6%	8.3%	63.7%	6.6%	8.7%	4.4%	74.7%	9.1%	7.1%	10.9%	72.9%
3	3	MLE	61.8%	0%	0.1%	32%	37.7%	0.7%	3.2%	49.7%	97.8%	0.5%	1%	0%
3	3	MAP	7%	19%	9.8%	58.7%	8.7%	7.7%	6.9%	69.4%	14%	9%	9.7%	66.9%
4	3	MLE	61.2%	0%	0%	32.3%	39.3%	0.3%	1.9%	51.2%	95.2%	2.2%	0.8%	0%
4	3	MAP	15.4%	9.2%	9.5%	60.5%	7.6%	9.9%	5.7%	70%	15.1%	6.1%	11.9%	65.9%
3	4	MLE	66.1%	0%	0%	28.6%	41.9%	0.1%	2.3%	46.9%	99.8%	0.1%	0.1%	0%
3	4	MAP	9.6%	18.3%	9.2%	58.8%	8.8%	9.1%	6.7%	68.2%	21.1%	7.1%	8.5%	61.3%

CHAPTER 5

Paths for marsupial agents

In the previous chapters, path planning for mobile agents and configurations for stationary agents were discussed. The stability and sensitivity of the configuration of stationary agents was also briefly described. If the stability analysis shows that a current configuration of stationary agents is sub-optimal, then a question that arises is whether the configuration can be re-optimized. This can be achieved by mobile agents relocating the stationary agents. This is an instance of a task with marsupial vehicles, i.e., vehicles that can deploy and retrieve other vehicles.

A marsupial system involves agents that can deploy and retrieve other agents (carrier agents) as well as agents that can be deployed and retrieved by others (passenger agents). Nested marsupial systems, i.e., passenger agents that are also carrier agents, are not treated in this dissertation.

In this chapter, path planning for marsupial agents is investigated. The results from this chapter are used to address the marsupial operations question stated in Section 1.2: *how can mobile marsupial agents cooperate to achieve monitoring tasks?* A general problem formulation is given and properties of optimal paths are provided. A problem involving a pair of marsupial agents observing a single stationary object while minimizing their likelihood of being detected is then treated and the optimal paths for the agents are provided.

5.1 Optimal control for marsupial agents

Optimal control for a pair of marsupial agents monitoring objects is studied using a constrained optimization formulation. The objective function and termination constraints in this generalized study account for the proper monitoring of the objects.

5.1.1 Generic marsupial path planning problem formulation

Here a generic path planning problem for two marsupial agents to optimize some functional J is formulated. The two marsupial agents consist of a carrier agent and a passenger agent. Variables and conditions are introduced for the problem as well as functions defined over time t, where t_0 and t_f are the mission start and end time respectively. Let $x_c(t)$ and $v_c(t)$ be the position and velocity vectors of the carrier agent where $t \in [t_0, t_f]$. The passenger agent is released at time α and has position and velocity vectors, $x_p(t)$ and $v_p(t)$ respectively, where $t \in [\alpha, t_f]$. The combined state vector for both agents is indicated by $x(t)$ where

$$x(t) = \begin{cases} \begin{bmatrix} x_c(t) & v_c(t) \end{bmatrix}^T & , t \in [t_0, \alpha[, \\ \begin{bmatrix} x_c(t) & x_p(t) & v_c(t) & v_p(t) \end{bmatrix}^T & , t \in [\alpha, t_f]. \end{cases} \tag{5.1}$$

The boundary conditions for the agents are as follows:

$$x_c(t_0) = x_{c0}, \tag{5.2}$$

$$v_c(t_0) = v_{c0}, \tag{5.3}$$

$$x_c(t_f) = x_{cf}, \tag{5.4}$$

$$x_p(t_f) = x_{pf}. \tag{5.5}$$

The agents must be colocated at release:

$$x_p(\alpha) = x_c(\alpha). \tag{5.6}$$

The dynamics of the carrier agent are

$$\begin{bmatrix} \dot{x}_c(t) \\ \dot{v}_c(t) \end{bmatrix} = \begin{bmatrix} v_c(t) \\ f_c(x_c, v_c, u_c, t) \end{bmatrix}, t \in [t_0, t_f], \tag{5.7}$$

where $u_c(t)$ is the control variable for the carrier agent and $u_c(t) \in U_c$. Similarly, the dynamics of the passenger agent are

$$\begin{bmatrix} \dot{x}_p(t) \\ \dot{v}_p(t) \end{bmatrix} = \begin{bmatrix} v_p(t) \\ f_p(x_p, v_p, u_p, t) \end{bmatrix}, t \in [\alpha, t_f], \tag{5.8}$$

where $u_p(t)$ is the control variable for the passenger agent and $u_p(t) \in U_p$.

Let $u(t)$ be the combined control vector for both agents:

$$u(t) = \begin{cases} u_c(t) & , \ t \in [t_0, \alpha[, \\ \begin{bmatrix} u_c(t) & u_p(t) \end{bmatrix}^T & , \ t \in [\alpha, t_f]. \end{cases} \tag{5.9}$$

The objective function is

$$J[x, u, t, x_0, t_0, x_f, t_f] = K(x_0, t_0, x_f, t_f) + \int_{t_0}^{\alpha} L^c(x(t), u(t), t)dt + \int_{\alpha}^{t_f} L^{c+p}(x(t), v(t), u(t), t)dt \tag{5.10}$$

$$= K(x_0, t_0, x_f, t_f) + J_1[x, u, t] + J_2[x, u, t]. \tag{5.11}$$

Given the variables, functions, and boundary conditions introduced above, we seek to find the control u for which $J[x]$ has a weak extremum.

5.1.2 Costates and Hamiltonian

The Hamiltonian H is:

$$H(x,p,u,t)=\begin{cases}\begin{bmatrix}p_{x_c} & p_{v_c}\end{bmatrix}\begin{bmatrix}v_c\\ f_c\end{bmatrix}-L^c(x,u,t) & ,\ t\in[t_0,\alpha[,\\ \begin{bmatrix}p_{x_c} & p_{x_p} & p_{v_c} & p_{v_p}\end{bmatrix}\begin{bmatrix}v_c\\ v_p\\ f_c\\ f_p\end{bmatrix}-L^{c+p}(x,u,t) & ,\ t\in[\alpha,t_f],\end{cases} \tag{5.12}$$

where the costates, $p=\begin{cases}\begin{bmatrix}p_{x_c} & p_{v_c}\end{bmatrix}^T & ,\ t\in[t_0,\alpha[,\\ \begin{bmatrix}p_{x_c} & p_{x_p} & p_{v_c} & p_{v_p}\end{bmatrix}^T & ,\ t\in[\alpha,t_f],\end{cases}$ can be found using:

$$\dot{p}_{x_c}(t)=-\frac{\partial H(x,p,u,t)}{\partial x_c},t\in[t_0,t_f], \tag{5.13}$$

$$\dot{p}_{v_c}(t)=-\frac{\partial H(x,p,u,t)}{\partial v_c},t\in[t_0,t_f], \tag{5.14}$$

$$\dot{p}_{x_p}(t)=-\frac{\partial H(x,p,u,t)}{\partial x_p},t\in[\alpha,t_f], \tag{5.15}$$

and

$$\dot{p}_{v_p}(t)=-\frac{\partial H(x,p,u,t)}{\partial v_p},t\in[\alpha,t_f]. \tag{5.16}$$

5.1.3 Optimal control

The optimal control u can be found using Pontryagin's maximum principle:

$$u = \underset{v \in U}{argmax}(H(x,p,v,t)). \tag{5.17}$$

5.1.4 Natural boundary conditions

The agents' paths must satisfy the boundary conditions given in (5.2)-(5.5).

5.1.5 Transversality conditions

The transversality conditions for this problem are now studied. We study the first variation of the functionals as indicated in (9.69) of [117],

$$\delta J = \delta K + [p^T \delta x - H \delta t]_{t_0}^{t_f}. \tag{5.18}$$

We take first variation of J_1:

$$\delta J_1 = [p^T \delta x - H \delta t]_{t_0}^{t=\alpha} \tag{5.19}$$

$$= p_{x_c}\Big|_{t=\alpha^-} \delta x_{c\alpha} + p_{v_c}\Big|_{t=\alpha^-} \delta v_{c\alpha} - H\Big|_{t=\alpha^-} \delta\alpha - p_{x_c}\Big|_{t=t_0} \delta x_{c0} - p_{v_c}\Big|_{t=t_0} \delta v_{c0} + H\Big|_{t=t_0} \delta t_0. \tag{5.20}$$

The initial time t_0 is fixed and x_c, v_c are fully constrained at t_0 hence

$$\delta J_1 = p_{x_c}\Big|_{t=\alpha^-} \delta x_{c\alpha} + p_{v_c}\Big|_{t=\alpha^-} \delta v_{c\alpha} - H\Big|_{t=\alpha^-} \delta\alpha. \tag{5.21}$$

The first variation of J_2 is

$$\delta J_2 = [p^T \delta x - H \delta t]_{\alpha}^{t_f} \tag{5.22}$$

$$= p_{x_c}\Big|_{t=t_f} \delta x_{cf} + p_{x_p}\Big|_{t=t_f} \delta x_{pf} + p_{v_c}\Big|_{t=t_f} \delta v_{cf} + p_{v_p}\Big|_{t=t_f} \delta v_{pf} - H\Big|_{t=t_f} \delta t_f$$

$$- p_{x_c}\Big|_{t=\alpha^+} \delta x_{c\alpha} - p_{x_p}\Big|_{t=\alpha^+} \delta x_{p_\alpha} - p_{v_c}\Big|_{t=\alpha^+} \delta v_{c\alpha} - p_{v_p}\Big|_{t=\alpha^+} \delta v_{p_\alpha} + H\Big|_{t=\alpha^+} \delta\alpha. \tag{5.23}$$

Recall that $x_c(\alpha) = x_p(\alpha)$, hence $\delta x_{c\alpha} = \delta x_{p_\alpha}$. Thus, (5.22) can be rewritten as

$$\delta J_2 = p_{x_c}\Big|_{t=t_f} \delta x_{cf} + p_{x_p}\Big|_{t=t_f} \delta x_{pf} + p_{v_c}\Big|_{t=t_f} \delta v_{cf} + p_{v_p}\Big|_{t=t_f} \delta v_{pf} - H\Big|_{t=t_f} \delta t_f$$

$$- (p_{x_c} + p_{x_p})\Big|_{t=\alpha^+} \delta x_{c\alpha} - p_{v_c}\Big|_{t=\alpha^+} \delta v_{c\alpha} - p_{v_p}\Big|_{t=\alpha^+} \delta v_{p_\alpha} + H\Big|_{t=\alpha^+} \delta\alpha. \tag{5.24}$$

The first variation of K is

$$\delta K = \frac{\partial K}{\partial x_{c0}} \delta x_{c0} + \frac{\partial K}{\partial v_{c0}} \delta v_{c0} + \frac{\partial K}{\partial t_0} \delta t_0 + \frac{\partial K}{\partial x_{cf}} \delta x_{cf} + \frac{\partial K}{\partial x_{pf}} \delta x_{pf} + \frac{\partial K}{\partial v_{cf}} \delta v_{cf}$$

$$+ \frac{\partial K}{\partial v_{pf}} \delta v_{pf} + \frac{\partial K}{\partial t_f} \delta t_f. \tag{5.25}$$

The initial conditions t_0, x_{c0}, v_{c0} are fixed hence

$$\delta K = \frac{\partial K}{\partial x_{cf}} \delta x_{cf} + \frac{\partial K}{\partial x_{pf}} \delta x_{pf} + \frac{\partial K}{\partial v_{cf}} \delta v_{cf} + \frac{\partial K}{\partial v_{pf}} \delta v_{pf} + \frac{\partial K}{\partial t_f} \delta t_f. \tag{5.26}$$

The transversality conditions are

$$
\begin{aligned}
\delta J &= \delta K + \delta J_1 + \delta J_2 = 0 \\
&= \left(H\Big|_{t=\alpha^+} - H\Big|_{t=\alpha^-}\right)\delta\alpha + \left(p_{x_c}\Big|_{t=\alpha^-} - \left(p_{x_c}+p_{x_p}\right)\Big|_{t=\alpha^+}\right)\delta x_{c\alpha} + \left(p_{v_c}\Big|_{t=\alpha^-} - p_{v_c}\Big|_{t=\alpha^+}\right)\delta v_{c\alpha} \\
&+ \left(-p_{v_p}\Big|_{t=\alpha^+}\right)\delta v_{p_\alpha} + \left(\frac{\partial K}{\partial t_f} - H\Big|_{t=t_f}\right)\delta t_f + \left(p_{x_c}\Big|_{t=t_f} + \frac{\partial K}{\partial x_{cf}}\right)\delta x_{cf} + \left(p_{x_p}\Big|_{t=t_f} + \frac{\partial K}{\partial x_{pf}}\right)\delta x_{pf} \\
&+ \left(p_{v_c}\Big|_{t=t_f} + \frac{\partial K}{\partial v_{cf}}\right)\delta v_{cf} + \left(p_{v_p}\Big|_{t=t_f} + \frac{\partial K}{\partial v_{pf}}\right)\delta v_{pf} = 0 \qquad (5.27)
\end{aligned}
$$

for all admissible variations.

There are 9 transversality conditions in (5.27) and 9 free variables for this problem (α, $x_{c\alpha}$, $v_{c\alpha}$, v_{p_α}, t_f, x_{cf}, x_{pf}, v_{cf}, v_{pf}), hence the system is determined.

In the following subsections, a variety of related problems with additional constraints or conditions are considered. The expressions for the costates, the Hamiltonian, and the optimal control remain the same thus only the transversality conditions are given.

5.1.5.1 Constrained initial and final time and positions

The initial and final positions as well as final time are constrained, i.e., x_{cf}, x_{pf}, t_f are fixed, hence the first variation of the functionals are:

$$
\delta J_1 = p_{x_c}\Big|_{t=\alpha^-}\delta x_{c\alpha} + p_{v_c}\Big|_{t=\alpha^-}\delta v_{c\alpha} - H\Big|_{t=\alpha^-}\delta\alpha, \qquad (5.28)
$$

$$
\begin{aligned}
\delta J_2 = p_{v_c}\Big|_{t=t_f}\delta v_{cf} + p_{v_p}\Big|_{t=t_f}\delta v_{pf} - (p_{x_c}+p_{x_p})\Big|_{t=\alpha^+}\delta x_{c\alpha} - p_{v_c}\Big|_{t=\alpha^+}\delta v_{c\alpha} \\
- p_{v_p}\Big|_{t=\alpha^+}\delta v_{p_\alpha} + H\Big|_{t=\alpha^+}\delta\alpha, \qquad (5.29)
\end{aligned}
$$

and

$$\delta K = \frac{\partial K}{\partial v_{cf}} \delta v_{cf} + \frac{\partial K}{\partial v_{pf}} \delta v_{pf}. \tag{5.30}$$

The following transversality conditions are thus obtained:

$$\begin{aligned}
\delta J &= \delta K + \delta J_1 + \delta J_2 = 0 \\
&= \left(H\Big|_{t=\alpha^+} - H\Big|_{t=\alpha^-} \right) \delta\alpha + \left(p_{x_c}\Big|_{t=\alpha^-} - \left(p_{x_c} + p_{x_p} \right)\Big|_{t=\alpha^+} \right) \delta x_{c\alpha} + \left(p_{v_c}\Big|_{t=\alpha^-} - p_{v_c}\Big|_{t=\alpha^+} \right) \delta v_{c\alpha} \\
&+ \left(-p_{v_p}\Big|_{t=\alpha^+} \right) \delta v_{p_\alpha} + \left(p_{v_c}\Big|_{t=t_f} + \frac{\partial K}{\partial v_{cf}} \right) \delta v_{cf} + \left(p_{v_p}\Big|_{t=t_f} + \frac{\partial K}{\partial v_{pf}} \right) \delta v_{pf} = 0.
\end{aligned} \tag{5.31}$$

for all admissible variations.

There are 6 transversality conditions in (5.31) and 6 free variables for this problem (α, $x_{c\alpha}$, $v_{c\alpha}$, v_{p_α}, v_{cf}, v_{pf}), hence the system is determined.

5.1.5.2 Termination conditions

Let $g(x_f, t_f) = 0$ be termination conditions where $g(x_f, t_f) \in \mathbb{R}^m$. Using the results on target sets in [117], these conditions alter the transversality conditions at the final time and states as follows:

$$\frac{\partial K}{\partial x_{cf}} + p_{x_c}\Big|_{t=t_f} - \frac{\partial g}{\partial x_{cf}} \mu = 0, \tag{5.32}$$

$$\frac{\partial K}{\partial x_{pf}} + p_{x_p}\Big|_{t=t_f} - \frac{\partial g}{\partial x_{pf}} \mu = 0, \tag{5.33}$$

$$\frac{\partial K}{\partial v_{cf}} + p_{v_c}\Big|_{t=t_f} - \frac{\partial g}{\partial v_{cf}} \mu = 0, \tag{5.34}$$

$$\frac{\partial K}{\partial v_{pf}} + p_{v_p}\Big|_{t=t_f} - \frac{\partial g}{\partial v_{pf}} \mu = 0, \tag{5.35}$$

$$\frac{\partial K}{\partial t_f} - H\Big|_{t=t_f} - \frac{\partial g}{\partial t_f} \mu = 0, \tag{5.36}$$

where $\mu \in \mathbb{R}^m$.

The transversality conditions are

$$\begin{aligned}
\delta J &= \delta K + \delta J_1 + \delta J_2 = 0 \\
&= \left(H\Big|_{t=\alpha^+} - H\Big|_{t=\alpha^-} \right)\delta\alpha + \left(p_{x_c}\Big|_{t=\alpha^-} - \left(p_{x_c} + p_{x_p}\right)\Big|_{t=\alpha^+} \right)\delta x_{c\alpha} + \left(p_{v_c}\Big|_{t=\alpha^-} - p_{v_c}\Big|_{t=\alpha^+} \right)\delta v_{c\alpha} \\
&+ \left(-p_{v_p}\Big|_{t=\alpha^+} \right)\delta v_{p_\alpha} + \left(\frac{\partial K}{\partial t_f} - H\Big|_{t=t_f} - \frac{\partial g}{\partial t_f}\mu \right)\delta t_f + \left(p_{x_c}\Big|_{t=t_f} + \frac{\partial K}{\partial x_{cf}} - \frac{\partial g}{\partial x_{cf}}\mu \right)\delta x_{cf} \\
&+ \left(p_{x_p}\Big|_{t=t_f} + \frac{\partial K}{\partial x_{pf}} - \frac{\partial g}{\partial x_{pf}}\mu \right)\delta x_{pf} + \left(p_{v_c}\Big|_{t=t_f} + \frac{\partial K}{\partial v_{cf}} - \frac{\partial g}{\partial v_{cf}}\mu \right)\delta v_{cf} \\
&+ \left(p_{v_p}\Big|_{t=t_f} + \frac{\partial K}{\partial v_{pf}} - \frac{\partial g}{\partial v_{pf}}\mu \right)\delta v_{pf} = 0 \qquad (5.37)
\end{aligned}$$

for all admissible variations and

$$g(x_f, t_f) = 0. \qquad (5.38)$$

There are 9 transversality conditions in (5.37), m termination conditions in (5.38), and $(9+m)$ free variables in this problem (α, $x_{c\alpha}$, $v_{c\alpha}$, v_{p_α}, t_f, x_{cf}, x_{pf}, v_{cf}, v_{pf}, μ). Thus, the system is determined.

5.1.5.3 Release occurs on a locus of points

The release may need to occur on a locus of points, $l(x_c, \alpha) = 0$. Using the results on target sets in [117], this condition alters the transversality conditions at release as follows:

$$p_{x_c}\Big|_{t=\alpha^-} - \left(p_{x_c} + p_{x_p}\right)\Big|_{t=\alpha^+} - \frac{\partial l}{\partial x_{c\alpha}}\mu = 0, \qquad (5.39)$$

$$H\Big|_{t=\alpha^+} - H\Big|_{t=\alpha^-} - \frac{\partial l}{\partial \alpha}\mu = 0, \qquad (5.40)$$

where $\mu \in \mathbb{R}$.

The transversality conditions are

$$\begin{aligned}
\delta J &= \delta K + \delta J_1 + \delta J_2 = 0 \\
&= \left(H\Big|_{t=\alpha^+} - H\Big|_{t=\alpha^-} - \frac{\partial l}{\partial \alpha}\mu \right)\delta\alpha + \left(p_{x_c}\Big|_{t=\alpha^-} - \left(p_{x_c} + p_{x_p}\right)\Big|_{t=\alpha^+} - \frac{\partial l}{\partial x_{c\alpha}}\mu \right)\delta x_{c\alpha} \\
&\quad + \left(p_{v_c}\Big|_{t=\alpha^-} - p_{v_c}\Big|_{t=\alpha^+} \right)\delta v_{c\alpha} + \left(-p_{v_p}\Big|_{t=\alpha^+} \right)\delta v_{p_\alpha} + \left(\frac{\partial K}{\partial t_f} - H\Big|_{t=t_f} \right)\delta t_f \\
&\quad + \left(p_{x_c}\Big|_{t=t_f} + \frac{\partial K}{\partial x_{cf}} \right)\delta x_{cf} + \left(p_{x_p}\Big|_{t=t_f} + \frac{\partial K}{\partial x_{pf}} \right)\delta x_{pf} + \left(p_{v_c}\Big|_{t=t_f} + \frac{\partial K}{\partial v_{cf}} \right)\delta v_{cf} \\
&\quad + \left(p_{v_p}\Big|_{t=t_f} + \frac{\partial K}{\partial v_{pf}} \right)\delta v_{pf} = 0 \qquad (5.41)
\end{aligned}$$

for all admissible variations and

$$l(x_c, \alpha) = 0. \qquad (5.42)$$

There are 9 transversality conditions in (5.41), one release condition in (5.42), and 10 free variables in this problem (α, $x_{c\alpha}$, $v_{c\alpha}$, v_{p_α}, t_f, x_{cf}, x_{pf}, v_{cf}, v_{pf}, μ). Thus, the system of equations is determined.

5.1.5.4 Including recovery

When accounting for the recovery of the passenger agent, a retrieval time β is added where

$$x_p(\beta) = x_c(\beta), \qquad (5.43)$$

$$v_p(\beta) = v_c(\beta). \qquad (5.44)$$

The objective function is

$$\begin{aligned}
J[x,u,t,x_0,t_0,x_f,t_f] &= K(x_0,t_0,x_f,t_f) + \int_{t_0}^{\alpha} L^c(x(t),u(t),t)dt + \int_{\alpha}^{\beta} L^{c+p}(x(t),v(t),u(t),t)dt \\
&\quad + \int_{\beta}^{t_f} L^c(x(t),u(t),t)dt \qquad (5.45) \\
&= K(x_0,t_0,x_f,t_f) + J_1[x,u,t] + J_2[x,u,t] + J_3[x,u,t]. \qquad (5.46)
\end{aligned}$$

The Hamiltonian H is:

$$H(x,p,u,t) = \begin{cases} \begin{bmatrix} p_{x_c} & p_{v_c} \end{bmatrix} \begin{bmatrix} v_c \\ f_c \end{bmatrix} - L^c(x,u,t) & , t \in [t_0, \alpha[, \\ \begin{bmatrix} p_{x_c} & p_{x_p} & p_{v_c} & p_{v_p} \end{bmatrix} \begin{bmatrix} v_c \\ v_p \\ f_c \\ f_p \end{bmatrix} - L^{c+p}(x,u,t) & , t \in [\alpha, \beta], \\ \begin{bmatrix} p_{x_c} & p_{v_c} \end{bmatrix} \begin{bmatrix} v_c \\ f_c \end{bmatrix} - L^c(x,u,t) & , t \in]\beta, t_f], \end{cases} \tag{5.47}$$

where the costates, $p = \begin{cases} \begin{bmatrix} p_{x_c} & p_{v_c} \end{bmatrix}^T & , t \in [t_0, \alpha[, \\ \begin{bmatrix} p_{x_c} & p_{x_p} & p_{v_c} & p_{v_p} \end{bmatrix}^T & , t \in [\alpha, \beta], \\ \begin{bmatrix} p_{x_c} & p_{v_c} \end{bmatrix}^T & , t \in]\beta, t_f], \end{cases}$ can be found using:

$$p = -\frac{\partial H}{\partial x}(x,p,u,t). \tag{5.48}$$

The variation of J_1 is

$$\delta J_1 = [p^T \delta x - H \delta t]_{t_0}^{t=\alpha} \tag{5.49}$$

$$= p_{x_c}\Big|_{t=\alpha^-} \delta x_{c\alpha} + p_{v_c}\Big|_{t=\alpha^-} \delta v_{c\alpha} - H\Big|_{t=\alpha^-} \delta\alpha - p_{x_c}\Big|_{t=t_0} \delta x_{c0} - p_{v_c}\Big|_{t=t_0} \delta v_{c0} + H\Big|_{t=t_0} \delta t_0. \tag{5.50}$$

The initial time t_0 is fixed and x_c, v_c are fully constrained at t_0, hence

$$\delta J_1 = p_{x_c}\Big|_{t=\alpha^-}\delta x_{c\alpha} + p_{v_c}\Big|_{t=\alpha^-}\delta v_{c\alpha} - H\Big|_{t=\alpha^-}\delta\alpha. \tag{5.51}$$

The first variation of J_2 is

$$\begin{aligned}\delta J_2 &= [p^T\delta x - H\delta t]_\alpha^\beta \qquad (5.52)\\ &= p_{x_c}\Big|_{t=\beta^-}\delta x_{c\beta} + p_{x_p}\Big|_{t=\beta^-}\delta x_{p_\beta} + p_{v_c}\Big|_{t=\beta^-}\delta v_{c\beta} + p_{v_p}\Big|_{t=\beta^-}\delta v_{p_\beta} - H\Big|_{t=\beta^-}\delta\beta\\ &\quad - p_{x_c}\Big|_{t=\alpha^+}\delta x_{c\alpha} - p_{x_p}\Big|_{t=\alpha^+}\delta x_{p_\alpha} - p_{v_c}\Big|_{t=\alpha^+}\delta v_{c\alpha} - p_{v_p}\Big|_{t=\alpha^+}\delta v_{p_\alpha} + H\Big|_{t=\alpha^+}\delta\alpha. \qquad (5.53)\end{aligned}$$

Recall the release constraint $x_c(\alpha) = x_p(\alpha)$, hence $\delta x_{c\alpha} = \delta x_{p_\alpha}$. Similarly, the retrieval constraints $x_c(\beta) = x_p(\beta)$ and $v_c(\beta) = v_p(\beta)$ imply $\delta x_{c\beta} = \delta x_{p_\beta}$ and $\delta v_{c\beta} = \delta v_{p_\beta}$. Thus, (5.52) can be rewritten as

$$\begin{aligned}\delta J_2 &= [p^T\delta x - H\delta t]_\alpha^\beta \qquad (5.54)\\ &= \left(p_{x_c} + p_{x_p}\right)\Big|_{t=\beta^-}\delta x_{c\beta} + \left(p_{v_c} + p_{v_p}\right)\Big|_{t=\beta^-}\delta v_{c\beta} - H\Big|_{t=\beta^-}\delta\beta\\ &\quad - \left(p_{x_c} + p_{x_p}\right)\Big|_{t=\alpha^+}\delta x_{c\alpha} - p_{v_c}\Big|_{t=\alpha^+}\delta v_{c\alpha} - p_{v_p}\Big|_{t=\alpha^+}\delta v_{p_\alpha} + H\Big|_{t=\alpha^+}\delta\alpha. \qquad (5.55)\end{aligned}$$

The first variation of J_3 is

$$\begin{aligned}\delta J_3 &= [p^T\delta x - H\delta t]_\beta^{t_f} \qquad (5.56)\\ &= p_{x_c}\Big|_{t=t_f}\delta x_{cf} + p_{v_c}\Big|_{t=t_f}\delta v_{cf} - H\Big|_{t=t_f}\delta t_f - p_{x_c}\Big|_{t=\beta^+}\delta x_{c\beta} - p_{v_c}\Big|_{t=\beta^+}\delta v_{c\beta} + H\Big|_{t=\beta^+}\delta\beta. \qquad (5.57)\end{aligned}$$

The first variation of K is

$$\delta K = \frac{\partial K}{\partial x_{c0}}\delta x_{c0} + \frac{\partial K}{\partial v_{c0}}\delta v_{c0} + \frac{\partial K}{\partial t_0}\delta t_0 + \frac{\partial K}{\partial x_{cf}}\delta x_{cf} + \frac{\partial K}{\partial v_{cf}}\delta v_{cf} + \frac{\partial K}{\partial t_f}\delta t_f. \tag{5.58}$$

The initial conditions t_0, x_{c0}, v_{c0} are fixed, hence

$$\delta K = \frac{\partial K}{\partial x_{cf}} \delta x_{cf} + \frac{\partial K}{\partial v_{cf}} \delta v_{cf} + \frac{\partial K}{\partial t_f} \delta t_f. \tag{5.59}$$

The transversality conditions are

$$\begin{aligned}
\delta J &= \delta K + \delta J_1 + \delta J_2 + \delta J_3 = 0 \\
&= \left(H\Big|_{t=\alpha^+} - H\Big|_{t=\alpha^-} \right) \delta\alpha + \left(p_{x_c}\Big|_{t=\alpha^-} - \left(p_{x_c} + p_{x_p}\right)\Big|_{t=\alpha^+} \right) \delta x_{c\alpha} + \left(p_{v_c}\Big|_{t=\alpha^-} - p_{v_c}\Big|_{t=\alpha^+} \right) \delta v_{c\alpha} \\
&+ \left(-p_{v_p}\Big|_{t=\alpha^+} \right) \delta v_{p_\alpha} + \left(H\Big|_{t=\beta^+} - H\Big|_{t=\beta^-} \right) \delta\beta + \left(\left(p_{x_c} + p_{x_p}\right)\Big|_{t=\beta^-} - p_{x_c}\Big|_{t=\beta^+} \right) \delta x_{c\beta} \\
&+ \left(\left(p_{v_c} + p_{v_p}\right)\Big|_{t=\beta^-} - p_{v_c}\Big|_{t=\beta^+} \right) \delta v_{c\beta} + \left(\frac{\partial K}{\partial t_f} - H\Big|_{t=t_f} \right) \delta t_f + \left(p_{x_c}\Big|_{t=t_f} + \frac{\partial K}{\partial x_{cf}} \right) \delta x_{cf} \\
&+ \left(p_{v_c}\Big|_{t=t_f} + \frac{\partial K}{\partial v_{cf}} \right) \delta v_{cf} = 0
\end{aligned} \tag{5.60}$$

for all admissible variations.

There are 10 transversality conditions in (5.60) and 10 free variables (α, $x_{c\alpha}$, $v_{c\alpha}$, v_{p_α}, β, $x_{c\beta}$, $v_{c\beta}$, t_f, x_{cf}, v_{cf}) for this problem; hence this system is determined.

5.1.5.5 Release and recovery on loci of points

The release occurs on a locus of points $l_\alpha(x_c, \alpha) = 0$ and pick-up occurs on a locus of points $l_\beta(x_c, \beta) = 0$. Using the results on target sets in [117], these conditions alter the transversality conditions at release and pick-up as follows:

$$p_{x_c}\Big|_{t=\alpha^-} - \left(p_{x_c} + p_{x_p}\right)\Big|_{t=\alpha^+} - \frac{\partial l_\alpha}{\partial x_{c\alpha}} \mu_\alpha = 0, \tag{5.61}$$

$$H\Big|_{t=\alpha^+} - H\Big|_{t=\alpha^-} - \frac{\partial l_\alpha}{\partial \alpha} \mu_\alpha = 0, \tag{5.62}$$

$$\left(p_{x_c} + p_{x_p}\right)\Big|_{t=\beta^-} - p_{x_c}\Big|_{t=\beta^+} - \frac{\partial l_\beta}{\partial x_{c\beta}} \mu_\beta = 0, \tag{5.63}$$

$$H\Big|_{t=\beta^+} - H\Big|_{t=\beta^-} - \frac{\partial l_\beta}{\partial \beta} \mu_\beta = 0, \tag{5.64}$$

where $\mu_\alpha \in \mathbb{R}$ and $\mu_\beta \in \mathbb{R}$.

The transversality conditions are

$$\begin{aligned}
\delta J &= \delta K + \delta J_1 + \delta J_2 + \delta J_3 = 0 \\
&= \left(H\Big|_{t=\alpha^+} - H\Big|_{t=\alpha^-} - \frac{\partial l_\alpha}{\partial \alpha}\mu_\alpha \right)\delta\alpha + \left(p_{x_c}\Big|_{t=\alpha^-} - \left(p_{x_c} + p_{x_p}\right)\Big|_{t=\alpha^+} - \frac{\partial l_\alpha}{\partial x_{c\alpha}}\mu_\alpha \right)\delta x_{c\alpha} \\
&+ \left(p_{v_c}\Big|_{t=\alpha^-} - p_{v_c}\Big|_{t=\alpha^+} \right)\delta v_{c\alpha} + \left(-p_{v_p}\Big|_{t=\alpha^+} \right)\delta v_{p_\alpha} + \left(H\Big|_{t=\beta^+} - H\Big|_{t=\beta^-} - \frac{\partial l_\beta}{\partial \beta}\mu_\beta \right)\delta\beta \\
&+ \left(\left(p_{x_c} + p_{x_p}\right)\Big|_{t=\beta^-} - p_{x_c}\Big|_{t=\beta^+} - \frac{\partial l_\beta}{\partial x_{c\beta}}\mu_\beta \right)\delta x_{c\beta} + \left(\left(p_{v_c} + p_{v_p}\right)\Big|_{t=\beta^-} - p_{v_c}\Big|_{t=\beta^+} \right)\delta v_{c\beta} \\
&+ \left(\frac{\partial K}{\partial t_f} - H\Big|_{t=t_f} \right)\delta t_f + \left(p_{x_c}\Big|_{t=t_f} + \frac{\partial K}{\partial x_{cf}} \right)\delta x_{cf} + \left(p_{v_c}\Big|_{t=t_f} + \frac{\partial K}{\partial v_{cf}} \right)\delta v_{cf} = 0 \qquad (5.65)
\end{aligned}$$

for all admissible variations and

$$l_\alpha(x_c, \alpha) = 0, \qquad (5.66)$$

$$l_\beta(x_c, \beta) = 0. \qquad (5.67)$$

There are 10 transversality conditions in (5.65), one release constraint in (5.66), one retrieval constraint in (5.67), and 12 free variables (α, $x_{c\alpha}$, $v_{c\alpha}$, v_{p_α}, β, $x_{c\beta}$, $v_{c\beta}$, t_f, x_{cf}, v_{cf}, μ_α, μ_β). Thus the system of equations is determined.

5.2 Pair of marsupial agents observing a stationary object

In this problem two marsupial agents are tasked with acquiring a given amount of information about a stationary object while minimizing their signal returned to an opponent's sensor in the area. An instance of such a problem is an intelligence gathering mission in an adversary's territory [83].

5.2.1 Model

The models for the agents and the opponent's sensor are given in this section.

5.2.1.1 Agents

The carrier and passenger agents travel in a plane at constant velocity, $v_c > 0$ and $v_p > 0$ respectively, and can turn instantaneously. The carrier agent is assumed to start at an initial location

$$\begin{bmatrix} x_c(0) \\ y_c(0) \end{bmatrix} = \begin{bmatrix} x_{c0} \\ y_{c0} \end{bmatrix}, \tag{5.68}$$

where x_c and y_c are the coordinates of the carrier agent along the two axes in the plane. The passenger agent is released at time α and

$$\begin{bmatrix} x_p(\alpha) \\ y_p(\alpha) \end{bmatrix} = \begin{bmatrix} x_c(\alpha) \\ y_c(\alpha) \end{bmatrix}, \tag{5.69}$$

where x_p and y_p are the coordinates of the passenger agent along the two axes in the plane. The dynamics for the carrier agent are

$$\begin{bmatrix} \dot{x}_c(t) \\ \dot{y}_c(t) \end{bmatrix} = \begin{bmatrix} v_c \cdot cos(u_c(t)) \\ v_c \cdot sin(u_c(t)) \end{bmatrix}, \tag{5.70}$$

where u_c is the control (heading) for the carrier agent. The dynamics for the passenger agent are

$$\begin{bmatrix} \dot{x}_p(t) \\ \dot{y}_p(t) \end{bmatrix} = \begin{bmatrix} v_p \cdot cos(u_p(t)) \\ v_p \cdot sin(u_p(t)) \end{bmatrix}, \tag{5.71}$$

where u_p is the control (heading) for the passenger agent. Both control variables are contained in $\vec{u}$.

The stationary object and opponent sensor are assumed to be colocated and without loss of generality placed at the origin of the plane. The distance of the carrier agent from the stationary object and opponent sensor is

$$r_c(t) = \sqrt{x_c(t)^2 + y_c(t)^2} \tag{5.72}$$

and the distance of the passenger agent from the stationary object and opponent sensor is

$$r_p(t) = \sqrt{x_p(t)^2 + y_p(t)^2}. \tag{5.73}$$

The stationary object has a constant reflectivity $k > 0$, the quality of the carrier agent's sensor is $w_c > 0$, and the quality of the passenger agent's sensor is $w_p > 0$. The agents use vision sensors, hence the signal to noise ratio for the sensors is proportional to $\frac{1}{r}$. The total information accumulated by the agents at time t is $I(t)$; the rate of information acquisition for the agents is

$$\dot{I}(t) = \begin{cases} \frac{w_c k}{r_c(t)} & , t \in [0, \alpha[, \\ \frac{w_c k}{r_c(t)} + \frac{w_p k}{r_p(t)} & , t \in [\alpha, t_f], \end{cases} \tag{5.74}$$

where t_f, the mission end time, is defined as follows

$$I(t_f) = I_f, \tag{5.75}$$

where I_f is the desired amount of information about the stationary object. Without loss of generality, the agents are assumed to start the mission with no information about the stationary object:

$$I(0) = 0. \tag{5.76}$$

5.2.1.2 Opponent sensor

The signal returned to the opponent's sensor by the carrier agent is

$$P_c(t) = \frac{\gamma s_c}{r_c(t)} \tag{5.77}$$

where $\gamma > 0$ is a constant accounting for the sensor properties and the constant $s_c > 0$ is the sensor signature of the carrier agent. Similarly, the signal returned to the opponent's sensor by the passenger agent is

$$P_p(t) = \frac{\gamma s_p}{r_p(t)} \tag{5.78}$$

where the constant $s_p > 0$ is the sensor signature of the passenger agent.

The accumulated signal returned to the opponent's sensor by both agents over the course of the mission is

$$J[\vec{x}] = \int_0^{\alpha} \gamma \frac{s_c}{r_c(t)} dt + \int_{\alpha}^{t_f} \gamma \left(\frac{s_c}{r_c(t)} + \frac{s_p}{r_p(t)} \right) dt, \tag{5.79}$$

where $\vec{x}$ is a vector containing all the agent states.

5.2.2 Problem formulation

The goal of the agents is to find a path that minimizes the accumulated signal returned to the opponent's sensor and obtains the desired amount of information about the stationary object. The problem can be formulated as follows: given (5.68), (5.69), (5.70), (5.71), (5.74), (5.76), find $x_c(t)$, $y_c(t)$, $x_p(t)$, $y_p(t)$, $0 \leq t \leq t_f$, such that (5.75) is satisfied and (5.79) is minimized.

5.2.3 Technical approach

The approach to solve the stated problem is now presented, in addition, characteristics of solutions are discussed. Necessary conditions for a solution to the problem are obtained using standard optimal control techniques, such as those found in [117] and [118].

5.2.3.1 Study of Hamiltonian and co-states

The costates for the problem are defined as follows:

$$\vec{p} = \begin{cases} \begin{bmatrix} p_{x_c}(t) & p_{y_c}(t) & p_I(t) \end{bmatrix}^T, t \in [0, \alpha[, \\ \begin{bmatrix} p_{x_c}(t) & p_{y_c}(t) & p_{x_p}(t) & p_{y_p}(t) & p_I(t) \end{bmatrix}^T, t \in [\alpha, t_f]. \end{cases} \tag{5.80}$$

The Hamiltonian is:

$$H(\vec{x}, \vec{p}, \vec{u}) = \begin{cases} p_{x_c}(t) \cdot v_c \cos(u_c(t)) + p_{y_c}(t) \cdot v_c \sin(u_c(t)) + p_I(t) \cdot \frac{w_c k}{r_c(t)} - \frac{s_c \gamma}{r_c(t)}, \, t \in [0, \alpha[, \\ p_{x_c}(t) \cdot v_c \cos(u_c(t)) + p_{y_c}(t) \cdot v_c \sin(u_c(t)) + p_{x_p}(t) \cdot v_p \cos(u_p(t)) + p_{y_p}(t) \cdot v_p \sin(u_p(t)) \\ \qquad + p_I(t) \cdot k \cdot \left(\frac{w_c}{r_c(t)} + \frac{w_p}{r_p(t)} \right) - \gamma \cdot \left(\frac{s_c}{r_c(t)} + \frac{s_p}{r_p(t)} \right), \, t \in [\alpha, t_f]. \end{cases} \tag{5.81}$$

The costate dynamics are derived from the Hamiltonian as follows:

$$\dot{p_I}(t) = -\frac{\partial H}{\partial I} = 0, \forall t, \tag{5.82}$$

$$\dot{p_{x_c}}(t) = -\frac{\partial H}{\partial x_c} = \frac{(s_c \gamma - p_I w_c k) x_c}{r_c^3(t)}, \forall t, \tag{5.83}$$

$$\dot{p_{y_c}}(t) = -\frac{\partial H}{\partial y_c} = \frac{(s_c \gamma - p_I w_c k) y_c}{r_c^3(t)}, \forall t, \tag{5.84}$$

$$\dot{p_{x_p}}(t) = -\frac{\partial H}{\partial x_p} = \frac{(s_p \gamma - p_I w_p k) x_p}{r_p^3(t)}, t \in [\alpha, t_f], \tag{5.85}$$

$$\dot{p_{y_p}}(t) = -\frac{\partial H}{\partial y_p} = \frac{(s_p \gamma - p_I w_p k) y_p}{r_p^3(t)}, t \in [\alpha, t_f]. \tag{5.86}$$

5.2.3.2 Transversality conditions

The transversality conditions for a solution to the problem are

$$\begin{aligned}\delta J = &\left(H(t)\Big|_{t=\alpha^+} - H(t)\Big|_{t=\alpha^-}\right)\delta\alpha + \left(H(t)\Big|_{t=t_f}\right)\delta t_f + \left(p_I(t)\Big|_{t=\alpha^-} - p_I(t)\Big|_{t=\alpha^+}\right)\delta I_\alpha \\ &+ \left(p_{x_c}(t)\Big|_{t=\alpha^-} - \left(p_{x_c}(t) + p_{x_p}(t)\right)\Big|_{t=\alpha^+}\right)\delta x_{c\alpha} + \left(p_{y_c}(t)\Big|_{t=\alpha^-} - \left(p_{y_c}(t) + p_{y_p}(t)\right)\Big|_{t=\alpha^+}\right)\delta y_{c\alpha} \\ &+ \left(p_{x_c}(t)\Big|_{t=t_f}\right)\delta x_{cf} + \left(p_{y_c}(t)\Big|_{t=t_f}\right)\delta y_{cf} + \left(p_{x_p}(t)\Big|_{t=t_f}\right)\delta x_{pf} + \left(p_{y_p}(t)\right)\Big|_{t=t_f}\delta y_{pf} = 0 \quad (5.87)\end{aligned}$$

for all admissible variations.

The nine transversality conditions in (5.87) and the termination condition (5.75) are used to compute the ten free constants in the problem: α, $I(\alpha)$, x_{c_α}, y_{c_α}, t_f, $I(t_f)$, x_{c_f}, y_{c_f}, x_{p_f}, y_{p_f} where x_{c_f}, y_{c_f}, x_{p_f}, y_{p_f} are the final positions of the carrier and the passenger agent.

From the $p_I(t)$ costate dynamics, we see that $p_I(t)$ is constant before release and after release. The transversality condition for δI_α, $p_I(t)\Big|_{t=\alpha^-} = p_I(t)\Big|_{t=\alpha^+}$, indicates that $p_I(t)$ is constant for the entire mission, i.e., $p_I(t) = p_I, \forall t$. p_I can be computed using the transversality conditions, starting with the transversality condition for δt_f,

$$0 = H(t)\Big|_{t=t_f} \quad (5.88)$$

$$\begin{aligned}0 = &p_{x_c}(t)\cdot v_c\cos(u_c(t))\Big|_{t=t_f} + p_{y_c}(t)\cdot v_c\sin(u_c(t))\Big|_{t=t_f} + p_{x_p}(t)\cdot v_p\cos(u_p(t))\Big|_{t=t_f} \\ &+ p_{y_p}(t)\cdot v_p\sin(u_p(t))\Big|_{t=t_f} + p_I(t)\cdot k\cdot\left(\frac{w_c}{r_c(t)} + \frac{w_p}{r_p(t)}\right)\Big|_{t=t_f} - \gamma\cdot\left(\frac{s_c}{r_c(t)} + \frac{s_p}{r_p(t)}\Big|_{t=t_f}\right). \quad (5.89)\end{aligned}$$

Using the transversality conditions for δx_{cf}, δy_{cf}, δx_{pf}, and δy_{pf}, (5.88) can be reduced to

$$p_I = \frac{\gamma\left(\frac{s_c}{r_{cf}} + \frac{s_p}{r_{pf}}\right)}{k\left(\frac{w_c}{r_{cf}} + \frac{w_p}{r_{pf}}\right)}, \quad (5.90)$$

where r_{cf} and r_{pf} are the distances from the carrier agent and passenger agent to the object of interest at the end of the mission.

5.2.3.3 Extremals

The optimal control $\vec{u}$ can be found using Pontryagin's maximum principle:

$$\vec{u} = \underset{\vec{v}}{argmax}(H(\vec{x}, \vec{p}, \vec{v}, t)). \tag{5.91}$$

Thus, the following equations need to be satisfied for $u_c(t)$ and $u_p(t)$:

$$p_{x_c}(t) \cdot sin(u_c(t)) = p_{y_c}(t) \cdot cos(u_c(t)), \tag{5.92}$$

$$p_{x_p}(t) \cdot sin(u_p(t)) = p_{y_p}(t) \cdot cos(u_p(t)). \tag{5.93}$$

The control is singular for $u_c(t)$ when $p_{x_c}(t) = 0$, $\dot{p}_{x_c}(t) = 0$, $p_{y_c}(t) = 0$, $\dot{p}_{y_c}(t) = 0$ which can only occur at the origin. Similarly, the control $u_p(t)$ is singular when $p_{x_p}(t) = 0$, $\dot{p}_{x_p}(t) = 0$,$p_{y_p} = 0$, $\dot{p}_{y_p}(t) = 0$ which can also only occur at the origin.

The optimal control $u_c(t)$ is thus

$$u_c(t) = \begin{cases} \frac{\pi}{2}, -\frac{\pi}{2} & , p_{x_c}(t) = 0 \wedge p_{y_c}(t) \neq 0, \\ 0, \pi & , p_{x_c}(t) \neq 0 \wedge p_{y_c}(t) = 0, \\ \tan^{-1}\left(\frac{p_{y_c}(t)}{p_{x_c}(t)}\right) & , p_{x_c}(t) \neq 0 \wedge p_{y_c}(t) \neq 0. \end{cases} \tag{5.94}$$

Similarly, the optimal control $u_p(t)$ is

$$u_p(t) = \begin{cases} \frac{\pi}{2}, -\frac{\pi}{2} & , p_{x_p}(t) = 0 \wedge p_{y_p}(t) \neq 0, \\ 0, \pi & , p_{x_p}(t) \neq 0 \wedge p_{y_p}(t) = 0, \\ \tan^{-1}\left(\frac{p_{y_p}(t)}{p_{x_p}(t)}\right) & , p_{x_p}(t) \neq 0 \wedge p_{y_p}(t) \neq 0. \end{cases} \tag{5.95}$$

5.2.3.4 Rectilinear motion

In simulations, the optimal paths traveled by the carrier and passenger agents are on a line. In this subsection, we investigate the conditions under which linear motion occurs. For the carrier agent or the passenger agent to travel on a line, the respective inputs to their dynamics need to be constant, i.e., $\dot{u}_c(t) = 0$ or $\dot{u}_p(t) = 0$ respectively where $t \in [0, t_f]$. Using the results from Pontryagin's maximum principle in the previous section:

$$u(t) = \tan^{-1}\left(\frac{p_y(t)}{p_x(t)}\right), \tag{5.96}$$

$$\frac{d}{dt}\left(\frac{p_y(t)}{p_x(t)}\right) = 0 \Rightarrow \dot{u}(t) = 0. \tag{5.97}$$

We now show that the necessary conditions for optimality imply linear motion. In the following development, the c and p subscripts indicating carrier and passenger respectively are dropped as the equations hold for both.

Lemma 5.2.1. $\exists \tau$ *s.t.* $\frac{y(\tau)}{x(\tau)} = \frac{p_y(\tau)}{p_x(\tau)} \Rightarrow \frac{d}{dt}\left(\frac{y(t)}{x(t)}\right) = 0 \wedge \frac{d}{dt}\left(\frac{p_y(t)}{p_x(t)}\right) = 0, t \in [0, t_f]$.

Proof. The time derivative of the ratio of states at time τ is zero:

$$\frac{d}{dt}\left(\frac{y(\tau)}{x(\tau)}\right) = \frac{\dot{x}(\tau)y(\tau) - \dot{y}(\tau)x(\tau)}{x^2(\tau)} \tag{5.98}$$

$$= \frac{v \cdot \left(\cos\left(\tan^{-1}\left(\frac{p_y(\tau)}{p_x(\tau)}\right)\right) \cdot y(\tau) - \sin\left(\tan^{-1}\left(\frac{p_y(\tau)}{p_x(\tau)}\right)\right) \cdot x(\tau)\right)}{x^2(\tau)} \tag{5.99}$$

$$= \frac{v \cdot \cos\left(\tan^{-1}\left(\frac{p_y(\tau)}{p_x(\tau)}\right)\right)}{x^2(\tau)} \cdot \left(y(\tau) - \frac{p_y(\tau)}{p_x(\tau)} x(\tau)\right) \tag{5.100}$$

$$= \frac{v \cdot \cos\left(\tan^{-1}\left(\frac{p_y(\tau)}{p_x(\tau)}\right)\right)}{x^2(\tau)} \cdot \left(y(\tau) - \frac{y(\tau)}{x(\tau)} x(\tau)\right) \tag{5.101}$$

$$= 0. \tag{5.102}$$

The time derivative of the ratio of the costates at time τ is zero:

$$\frac{d}{dt}\left(\frac{p_y(\tau)}{p_x(\tau)}\right) = \frac{\dot{p}_x(\tau)p_y(\tau) - \dot{p}_y(\tau)p_x(\tau)}{p_x^2(\tau)} \tag{5.103}$$

$$= \frac{(s\gamma - p_I wk)}{r^3(\tau)p_x^2(\tau)} \cdot \Big(x(\tau)p_y(\tau) - y(\tau)p_x(\tau)\Big) \tag{5.104}$$

$$= \frac{(s\gamma - p_I wk)\cdot x(\tau)}{r^3(\tau)p_x^2(\tau)} \cdot \left(p_y(\tau) - \frac{y(\tau)}{x(\tau)}p_x(\tau)\right) \tag{5.105}$$

$$= 0. \tag{5.106}$$

Then,

$$\frac{d}{dt}\left(\frac{y(\tau)}{x(\tau)}\right) = 0 \Rightarrow \frac{y(\tau^-)}{x(\tau^-)} = \frac{y(\tau)}{x(\tau)} = \frac{y(\tau^+)}{x(\tau^+)} \tag{5.107}$$

and

$$\frac{d}{dt}\left(\frac{p_y(\tau)}{p_x(\tau)}\right) = 0 \Rightarrow \frac{p_y(\tau^-)}{p_x(\tau^-)} = \frac{p_y(\tau)}{p_x(\tau)} = \frac{p_y(\tau^+)}{p_x(\tau^+)}. \tag{5.108}$$

The results from time τ can be applied to times τ^- and τ^+; continuing this development until the time bounds are reached shows that

$$\frac{y(t)}{x(t)} = \frac{p_y(t)}{p_x(t)} = c_1,\ c_1 \in \mathbb{R},\ t \in [0, t_f]. \tag{5.109}$$

□

We now show that such a τ exists in this problem.

Lemma 5.2.2. *For optimal solutions,* $\exists\tau$ *s.t.* $\frac{y(\tau)}{x(\tau)} = \frac{p_y(\tau)}{p_x(\tau)}$.

Proof. The costates p_x and p_y at the instant before the completion of the mission are studied:

$$p_x(t)\Big|_{t=t_f} = p_x(t)\Big|_{t=t_f^-} + \dot{p}_x(t)\Big|_{t=t_f^-} \cdot \epsilon, \tag{5.110}$$

where ϵ is an infinitesimal increment in time. Recall from the transversality conditions (5.87) that $p_x(t)\Big|_{t=t_f} = 0$, hence

$$p_x(t)\Big|_{t=t_f^-} = -\frac{(s\gamma - p_I wk)\cdot x(t_f^-)}{r^3(t_f^-)} \cdot \epsilon. \tag{5.111}$$

Similarly,

$$\begin{aligned} p_y(t)\Big|_{t=t_f} &= p_y(t)\Big|_{t=t_f^-} + \dot{p}_y(t)\Big|_{t=t_f^-} \cdot \epsilon & (5.112)\\ &= -\frac{(s\gamma - p_I wk)\cdot y(t_f^-)}{r^3(t_f^-)} \cdot \epsilon. & (5.113)\end{aligned}$$

Thus,

$$\frac{p_y(t)}{p_x(t)}\Big|_{t=t_f^-} = \frac{y(t_f^-)}{x(t_f^-)}. \tag{5.114}$$

□

Theorem 5.2.3. *The necessary conditions for optimality imply rectilinear motion for both agents along a line passing through the origin.*

Proof. Lemmas 5.2.2 and 5.2.1 ⇒ $\frac{d}{dt}\left(\frac{y(t)}{x(t)}\right) = 0$, $t \in [0, t_f]$ ⇒

$$\frac{y(t)}{x(t)} = \frac{y(\alpha)}{x(\alpha)} = \frac{y_c(\alpha)}{x_c(\alpha)} = \frac{y_c(0)}{x_c(0)} = \frac{y_{c_0}}{x_{c_0}},\ t \in [0, t_f]. \tag{5.115}$$

□

It is important to note that while Theorem 5.2.3 states that the agents travel along a

Table 5.1: Simulation parameters.

Scenario	(x_{c_0}, y_{c_0})	γ	k	I_f	$\frac{v_c}{v_p}$	$\frac{w_c}{w_p}$	$\frac{s_c}{s_p}$
A1	(-5,20)	1	1	0.35	2	0.5	2
A2	(-40,10)	1	1	0.35	2	0.5	2
A3	(-60,50)	1	1	0.35	2	0.5	2
A4	(-25,30)	1	1	0.35	2	0.5	2
B1	(-5,20)	1	1	0.35	2	2	2
B2	(-40,10)	1	1	0.35	2	2	2
B3	(-60,50)	1	1	0.35	2	2	2
B4	(-25,30)	1	1	0.35	2	2	2
C1	(-5,20)	1	1	0.35	0.5	0.5	2
C2	(-40,10)	1	1	0.35	0.5	0.5	2
C3	(-60,50)	1	1	0.35	0.5	0.5	2
C4	(-25,30)	1	1	0.35	0.5	0.5	2

straight line, it does not describe how the agents move along that line.

5.2.4 Simulations

In this section, several simulations illustrating the results from the previous section are shown. The parameters for the simulations discussed in this section are shown in Table 5.1; the scenario letter dictates the agent and opponent sensor parameters and the number after the letter dictates the initial condition for the carrier agent. Figures 5.1-5.3 illustrate the optimal paths for the agents, the release point, and the stationary object location at the origin; the arrows indicate the agent's position and direction at the end of the mission. The optimal paths are computed by generating a number of candidate solutions, running MATLAB's fmincon function for each of these candidates, and storing the best result.

In scenario A, the carrier agent is faster and has a larger signature than the passenger agent but the passenger agent possesses better sensors. The paths of the carrier and passenger agents for scenario A are illustrated in Figure 5.1 for multiple initial locations. For all initial locations, the carrier agent immediately deploys the passenger agent. The carrier agent moves away from the stationary object while the passenger agent moves towards the stationary object. The distances traveled by the agents vary with differing initial carrier agent location.

In scenario B, the carrier agent is faster, possesses better sensors, and returns a larger signal to the opponent's sensor than the passenger agent. The agents' paths for the different initial conditions are shown in Figure 5.2. For all problem instances, the passenger agent is not deployed and the carrier agent moves away from the object of interest.

In scenario C, the carrier agent is slower than the passenger agent while the passenger agent carries sensors of a higher quality and has a smaller radar signature. The passenger agent is released immediately and moves towards the object of interest while the carrier agent moves away from the object of interest. The deployment and direction of the agents' motion is the same as in scenario A. The only difference between scenarios A and C is the velocity ratio of the agents; this simulation suggests that the velocity of the agents does not affect the deployment and direction of the vehicles.

These simulations confirm rectilinear motion for both agents as predicted in Theorem 5.2.3. In addition, the passenger is not always released, but when release occurs, $\alpha = 0$. The fact that the passenger agent is deployed for all initial conditions in scenarios A and C but never deployed in scenario B suggests that the deployment of the passenger depends on the configuration of the agents and not the initial position of the carrier agent.

5.3 Summary

In this chapter, optimal control for a pair of marsupial agents performing monitoring tasks was studied and necessary conditions for optimal paths of the two marsupial agents were provided. The cases treated accounted for constrained initial positions for the carrier agent, constrained final positions for both agents, termination conditions for the agents, constrained release of the passenger agent, recovery of the passenger agent, or constrained release and recovery of the passenger agent.

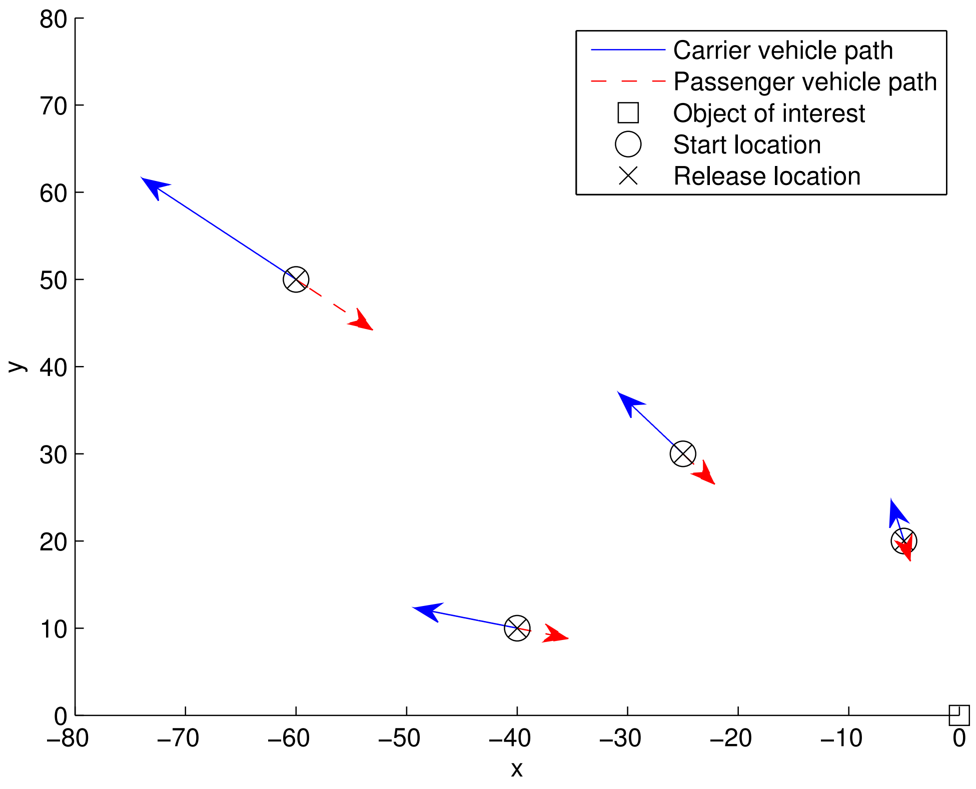


Figure 5.1: Carrier and passenger optimal paths for different initial conditions in scenario A.

A path planning problem for a pair of marsupial agents acquiring information about a stationary object while avoiding detection is also considered. A constrained optimization problem is formulated and the necessary conditions for optimal solutions to this problem are provided. We demonstrate that to satisfy the necessary conditions for optimality, each agent's motion must be linear (along a line passing through the stationary object). This result is illustrated in several simulations.

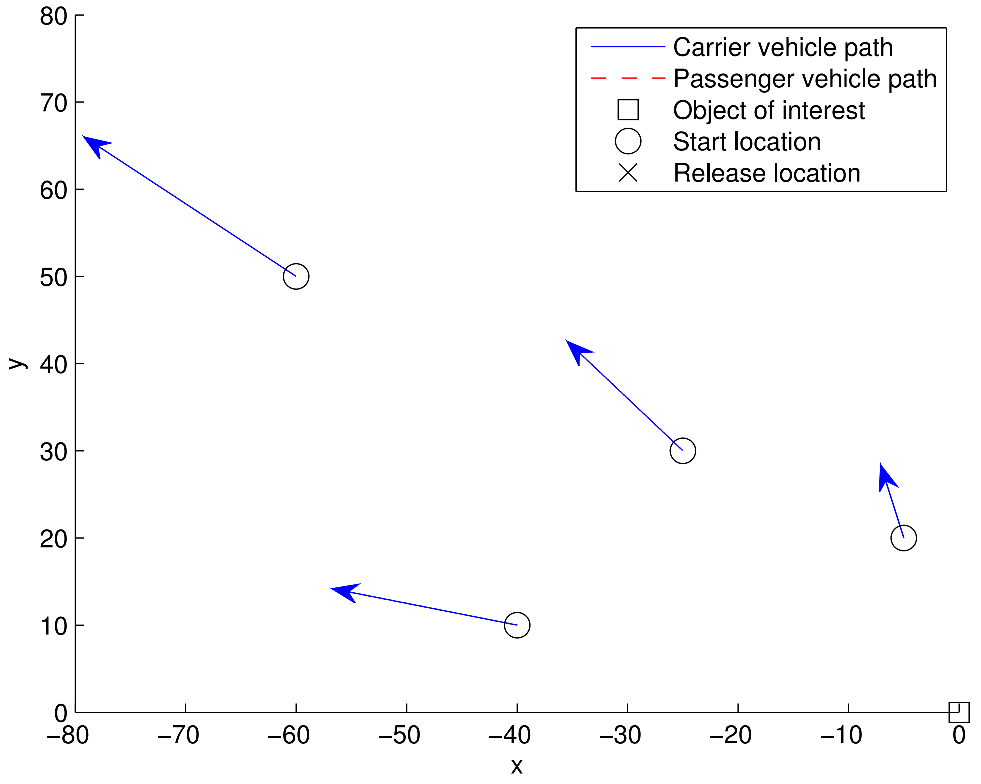


Figure 5.2: Carrier and passenger optimal paths for different initial conditions in scenario B.

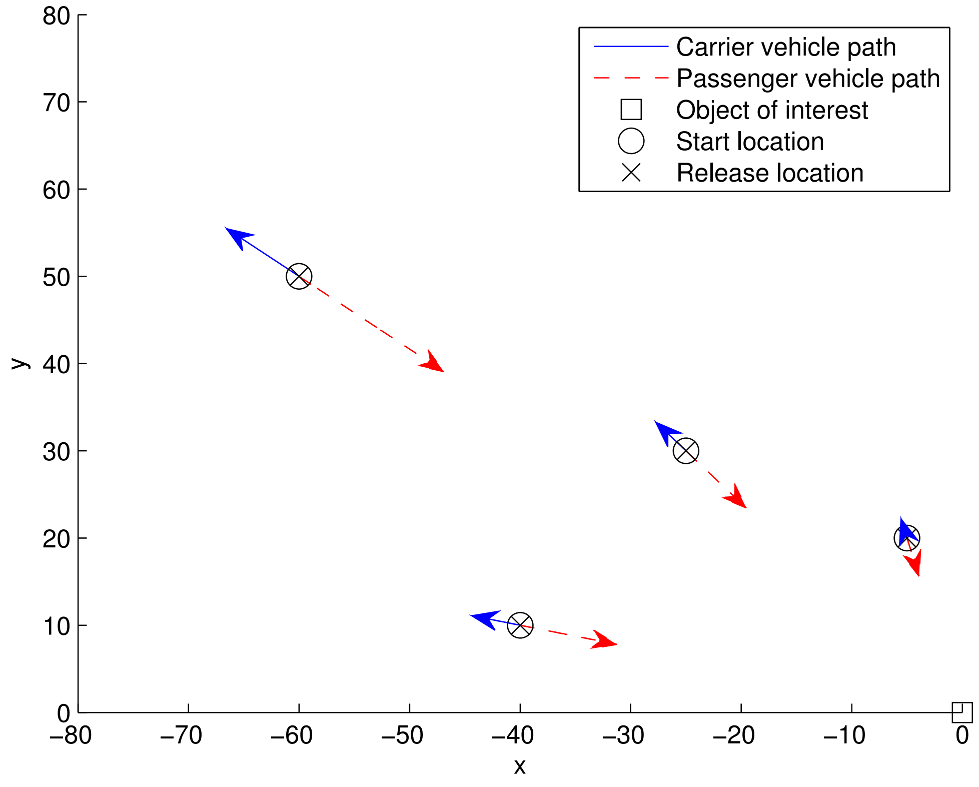


Figure 5.3: Carrier and passenger optimal paths for different initial conditions in scenario C.

CHAPTER 6

Conclusions

Autonomous agents, both mobile and stationary, are used for a widespread set of tasks and their usage is only to increase in the future. With their increased usage, the tasks to which they are assigned are more complex. Because the tasks are complex, cooperation between autonomous agents is often required. In addition, cooperation may need to occur between autonomous agents of different types. This dissertation studies how monitoring tasks, e.g., collecting information or pursuing objects of interest in an area, can be accomplished using multiple heterogeneous autonomous agents.

6.1 Summary

The dissertation treats the problem differently depending on the types of agents and objects in the monitoring task. Mobile, stationary, and marsupial agents as well as mobile and stationary objects are considered. In Chapter 1, the work is motivated and a general problem statement for agents monitoring objects is given. Literature relevant to path planning for autonomous vehicles, optimal sensor configurations, and marsupial vehicle operations is reviewed in Chapter 2. The main body of the dissertation is divided into three chapters, each treating a different type of agent: mobile, stationary, and marsupial.

In Chapter 3, the first set of problems is treated; they involve a mobile agent and multiple stationary objects that need to be revisited. Almost periodic paths that solve the problems are demonstrated to exist. Heuristics to compute paths for a mobile agent without fuel concerns are given and an algorithm that minimizes refueling costs when fuel is accounted for is provided. A pursuit problem involving mobile agents and stationary agents pursuing a mobile object is then treated. The problem is shown to be NP-hard and a heuristic to compute paths for the mobile agents that meet revisit deadlines to the stationary agents and pursue the object is provided.

In Chapter 4 the pursuit problem is further studied, methods to optimally place the stationary agents in this problem are provided and their effectiveness compared. These methods are the minimization of the probability of misclassification, the penalty incurred by the placement, and the maximization of Fisher information. A method to select revisit deadlines for the stationary agents using the model of the mobile object's motion is also given; the method selects the revisit deadline of a stationary agent by matching the mobile agent's recurrence time to the same object. Methods to fuse the measurements from colocated stationary agents are also presented.

In Chapter 5, problems involving a pair of marsupial agents, one carrier and one passenger, are studied and necessary conditions for optimal paths under a variety of constraints are given. A problem where two marsupial agents gather information about an object while avoiding detection is then formulated, necessary conditions for optimal paths for the agents are provided, and the optimal paths of the agents are demonstrated to be rectilinear.

6.2 Concluding remarks

There exist numerous applications for autonomous agents performing monitoring tasks. These different applications share key aspects which can be leveraged to formulate similar problems; in turn these similar problems are used to generate common solutions. The contents of this dissertation can be used to plan the paths and configurations for the agents performing certain tasks. The original contribution of proving revisit deadlines constraints result in the existence of almost periodic paths can be leveraged to find solutions faster by safely reducing the search space. The algorithm provided to compute the minimal fuel cost path is one instance of leveraging periodicity. The original contribution of proving most of the problems treated are intractable validates the use of heuristics.

The contributions regarding methods to place stationary agents, select the revisit deadlines of stationary agents, and plan the mobile agents' paths for the pursuit task can be used to best allocate the agents before and during the tracking task. The necessary conditions for optimal paths of marsupial agents can be utilized to find optimal solutions or reject sub-optimal solutions faster. In addition, the proof that rectilinear motion is optimal for marsupial agents collecting information while avoid detection in the case studied further reduces the search space and accelerates the computation of solutions.

While this dissertation does not treat every known monitoring problem using heterogeneous agents, it does provide methods to compute solutions or compute solutions faster for persistent visitation problems, pursuit problems using stationary and mobile agents, and monitoring problems using a pair of marsupial agents.

6.3 Future directions

Several future directions exist for the work presented in this dissertation, they are listed below;

- **Alternative heuristics:** Investigate different heuristics for the selection of paths for the pursuit of a mobile object using mobile agents and stationary agents. These heuristics could also use the time spent in flight to pre-compute future paths for the agents or anticipate potential mobile object paths; these paths could then be used for an accelerated selection of the optimal action once new information is obtained.

- **Degraded communications:** Investigate methods to handle degraded communications between agents in Chapters 3 and 5. For the pursuit problem with multiple mobile agents treated in Section 3.1, one way to handle degraded communications between the mobile agents and/or the central authority is to partition the area and assign agents to each area. Short communications regarding detections or the lack thereof could be transmitted by leaving messages at stationary agents on the borders of the partitions.

- **Multiple carrier and passenger marsupial operations:** Study optimal paths when multiple carriers and passengers are involved. The carriers may not necessarily all carry the same number or type of passengers.

- **Discrete marsupial operations:** Investigate marsupial operations in discrete spaces such as graphs and compare to the results obtained in the pursuit problem with mobile and stationary agents.

- **Nested marsupial operations:** Investigate optimal paths when marsupial agents are nested, i.e., a passenger agent may also be a carrier agent. To treat this problem, a formulation and theory different than that presented in Chapter 5 is likely needed. In addition, complexity problems may arise when considering many marsupial agents and large levels of nesting.

BIBLIOGRAPHY

[1] J. Las Fargeas, B. Hyun, P. Kabamba, and A. Girard. Persistent visitation with fuel constraints. *Procedia - Social and Behavioral Sciences*, 54:1037–1046, 2012. Proceedings of the 15th Meeting of the Euro Working Group on Transportation.

[2] J. Las Fargeas, B. Hyun, P. Kabamba, and A. Girard. Persistent visitation under revisit constraints. In *Proceedings of the International Conference on Unmanned Aircraft Systems*, pages 952–957, 2013.

[3] J. Las Fargeas, P. Kabamba, and A. Girard. Cooperative surveillance and pursuit using unmanned aerial vehicles and unattended ground sensors. *Sensors*, 15(1):1365–1388, 2015.

[4] J. Las Fargeas, P. Kabamba, and A. Girard. Optimal configuration of alarm sensors for monitoring mobile ergodic markov phenomena on arbitrary graphs. *IEEE Sensors Journal*, 2015. Accepted for publication.

[5] J. Las Fargeas, P. Kabamba, and A. Girard. Path planning for information acquisition and evasion using marsupial vehicles. In *Proceedings of the American Control Conference*, 2015. Accepted.

[6] J. Las Fargeas and A. Girard. Optimal paths for marsupial systems with one carrier agent and one passenger agent. *IEEE Transactions on Robotics*. In preparation.

[7] P.F. Hokayem, D. Stipanovic, and M.W. Spong. On persistent coverage control. In *Proceedings of the 46th IEEE Conference on Decision and Control*, pages 6130–6135, December 2007.

[8] N. Nigam and I. Kroo. Persistent surveillance using multiple unmanned air vehicles. In *Proceedings of the IEEE Aerospace Conference*, pages 1–14, 2008.

[9] N. Nigam, S. Bieniawski, I. Kroo, and J. Vian. Control of multiple uavs for persistent surveillance: Algorithm and flight test results. *IEEE Transactions on Control Systems Technology*, 20(99):1–17, 2011.

[10] 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*, pages 385–390, April 2007.

[11] Y. Elmaliach, A. Shiloni, and G.A. Kaminka. A realistic model of frequency-based multi-robot fence patrolling. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multi-Agent Systems*, volume 1, pages 63–70, 2008.

[12] S.L. Smith, M. Schwager, and D. Rus. Persistent monitoring of changing environments using a robot with limited range sensing. In *Proceedings of the IEEE International Conference Robotics and Automation*, pages 5448–5455, May 2011.

[13] C.G. Cassandras, X.C. Ding, and X. Lin. An optimal control approach for the persistent monitoring problem. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, pages 2907–2912, December 2011.

[14] A. Nasir, E. Atkins, and I. V. Kolmanovsky. Science optimal spacecraft attitude maneuvering while accounting for failure mode. In *Proceedings of the 18th IFAC World Conference*, pages 812–817, August-September 2011.

[15] N. Ozay, Ufuk Topcu, R.M. Murray, and T. Wongpiromsarn. Distributed synthesis of control protocols for smart camera networks. In *IEEE/ACM International Conference on Cyber-Physical Systems*, pages 45–54, 2011.

[16] N. Christofides. The vehicle routing problem. *Revue Française d'Automatique, Informatique, et Recherche Opérationelle*, 10(2):55–70, 1976.

[17] N. Christofides and J.E. Beasley. The period routing problem. *Networks*, 14:237–256, 1984.

[18] M. Desrochers, J.K. Lenstra, M.W.P. Savelsbergh, and F. Soumis. Vehicle routing with time windows: Optimization and approximation. In *Vehicle Routing: Methods and Studies*, pages 65–84. Elsevier Science Publishers, Amsterdam, 1988.

[19] J-F. Cordeau, G. Laporte, and A. Mercier. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Soc.*, (52):928–936, 2001.

[20] Anjan Chakrabarty and Jack W Langelaan. Energy-based long-range path planning for soaring-capable unmanned aerial vehicles. *Journal of Guidance, Control, and Dynamics*, 34(4):1002–1015, 2011.

[21] M. Gendreau, G. Laporte, and F. Semet. The covering tour problem. *Operations Research*, 45(4):568–576, July/August 1997.

[22] C. Groër, B. Golden, and E. Wasil. The consistent vehicle routing problem. *Manufacturing & Service Operations Management*, 11(4):630–643, 2009.

[23] M. Figliozzi. Vehicle routing problem for emissions minimization. *Transportation Research Record: Journal of the Transportation Research Board*, 2197(1):1–7, 2010.

[24] P. Oberlin, S. Rathinam, and S. Darbha. Today's traveling salesman problem. *IEEE Robotics & Automation Magazine*, 17(4):70–77, 2010.

[25] I. Chao, B.L. Golden, and E.A. Wasil. A new heuristic for the period traveling salesman problem. *Computers & Operations Research*, 22(5):553–565, 1995.

[26] K. Savla, F. Bullo, and E. Frazzoli. On traveling salesperson problem for dubins' vehicle: Stochastic and dynamics environments. In *Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference*, pages 4530–4535, December 2005.

[27] R. Wolfler Calvo and R. Cordone. A heuristic approach to the overnight security service problem. *Computers & Operations Research*, 30(9):1269–1287, 2003.

[28] Y. Chevaleyre. Theoretical analysis of the multi-agent patrolling problem. In *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 302–308, September 2004.

[29] V. Huynh, J.J. Enright, and E. Frazzoli. Persistent patrol with limited-range on-board sensors. In *Proceedings of the 49th IEEE Conference on Decision and Control*, pages 7661–7668, December 2010.

[30] F. Pasqualetti, J.W. Durham, and F. Bullo. Cooperative patrolling via weighted tours: Performance analysis and distributed algorithms. *IEEE Transactions on Robotics*, 28(5):1181–1188, 2012.

[31] D. Portugal and R. Rocha. A survey on multi-robot patrolling algorithms. In *Proceedings of the 3rd Doctoral Conference on Computing, Electrical, and Industrial Systems*, number 349, pages 139–146, 2011.

[32] N. Basilico, N. Gatti, and F. Amigoni. Developing a deterministic patrolling strategy for security agents. In *Proceedings of the IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technologies*, volume 2, pages 565–572, September 2009.

[33] N. Basilico, N. Gatti, and F. Villa. Asynchronous multi-robot patrolling against intrusions in arbitrary topologies. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 1224–1229, 2010.

[34] K. Barton and D. Kingston. Systematic surveillance for uavs: A feedforward iterative learning control approach. In *Proceedings of the American Control Conference*, pages 5917–5922, Jun. 2013.

[35] R.W. Beard, T.W. McLain, M.A. Goodrich, and E.P. Anderson. Coordinated target assignment and intercept for unmanned air vehicles. *IEEE Transactions on Robotics and Automation*, 18(6):911–922, Dec 2002.

[36] U. Zengin and A. Dogan. Real-time target tracking for autonomous uavs in adversarial environments: A gradient search algorithm. *IEEE Transactions on Robotics*, 23(2):294–307, April 2007.

[37] E. Frazzoli, M. A. Dahleh, and E. Feron. Real-time motion planning for agile autonomous vehicles. *Journal of Guidance, Control, and Dynamics*, 25(1):116–129, 2002.

[38] D. Kingston. Intruder tracking using uav teams and ground sensor networks. In *Proceedings of the German Aerospace Congress*. German Society for Aeronautics and Astronautics, September 2012.

[39] D. W. Casbeer, K. Krishnamoorthy, A. Eggert, P. Chandler, and M. Pachter. Optimal search for a random moving intruder. In *Proceedings of the AIAA Infotech@Aerospace Conference*, Jun. 2012.

[40] K. Krishnamoorthy, D. Casbeer, P. Chandler, M. Pachter, and S. Darbha. UAV search and capture of a moving ground target under delayed information. In *Proceedings of the 51st IEEE Conference on Decision and Control*, pages 3092–3097, Dec. 2012.

[41] D. W. Casbeer, K. Meier, and Y. Cao. Estimating the state of an intruder with a uav and unattended ground sensors. In *Proceedings of the AIAA Infotech@Aerospace Conference: Guidance, Navigation, and Control and Co-located Conferences*, pages 300–318, 2013.

[42] K. Krishnamoorthy, S. Darbha, P.P. Khargonekar, D. Casbeer, P. Chandler, and M. Pachter. Optimal minimax pursuit evasion on a manhattan grid. In *Proceedings of the American Control Conference*, pages 3421–3428, Jun. 2013.

[43] A. T. Klesh and P. T. Kabamba. Solar-powered aircraft: Energy-optimal path planning and perpetual endurance. *Journal of Guidance, Control, and Dynamics*, 32(4):1320–1329, 2009.

[44] P. Niedfeldt, D. Kingston, and R. Beard. Vehicle state estimation within a road network using a bayesian filter. In *Proceedings of the American Control Conference*, pages 4910–4915, June 2011.

[45] T. H. Summers, M. R. Akella, and M. J. Mears. Coordinated standoff tracking of moving targets: Control laws and information architectures. *Journal of Guidance, Control, and Dynamics*, 32(1):56–69, 2009.

[46] N. Slegers, J. Kyle, and M. Costello. Nonlinear model predictive control technique for unmanned air vehicles. *Journal of Guidance, Control, and Dynamics*, 29(5):1179–1188, 2006.

[47] M. Stachura and E. W. Frew. Cooperative target localization with a communication-aware unmanned aircraft system. *Journal of Guidance, Control, and Dynamics*, 34(5):1352–1362, 2011.

[48] A. Heydari and N. Balakrishnan. Path planning using a novel finite horizon suboptimal controller. *Journal of Guidance, Control, and Dynamics*, 36(4):1210–1214, 2013.

[49] A. Quilliot. A Short Note About Pursuit Games Played on a Graph with a given Genus. *Journal of Combinatorial Theory Series B*, 38(1):89–92, 1985.

[50] R. Nowakowski and P. Winkler. Vertex-to-vertex Pursuit in a Graph. *Discrete Mathematics*, 43(2-3):235–239, 1983.

[51] M. Aigner and M. Fromme. A game of cops and robbers. *Discrete Applied Mathematics*, 8(1):1–12, 1984.

[52] T. H. Chung, G. A. Hollinger, and V. Isler. Search and pursuit-evasion in mobile robotics. *Autonomous Robots*, 31(4):299–316, November 2011.

[53] N. Basilico, N. Gatti, and F. Amigoni. Patrolling security games: Definition and algorithms for solving large instances with single patroller and single intruder. *Artificial Intelligence*, 184-185:78–123, June 2012.

[54] G.M. Stanley and R.S.H. Mah. Observability and redundancy classification in process networks: Theorems and algorithms. *Chemical Engineering Science*, 36(12):1941–1954, 1981.

[55] A. Kretsovalis and R. Mah. Observability and redundancy classification in multicomponent process networks. *AIChE Journal*, 33(1):70–82, Jan. 1987.

[56] N.S.V. Rao. Computational complexity issues in operative diagnosis of graph-based systems. *IEEE Transactions on Computers*, 42(4):447–457, 1993.

[57] K. Chakrabarty, S.S. Iyengar, Hairong Qi, and Eungchun Cho. Coding theory framework for target location in distributed sensor networks. In *Proceedings of the International Conference on Information Technology: Coding and Computing*, pages 130–134, 2001.

[58] S. Y. Chen and Y.F. Li. Automatic sensor placement for model-based robot vision. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(1):393–408, 2004.

[59] D.J. Chmielewski, T. Palmer, and V. Manousiouthakis. On the theory of optimal sensor placement. *AIChE Journal*, 48(5):1001–1012, 2002.

[60] S. Sen, S. Narasimhan, and K. Deb. Sensor network design of linear processes using genetic algorithms. *Computer & Chemical Engineering*, 22(3):385–390, 1998.

[61] K. Worden and A.P Burrows. Optimal sensor placement for fault detection. *Engineering Structures*, 23(8):885–901, 2001.

[62] J. Berry, L. Fleischer, W. Hart, C. Phillips, and J. Watson. Sensor placement in municipal water networks. *Journal of Water Resources Planning and Management*, 131(3):237–243, 2005.

[63] J. A. Carballido, I. Ponzoni, and N. B. Brignole. CGD-GA: A graph-based genetic algorithm for sensor network design. *Information Sciences*, 177(22):5091–5102, Nov. 2007.

[64] S. Martínez and F. Bullo. Optimal sensor placement and motion coordination for target tracking. *Automatica*, 42(4):661–668, 2006.

[65] A.N. Bishop, B. Fidan, B. D.O. Anderson, K. Doğançay, and Pathirana P. N. Optimality analysis of sensor-target localization geometries. *Automatica*, 46(3):479–492, 2010.

[66] D. B. Jourdan and N. Roy. Optimal sensor placement for agent localization. *ACM Transactions on Sensor Networks*, 4(3):13:1–13:40, June 2008.

[67] D.C. Kammer. Sensor placement for on-orbit modal identification and correlation of large space structures. *Journal of Guidance, Control, and Dynamics*, 14(2):251–259, 1991.

[68] Y. Shastri and U. Diwekar. Sensor placement in water networks: A stochastic programming approach. *Journal of Water Resources Planning and Management*, 132(3):192–203, May-Jun. 2006.

[69] D. Feijer, K. Savla, and E. Frazzoli. Strategic dynamic vehicle routing with spatio-temporal dependent demands. In *Proceedings of the American Control Conference*, pages 3974–3979, 2012.

[70] S. Joshi and S. Boyd. Sensor selection via convex optimization. *IEEE Transactions on Signal Processing*, 57(2):451–462, 2009.

[71] R. Tharmarasa, T. Kirubarajan, and M.L. Hernandez. Large-scale optimal sensor array management for multitarget tracking. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(5):803–814, 2007.

[72] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.

[73] J. Yick, B. Mukherjee, and D. Ghosal. Wireless sensor network survey. *Computer Networks*, 52(12):2292–2330, 2008.

[74] M.M.R. Mozumdar, A. Ganesan, and A. Ameri. Synthesizing sensor networks backbone architecture for smart buildings. *IEEE Sensors Journal*, 14(12):4273–4283, Dec. 2014.

[75] J. Yick, B. Mukherjee, and D. Ghosal. Analysis of a prediction-based mobility adaptive tracking algorithm. In *2nd International Conference on Broadband Networks*, volume 1, pages 753–760, 2005.

[76] Kazuya Tsukamoto, Hirofumi Ueda, Hitomi Tamura, Kenji Kawahara, and Yuji Oie. Deployment design of wireless sensor network for simple multi-point surveillance of a moving target. *Sensors*, 9(5):3563–3585, 2009.

[77] Y. Chen, C. Chuah, and Q. Zhao. Sensor placement for maximizing lifetime per unit cost in wireless sensor networks. In *IEEE Military Communications Conference*, volume 2, pages 1097–1102, Oct. 2005.

[78] S. Mini, S.K. Udgata, and S.L. Sabat. Sensor deployment and scheduling for target coverage problem in wireless sensor networks. *IEEE Sensors Journal*, 14(3):636–644, Mar. 2014.

[79] V. Akbarzadeh, C. Gagne, M. Parizeau, M. Argany, and M.A. Mostafavi. Probabilistic sensing model for sensor placement optimization based on line-of-sight coverage. *IEEE Transactions on Instrumentation and Measurement*, 62(2):293–303, Feb. 2013.

[80] A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg. Near-optimal sensor placements: Maximizing information while minimizing communication cost. In *Proceedings of the 5th International Conference on Information Processes in Sensor Networks*, pages 2–10, 2006.

[81] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella. Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, 7(3):537–568, 2009.

[82] M. Cardei and D. Du. Improving wireless sensor network lifetime through power aware organization. *Wireless Networks*, 11(3):333–340, 2005.

[83] United States Air Force. United States Air Force Unmanned Aircraft Systems Flight Plan 2009-2047. *USAF Headquarters, Washington, DC*, 2009.

[84] N. Friedman. *US submarines since 1945: an illustrated design history*. Naval Institute Press, 1994.

[85] L. A. Young, E. Aiken, P. Lee, and G. Briggs. Mars rotorcraft: possibilities, limitations, and implications for human/robotic exploration. In *IEEE Aerospace Conference*, pages 300–318, 2005.

[86] H. Hourani. A marsupial relationship in robotics: A survey. *Intelligent Robotics and Applications, Part I*, 7101:335–345, 2011.

[87] R. R. Murphy. Marsupial and shape-shifting robots for urban search and rescue. *IEEE Intelligent Systems*, 15(2):14–19, 2000.

[88] A. Wolf, H. B. Brown, R. Casciola, A. Costa, M. Schwerin, E. Shamas, and H. Choset. A mobile hyper redundant mechanism for search and rescue tasks. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2889–2895, 2003.

[89] J. Huff, S. Conyers, and R. Voyles. Mothership - a serpentine tread/limb hybrid marsupial robot for usar. In *IEEE International Symposium on Safety, Security, and Rescue Robotics*, pages 1–7, 2012.

[90] A. Ferworn, C. Wright, J. Tran, Chao Li, and H. Choset. Dog and snake marsupial cooperation for urban search and rescue deployment. In *IEEE International Symposium on Safety, Security, and Rescue Robotics*, pages 1–5, 2012.

[91] R. R. Murphy, M. Ausmus, M. Bugajska, T. Ellis, T. Johnson, N. Kelley, J. Kiefer, and L. Pollock. Marsupial-like mobile robot societies. In *Proceedings of the 3rd Annual Conference on Autonomous Agents*, pages 364–365. ACM, 1999.

[92] M. Matusiak, J. Paanajärvi, P. Appelqvist, M. Elomaa, M. Vainio, T. Ylikorpi, and A. Halme. A novel marsupial robot society: Towards long-term autonomy. *Distributed Autonomous Robotic Systems 8*, pages 523–532, 2009.

[93] C. Zhou, Z. Cao, S. Wang, and M. Tan. A marsupial robotic fish team: Design, motion and cooperation. *Science China Technological Sciences*, 53(11):2896–2904, 2010.

[94] A. Drenner, M. Janssen, A. Kottas, A. Kossett, C. Carlson, R. Lloyd, and N. Papanikolopoulos. Coordination and longevity in multi-robot teams involving miniature robots. *Journal of Intelligent & Robotic Systems*, 72(2):263–284, 2013.

[95] B. W. Minten, R. R. Murphy, J. Hyams, and M. Micire. A communication-free behavior for docking mobile robots. In *Distributed Autonomous Robotic Systems 4*, pages 357–367. Springer, 2000.

[96] B. W. Minten, R. R. Murphy, J. Hyams, and M. Micire. Low-order-complexity vision-based docking. *IEEE Transactions on Robotics and Automation*, 17(6):922–930, 2001.

[97] M. Janssen and N. Papanikolopoulos. Enabling complex behavior by simulating marsupial actions. In *Mediterranean Conference on Control Automation*, pages 1–6, 2007.

[98] K. M. Wurm, C. Dornhege, P. Eyerich, C. Stachniss, B. Nebel, and W. Burgard. Coordinated exploration with marsupial teams of robots using temporal symbolic planning. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5014–5019, 2010.

[99] K. M. Wurm, C. Dornhege, B. Nebel, W. Burgard, and C. Stachniss. Coordinating heterogeneous teams of robots using temporal symbolic planning. *Autonomous Robots*, 34(4):277–294, May 2013.

[100] A. Drenner and N. Papanikolopoulos. A framework for large-scale multi-robot teams. In *Modeling and Control of Complex Systems*, pages 297–337. CRC Press, 2008.

[101] H. J. Min and N. Papanikolopoulos. Vision-based effective dispersion of miniature robots by using local sensing. In *SPIE Defense, Security, and Sensing*, volume 7332, pages 73321V–73321V, 2009.

[102] G.L. Gopalakrishnan. *Computation Engineering: Applied Automata Theory and Logic*, chapter 19, page 348. Springer, 2006.

[103] A. M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42(2):230–265, 1936.

[104] S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, pages 151–158, New York, NY, USA, 1971. ACM.

[105] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[106] K. Helsgaun. General k-opt submoves for the linkernighan tsp heuristic. *Mathematical Programming Computation*, 1(2-3):119–163, 2009.

[107] G.F. Franklin, J.D. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems*. Pearson Prentice Hall, 5th edition, 2005.

[108] A.T. Klesh, P.T. Kabamba, and A.R. Girard. Path planning for cooperative time-optimal information collection. In *Proceedings of the American Control Conference*, pages 1991–1996, June 2008.

[109] K.A. Ross. *Elementary Analysis: The Theory of Calculus*. Springer, 1980.

[110] M.R. Garey, R.L. Graham, and D.S. Johnson. Some np-complete geometric problems. In *Proceedings of the 8th ACM Symposium Theory of Computing*, STOC '76, pages 10–22, 1976.

[111] M.L. Dertouzos. Control robotics: the procedural control of physical processes. In *Proceedings of IFIP Congress*, pages 807–813, 1974.

[112] J.A. Ratches. Review of current aided/automatic target acquisition technology for military target acquisition tasks. *Optical Engineering*, 50(7):072001, July 2011.

[113] R. Dai and J. Cochran. Path planning and state estimation for unmanned aerial vehicles in hostile environments. *Journal of Guidance, Control, and Dynamics*, 33(2):595–601, 2010.

[114] B. Hyun, M. Faied, P. Kabamba, and A. Girard. Mixed-initiative nested classification by optimal thresholding. In *Proc. 50th IEEE Conference on Decision and Control and European Control Conf.*, pages 7653–7658, Dec. 2011.

[115] M. Niendorf, J. Las Fargeas, P. Kabamba, and A. Girard. Stability analysis of stochastic integer optimization problems. In *Proceedings of the 53rd IEEE Conference on Decision and Control*, pages 402–407, Dec. 2014.

[116] J. Las Fargeas, M. Niendorf, P. Kabamba, and A. Girard. Stability and criticality analysis for integer linear programs with markovian problem data. *IEEE Transactions on Automatic Control*, 2015. Under review.

[117] P. T. Kabamba and A. R. Girard. *Fundamentals of Aerospace Navigation and Guidance*. Cambridge University Press, 2014.

[118] A. E. Bryson Jr. and Y. Ho. *Applied Optimal Control*. Blaisdell Publishing Company, 1969.