

UC San Diego

UC San Diego Electronic Theses and Dissertations

Title

Optimal tasking of mobile autonomous sensing assets in uncertain adversarial settings

Permalink

<https://escholarship.org/uc/item/1d56081j>

Author

Oran, Ali

Publication Date

2010

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA, SAN DIEGO

**Optimal Tasking of Mobile Autonomous Sensing Assets in Uncertain
Adversarial Settings**

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Engineering Sciences (Mechanical Engineering)

by

Ali Oran

Committee in charge:

Professor William McEneaney, Chair
Professor Philip Gill
Professor Miroslav Krstic
Professor Sonia Martinez
Professor Ruth Williams

2010

Copyright
Ali Oran, 2010
All rights reserved.

The dissertation of Ali Oran is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2010

DEDICATION

To my family.

EPIGRAPH

-Silence is golden.

TABLE OF CONTENTS

Signature Page	iii
Dedication	iv
Epigraph	v
Table of Contents	vi
List of Figures	viii
Acknowledgements	ix
Vita and Publications	xi
Abstract of the Dissertation	xii
Chapter 1 Introduction	1
1.1 A Brief History	1
1.2 Modern UAV Technology: Its Advantages and Necessities	3
1.3 Analyzing Reconnaissance Missions	6
1.3.1 A Convoy Path Problem	6
1.4 The Sensor Tasking Problem	12
1.5 An Overview of Dissertation	14
Chapter 2 Observation Control Problem	15
2.1 Introduction	15
2.2 Assessing the Value of Information for Reconnaissance	
Missions	17
2.3 Observation Control Problem	24
2.4 Open Loop Analysis of the Urban Operation	26
2.4.1 Operation Setting	27
2.4.2 Formulation	28
2.4.3 Results	33
Chapter 3 The State Feedback Control Problem	38
3.1 Solution of the State Feedback Control Problem with	
Dynamic Programming	38
3.2 Extension of Theorem 3.1.4 to Other Possible Cases . . .	45
3.2.1 N UAVs for a Region of N Subregions	46
3.2.2 Single UAV for a Region of N Subregions	51

Chapter 4	Refining and Pruning Methods	63
4.1	Introduction	63
4.2	Refining the Index Set, \mathcal{I}_t	63
4.3	Pruning the Refined Set, \mathcal{R}_t	68
Chapter 5	Conclusion and Future Work	89
Bibliography	91

LIST OF FIGURES

Figure 1.1:	Simplified Example of a Mission Map	7
Figure 2.1:	Blue COA	20
Figure 2.2:	An example of $\bar{J}(q)$ (blue surface). Red planes are individual $v^i \cdot q$ defining $\bar{J}(q)$, whereas purple planes are inactive.	23
Figure 2.3:	Discretized Blue Course of Action	28
Figure 2.4:	Expected ground unit survivability and unit decisions in subregion 1 as a function of p^1	35
Figure 2.5:	Expected attrition on ground units following optimal UAV support. (On this figure axis q1 corresponds to p^1 and axis q2 corresponds to p^6).	36
Figure 2.6:	Expected attrition on ground units following heuristic UAV support. (On this figure axis q1 corresponds to p^1 and axis q2 corresponds to p^6).	37
Figure 2.7:	Expected percent change in attrition when switched from optimal UAV support to heuristic UAV support. (On this figure axis q1 corresponds to p^1 and axis q2 corresponds to p^6).	37
Figure 3.1:	Growth of \mathcal{I}_T	44
Figure 4.1:	A counter example for pruning with ε_∞ . Rather than any 2 of the original 3 vectors (solid lines), vectors with dashed lines comprise the optimal set approximating $\max_i(v^i \cdot q)$	73

ACKNOWLEDGEMENTS

I would like first and foremost thank my adviser Prof. William M. McEneaney for his support, guidance, and mentorship throughout my doctoral study. Not only I learned the methods of Optimal Control Theory from him but also how to work with people under my supervision.

I would also like to express my sincere thanks for my thesis committee members, Professors Philip Gill, Miroslav Krstic, Sonia Martinez, and Ruth Williams, for their valuable time and consideration. I am also thankful to them for helping me in shaping the final form of my thesis and their valuable suggestions.

I would also like to thank the UCSD MAE Department graduate student advisers and coordinators for their assistance throughout my doctoral and earlier masters education.

I would also like to thank UCSD Affiliated Housing Office and UCSD Office of Graduate Studies and Research for their support again throughout my doctoral and earlier masters education.

I believe there many other people worked at UCSD that I never met in person but their efforts provided an enjoyable and efficient study environment for UCSD students. For this reason, I would like to thank all members of the UCSD community for their support of the students.

This dissertation includes reprints of the following papers.

Chapter 2 is in part a reprint of:

Ali Oran and William McEneaney “Max-Plus Enabled Dynamic Programming for Sensor Platform Tasking”, *American Control Conference*, 2010.

William McEneaney and Ali Oran “Observation Process Control in Support of Stochastic Tasking Operations”, *Mathematical Theory of Networks and Systems*, 2010.

William McEneaney, Ali Oran and Andrew Cavender “Value-Based Control of the Observation-Decision Process”, *American Control Conference*, 2008.

William McEneaney Ali Oran and Andrew Cavender “Value-Based Tasking Controllers for Sensing Assets”, *AIAA Guidance, Navigation & Control Conference*, 2008.

Chapter 3 is in part a reprint of:

Ali Oran and William McEneaney “Max-Plus Enabled Dynamic Programming for Sensor Platform Tasking”, *American Control Conference*, 2010.

William McEneaney and Ali Oran “Observation Process Control in Support of Stochastic Tasking Operations”, *Mathematical Theory of Networks and Systems*, 2010.

William McEneaney, Ali Oran and Andrew Cavender “Value-Based Control of the Observation-Decision Process”, *American Control Conference*, 2008.

William McEneaney Ali Oran and Andrew Cavender “Value-Based Tasking Controllers for Sensing Assets”, *AIAA Guidance, Navigation & Control Conference*, 2008.

Chapter 4 is in part a reprints of:

Ali Oran and William McEneaney “Max-Plus Enabled Dynamic Programming for Sensor Platform Tasking”, *American Control Conference*, 2010. William McEneaney and Ali Oran “Observation Process Control in Support of Stochastic Tasking Operations”, *Mathematical Theory of Networks and Systems*, 2010.

The dissertation author was the primary author and the co-author listed in these publications directed and supervised the research.

VITA

2001	B. S. in Mechanical Engineering, Istanbul Technical University, Turkey
2002-2003	Graduate Research Assistant, Bogazici University, Turkey
2005	M. S. in Engineering Sciences (Mechanical Engineering), University of California, San Diego
2005-2010	Graduate Research Assistant, University of California, San Diego
2010	Ph. D in Engineering Sciences (Mechanical Engineering), University of California, San Diego
2010	Engineering Intern, Insight Access

PUBLICATIONS

Ali Oran and William McEneaney, “A Max-Plus Based Method for Sensor Tasking”, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, (submitted).

Ali Oran and William McEneaney “Max-Plus Enabled Dynamic Programming for Sensor Platform Tasking”, *American Control Conference*, 2010.

William McEneaney and Ali Oran “Observation Process Control in Support of Stochastic Tasking Operations”, *Mathematical Theory of Networks and Systems*, 2010.

William McEneaney, Ali Oran and Andrew Cavender “Value-Based Control of the Observation-Decision Process”, *American Control Conference*, 2008.

William McEneaney Ali Oran and Andrew Cavender “Value-Based Tasking Controllers for Sensing Assets”, *AIAA Guidance, Navigation & Control Conference*, 2008.

ABSTRACT OF THE DISSERTATION

Optimal Tasking of Mobile Autonomous Sensing Assets in Uncertain Adversarial Settings

by

Ali Oran

Doctor of Philosophy in Engineering Sciences (Mechanical Engineering)

University of California, San Diego, 2010

Professor William McEneaney, Chair

In this study we consider the sensor tasking problem, with the case of military reconnaissance as a motivating example. In order to determine the optimal control set, we develop a mathematical formulation for the value of information. This value is inherited from the expected payoff for the activity which uses the information. We address the problem of optimal control of the sensing assets given this inherited payoff through both open loop and state feedback control approaches. A micro-UAV deployment scenario for a three stage urban operation was analyzed with the open-loop approach. For the feedback case, it was found that a particular generic form of the value function is preserved under backward dynamic programming. This form is exploited to develop an idempotent-based

numerical approach. Also, efficient refining and pruning methods were developed for attenuating the curse-of-complexity associated with this class of methods.

Chapter 1

Introduction

In the beginning of 2010s, unmanned mobile sensing assets, such as Unmanned Airborne Vehicles (UAVs), are bringing an enormous amount of support to the military operations worldwide. Especially through their deployment in reconnaissance missions, these state of the art vehicles have showed their superior capabilities since the mid 1990s. Meanwhile, as a result of their quick deployment in recent conflicts following their developments, there are still a number of issues that needs to be addressed concerning these vehicles. In this study we develop a new methodology to improve the mission performance of the most utilized type of mobile sensing missions, the reconnaissance-based sensing mission, through optimal control theory formulations. In this introductory chapter we will address the current needs of the UAV technology, and also give an idea of our foundational motivations for this study. Before moving into these points, we would like to briefly mention some of the early pioneers in aerial reconnaissance technology, and also introduce the reader to world's possibly first open-loop controlled UAV from the 19th century.

1.1 A Brief History

The utilization of aerial vehicles in support of the military units, operating on the ground, can be traced back to the early discoveries of flight. One of the earliest aerial vehicle in human history is the kite, developed around 300BC by

the Chinese. Soon after its discovery, its possible military potentials were realized, and they were utilized for the Chinese military operations, such as lifting men into the air with a kite to spot the enemy, and to track their movements. This development was also the birth of aerial reconnaissance. Soon after this, again the Chinese developed the Kongming Lantern, the first pre-modern hot air balloon and they again quickly utilized this new discovery in a similarly fashion for the military operations, such as for signaling purposes within the army ranks. Later, towards the end of the 18th century French inventors Joseph and Michel Montgolfier developed the modern hot air balloon in 1783. Once again, soon after this aerial discovery, its possible military benefits got noticed, and the French became the first nation to exploit this new technology by forming the French Aerostatic Corps in 1794, which is now considered world's first Air Force. As can be guessed, the Corps primary duty was to collect reconnaissance information, and even in the year of their foundation they found the opportunity to show the potentials of airborne reconnaissance at the battle of Fleurus against the Allied forces. The balloon, *l'Entreprenant*, operated by the Aerostatic Corps, continuously informed the French general Jean-Baptiste Jourdan about the Austrian army's movements, and gave the French side a considerable advantage. [Buc99][Boy03] In a similar fashion to this pioneering discoveries, during the following centuries every new aerial discovery such as the aeroplanes, or later the jet motor planes was successfully exploited for reconnaissance advantages soon after their discoveries as well.

While these developments were the examples of early manned reconnaissance missions, the earliest unmanned airborne operation did not happen until the Austrian siege of Venice in 1849. When the Austrians couldn't surrender the Venetians after a long artillery fire, a bright Austrian artillery officer Lieutenant Franz Uchatius, envisaged the use of balloons to drop bombs on the city. Although sort of mocked by a few journals back at the day, Uchatius developed the path plan for the balloons, considering the wind, distance and other factors. This try might also be considered in history books as the first open-loop controlled aerial attack mission. In the end like many open-loop controlled systems the noise (the wind) became the main factor that reduced the efficiency of the operation. [Boy03]

Although this operation raised more interest in aerial vehicles for usage in military operations besides the reconnaissance, it took another 100 years for the aerospace and control technology to mature up to be used for the unmanned aerial operations. It was not before the end of the second world war, when nations systematically started developing UAVs.

1.2 Modern UAV Technology: Its Advantages and Necessities

After the second world war advanced vehicle control systems and sensor technology had matured up to a point sufficient enough for their utilization in UAVs, and UAV programs have been developed since then. However their deployment were negligible during the Cold War years, and until the beginning of 20th century aerial missions, including reconnaissance operations, were still carried out by manned aerial vehicles, such as the famous Lockheed U-2. Meanwhile, after the early 1980s, with the advent of enhanced satellite communications, miniaturized electronics, and sophisticated sensors, UAVs' possible future mission capabilities were realized, and a rapid UAV development was planned by the UAV Joint Program Office [DO05]. Following their technological development, like their technological ancestors, the UAVs also found their initial deployments in military environments, such as during the conflicts in late 1990s, and in 2000s, for reconnaissance missions. In a very short period of time after their developments and initial deployments, being noticed for their up to date successful operations, in the beginning of 2010s, they are currently being considered as an alternative option to many manned aerial reconnaissance units.

This rather quick consideration might come as a surprise to many but there are several advantages UAV missions possess compared to traditional aerial manned missions. First of all, not having a human pilot on board the vehicle gives the mission command & control office the opportunity to deploy the UAVs in situations that would be considered risky for a human pilot. Second, the cost of manufacturing and maintaining of a UAV fleet could be comparably lower than

the cost of a traditional fleet with the addition of pilot training costs. Third, and of particular importance, UAV missions bring what is termed as "persistence" to intelligence-collection capabilities. By this term we address a UAV's capability of positioning itself over a specified area of interest for 24 hours or longer, something not obtainable from manned aircrafts or spacecrafts. [DO05] The reader should particularly notice the following assessment of the superiority of Global Hawk, a particular UAV, over other aerial assets.

The lack of "persistence" has long been stated as a major deficiency in U.S. intelligence-collection capabilities. Orbiting satellites revisit specified targets only a few times a day and on a predictable schedule. An enemy discovering that schedule can "hide" from the satellite observation. Manned aircraft (for example, the U-2), while not as predictable as satellites, can stay over a specified target area for only a short time. Their small inventories make revisiting these areas infrequent in most circumstances. Global Hawk, in contrast, can orbit over a specified area for 24 hours or longer, continuously streaming data on the target area back to the United States for near-real-time processing and dissemination. The ability to gain an enhanced degree of "persistence" was probably the strongest argument for acquiring Global Hawk. [DO05]

Fourth, UAVs come in different sizes. MicroUAVs unlike a manned aircraft can be carried by a mere human being, and could be deployed in the front lines against an enemy immediately after the platoon commander might need intel assistance. Additionally, such UAVs might not need their observations exploited at an image recovery site, but at the platoon commander's camp for rapid analysis.

Meanwhile considering their rapid development, and early deployment in current conflicts, at the present time there are several issues that need to be addressed regarding them. While each UAV has its own particular potential problems, there are also common issues that need attention as well. We want to list some of the known problems here. The first possible problem stems from the deployment and later control characteristics of the UAVs. For example, Global Hawk, one of the most capable UAV, has its mission pre-planned, and it completes this mission by flying from one given way point to another using its autopilot. Here, the pre-planned mission uses older information, and although the mission plan can

be changed at flight, at today's dynamic settings we need more adaptive automated decision mechanisms that would let the UAV make the decisions by itself when necessary. An early publication analyzing the challenges of the autonomous control of the UAVs, states this issue very well. "Near real-time replanning is needed as new sensor information, commands, or intelligence is received by the UAV. The UAV starts with a plan defined off-line, and the challenge is to optimally update that plan as new information is received and/or unforeseen events occur." [PC98] Second, the UAV needs to maintain a connection to an operation base, where the UAV observations (such as imagery) will be processed and analyzed. From that base the updated information would be introduced to the field commander who might be needing the intel assistance in the first place. In a worst case scenario the adversary might jack the connection between the operation base and the UAV or the field commander's camp, leaving the UAV operation inefficient. A possible solution would be to have the UAV have a direct connection with the field commander. A related problem that is worth mentioning is the possibility of excess information buildup from the sensing asset observations. While the sensing observations are being continuously channeled to the operation base, there may not be enough personnel to analyze this huge amount of data, plus even so it might not be optimal to spend the limited resources for analyzing such a huge amount of raw data. This issue was pointed out in one of the recent issues of the IEEE Spectrum magazine.

In 2009 alone, the U.S Air Force shot 24 years' worth of video over Iraq and Afghanistan using spy drones. The trouble is, there aren't enough human eyes to watch it all. The deluge of video data from these unmanned aerial vehicles, or UAVs, is likely to get worse. By next year, a single new Reaper drone will record 10 video feeds at once, and the Air Force plans to eventually upgrade that number to 65. John Rush, chief of the Intelligence, Surveillance and Reconnaissance Division of the U.S. National Geospatial-Intelligence Agency, projects that it would take an untenable 16000 analysts to study the video footage from UAVs and other airborne surveillance systems. [Ble10]

Following these mentioned issues, at the present time there is a strong need for automated decision-support tools for the UAVs that would give the UAVs more

autonomy and in return make their missions safer and more efficient. Since the beginning of 2000s a great amount of research have been dedicated to developing control algorithms for the UAVs by following different approaches. Meanwhile automated decision-support tools also bring many new issues to existing ones. Especially, the complexity associated with proposed mission models becomes a drawback for the developed algorithms. Articles by Clough in [Clo02], and Pachter and Chandler in [PC98] list some of the issues in this field.

1.3 Analyzing Reconnaissance Missions

As mentioned earlier, airborne reconnaissance units has been utilized for more than 2000 years, with the purpose of spotting enemy's movements and capabilities in order to have an advantage on the battlefield. Following the introduction of mobile units to the battlefield such as tanks, armored personal carriers and bombers there is currently a much greater necessity to keep a constant eye on the enemy, for the dynamics of the battlefield has become a very rapidly evolving game in the past century. In such large, dynamics settings with very different scenario possibilities, one needs to consider not only merely reducing the uncertainty, but also to compare his possible options about how much he would benefit as a result of any action done for the purpose of reducing uncertainty. Basically, when analyzing modern reconnaissance operations one should focus on maximizing his benefits, which would also be a function of the uncertainty, rather than bluntly minimizing the uncertainty. To give an idea about this concept, we propose the following simple problem as a motivation.

1.3.1 A Convoy Path Problem

Suppose that a military convoy will be advancing from an initial position to its destination, and the convoy leader has to decide whether to pass through the desert or through the forest. Figure 1.1 very simply illustrates such a scenario where the convoy can advance from point A to point B through Path 1 (through the desert) or Path 2 (through the forest). Intel reports that there are high adversarial

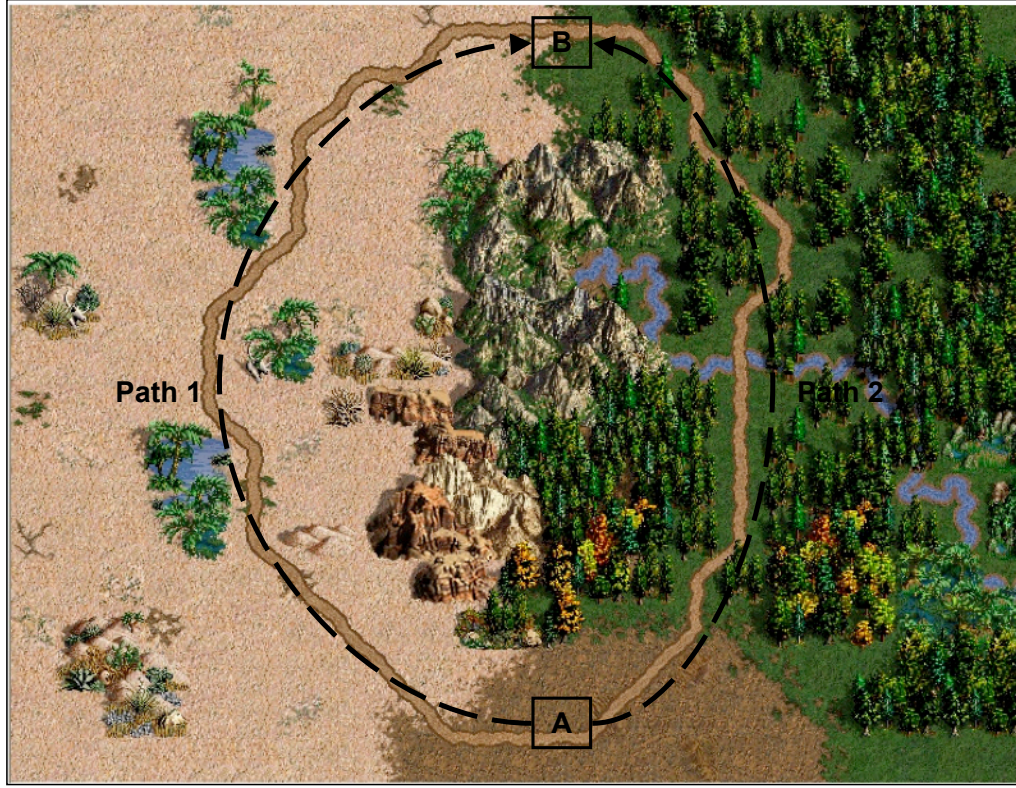


Figure 1.1: Simplified Example of a Mission Map

activity in the area, and for this reason it is very likely that the convoy will come under fire. Meanwhile, intel can not provide a certain information about the presence of the adversary in neither of the paths, and the convoy leader needs to decide which path to follow under these uncertain circumstances. In order to give a structured example, we start our discussion of this problem by considering the case when there are no reconnaissance options (such as UAV deployment) available to the commander, and later carry the framework to a reconnaissance assisted operation, which is the real motivation for the remaining chapters.

In a situation where there are no reconnaissance options available to the commander, if the length of both paths were not very different, one would expect the convoy leader to choose to go through the desert path (safer option) to avoid possible ambushes in the forest area where enemy hideouts would be harder to detect by the radar of the convoy vehicles. At this point, if we try to analyze and quantify commander's decision-making process, we would notice that his final

decision is basically based on two factors; first, the possible attrition his convoy might suffer, and second, the operation cost at each path (such as fuel cost or operation time cost). For a very brief quantitative formulation let $x \in \{1, 2\}$ represent the state of the region of interest with the state space being the set of possible enemy forces' locations (enemy along path 1 or 2). Also let $u^g \in \{1, 2\}$ denote his decision option (whether to choose path 1 or 2). Following this argument, one might formulate an overall mission cost function, $J(x, u^g)$ as:

$$J(x, u^g) \doteq T(x, u^g) + C(x, u^g) \quad (1.1)$$

where we used $T(x, u^g)$ as the cost function due to attrition on the convoy, and $C(x, u^g)$ as the cost function due to everything other than the attrition, given state x and decision u^g . With this formulation, commander's final decision, \bar{u}^g , will be minimizing the mission cost function:

$$\bar{u}^g \doteq \operatorname{argmin}_{u^g \in \{1, 2\}} \left\{ \max_{x \in \{1, 2\}} J(x, u^g) \right\} \quad (1.2)$$

Now, as noticed from 1.1, in order to model the convoy leader's decision process in this scenario, both the cost of attrition, $T(x, u^g)$, and other costs, $C(x, u^g)$ needs to be formulated. Compared to $T(x, u^g)$, the mathematical formulation of $C(x, u^g)$ can be done much more easily, and has been majorly developed for many military logistics operations since the second world war. A recent example of such an analysis was utilized during the operation desert storm in the early 90s, for details see Hilliard et al. [HSL⁺92]. On the other hand, while an experienced military officials can assess an overall payoff function very quickly in his mind, explicit formulation of the attrition cost function, $T(x, u^g)$, could become more challenging. Still, by careful modelling of the interactions between the convoy and the adversary, the attrition function $T(x, u^g)$ could also be modeled mathematically, and a quantitative model of the complete mission, considering attrition, could be developed. Although a model of attrition might seem a little futuristic, recently developed Real-time Adversarial Intelligence and Decision-making (RAID) project could be such an example of mission planning considering attrition, for details see Kott and Ownby [KO07].

Back to our motivating example, notice that had this example been formulated in a perfect information (deterministic) setting where the convoy leader had known the adversaries locations, minimization of $J(x, u^g)$, and finding the optimal path would have become straightforward after the formulation of $T(x, u^g)$ and $C(x, u^g)$. In this more complex (and realistic) situation the convoy leader might only know partial information about the enemy, for example intel could only give an estimated probability of the presence of the adversary rather than certain information. For example, assume that the intel gives the convoy commander an estimate of the enemy presence in each path, with Path 1 (through the desert) having a probability of %80 and Path 2 (through the forest) of %20 chance of enemy. In this situation, x , the state of the region, would no longer be a variable but would become a random variable, and so would $T(x, u)^g$ and $C(x, u^g)$. In this case blue knowledge of the system state could be specified as a probability distribution q . We let q_0 denote the initial probability distribution available to the commander, in this example $q_0 = [0.8 \ 0.2]^T$. Depending on the decision analysis criteria utilized to analyze this situation, this time the safer option of the desert path might be considered more dangerous considering the possible attrition the team might suffer. For this type of a problem Bayes' Decision Rule could possibly be employed with consideration of expected values of attrition on the convoy at each path. Other possible analysis could be the conservative Worst Case Scenario Analysis or the Maximum Likelihood criterion [HL05]. For this simple example, assume that the commander formulates the simple $T(x, u^g)$ function as, "Probability of survivability of the convoy following path u , given enemy state x ", and remember that we have let $x \in \{1, 2\}$ where x will only be the state reflecting adversaries location (being at path 1 or at 2).

$$T(x, u^g) = P(\text{Convoy survives at path } u^g \mid \text{state} = x) = \begin{cases} \text{for } u^g = 1 & \begin{cases} 0.9 & x=1 \\ 1 & x=2. \end{cases} \\ \text{for } u^g = 2 & \begin{cases} 1 & x=1, \\ 0.7 & x=2. \end{cases} \end{cases}$$

Following Bayes' Decision Rule, with considering probabilities of each state, convoy

leader's decision could be formulated as:

$$\bar{u}^g = \operatorname{argmax}_{u^g} \mathbf{E}_x[J(x, u^g)] \quad (1.3)$$

where we used \mathbf{E}_x to denote the expectation over the red states. Following this formulation the expected troop survivability probabilities at each path could be found as:

$$\text{Expected Survivability of Convoy Through Path 1} = 0.8 * 0.9 + 0.2 * 1.0 = 0.92$$

$$\text{Expected Survivability of Convoy Through Path 2} = 0.2 * 0.7 + 0.8 * 1.0 = 0.94$$

Here, for a concise discussion, we omitted the operational costs of the mission, $C(x, u^g)$ in this example. (This could also be thought as a result of same lengths of path 1 and path 2.) Following this omission, and the results above, our analysis suggests that the commander would favor to choose the more risky forest path rather than the desert in this imperfect information setting.

Now, in the final part of this example, we assume the same situation when the convoy leader has a micro UAV available for deployment for a reconnaissance operation. With the addition of the reconnaissance unit/units, the problem becomes more complex. The convoy leader needs to decide (if given for his authority) where to send the reconnaissance unit, and how to interpret the observation results. For this simple problem it will be 'would sending the UAV to desert improve my cost function better than sending it to the forest'. Since observation process might also be corrupted (intentionally by the enemy or wrong observations) the results might require more careful analysis. In literature there are different approaches about how to approach uncertainty, and how to reduce it by the aerial missions. One common methodology is to utilize the entropy argument. But, besides these approaches one could also utilize the payoff function, $J(x, u^g)$, in 1.1 to analyze the effects of uncertainty and reconnaissance. Briefly speaking, one can estimate the future probability distribution of the state of the region, \hat{q} , from the initial distribution, $q_0 = [0.8 \ 0.2]^T$, through possible observations the UAVs would yield. Considering these possible a posteriori distributions one could find the possible changes in $J(x, u^g)$ related to UAV controls. Again omitting $C(x, u^g)$,

following 1.3 one can formulate:

$$\bar{u}^g = \operatorname{argmax}_{u^g} \mathbf{E}_{\hat{q}}[T(x, u^g)] \quad (1.4)$$

Since \hat{q} is dependent on the UAV controls, u^o , one should find the optimal UAV control initially. The optimal UAV control can be formulated as:

$$\bar{u}^0 = \operatorname{argmax}_{u^o} \mathbf{E}_{y(u^o)}[\mathbf{E}_{\hat{q}(y, u^o)}[T(x, u^g)]] \quad (1.5)$$

The first thing to do in this analysis is to update the probability distribution considering possible observations, $y(u^o)$. For instance, a Bayesian estimator could be utilized for this purpose. We leave the details of the estimator to chapter 2, and in here only mention the updated probability vector, \hat{q} found after the estimation for this problem. We assume that $y(u^o)$ takes values in the set $\{1, 2\}$, where 1 corresponds to an observation that results in the detection of the adversary, and 2 corresponds to an observation that does not result in the detection of the adversary. Since $y(u^o)$ could be corrupted by noise we assume that it is also a random variable, and one can find the probability of each observation result by conditional probability formulations:

$$P(y(u^o) = 1) = \sum_i P(y(u^o) = 1 | x = i) P(x = i) \quad (1.6)$$

In our simulation, the Bayesian estimators would yield the following results. Notation-wise we used $\hat{q}^{y(u^o)=1}$ to denote the updated probability given the UAV decision u^o and the observation $y(u^o) = 1$.

$$\text{for } u^o = 1 : \quad \hat{q}^{y(u^o)=1} = \begin{bmatrix} 0.9863 \\ 0.0137 \end{bmatrix}, \quad \hat{q}^{y(u^o)=2} = \begin{bmatrix} 0.2963 \\ 0.7037 \end{bmatrix}$$

$$\text{for } u^o = 2 : \quad \hat{q}^{y(u^o)=1} = \begin{bmatrix} 0.1818 \\ 0.8182 \end{bmatrix}, \quad \hat{q}^{y(u^o)=2} = \begin{bmatrix} 0.9744 \\ 0.0256 \end{bmatrix}$$

On the other hand,

$$\begin{aligned} \text{for } u^0 = 1: \quad & P(y(u^o) = 1) = 0.73, \quad P(y(u^o) = 2) = 0.27, \\ \text{for } u^0 = 2: \quad & P(y(u^o) = 1) = 0.22, \quad P(y(u^o) = 2) = 0.78. \end{aligned}$$

At this time we have analyzed all possible observation controls the commander could consider for the UAV, and also the possible results corresponding to each control. This kind of analysis would actually be the open-loop control of a UAV mission, and we discuss this in further detail in chapter 2. A better approach would be utilize the state feedback approach and this is discussed again in 3. In the last stage we need to find the expected value of the mission cost function as a function of each possible UAV decision, u^o .

$$\text{for } u^o = 1 : \quad \mathbf{E}_{y(u^o)}[\mathbf{E}_{\hat{q}(y, u^o)}[T(x, \bar{u}^g)]] = 0.989 \quad (1.7)$$

$$\text{for } u^o = 2 : \quad \mathbf{E}_{y(u^o)}[\mathbf{E}_{\hat{q}(y, u^o)}[T(x, \bar{u}^g)]] = 0.990 \quad (1.8)$$

Considering this values, we can conclude that the convoy commander should choose to task the UAV to make sensing on the adversary on path 2 rather than path 1.

Having found the optimal path the UAV should be sent to, we complete our discussion of this motivational example. Our main motivation was to show the necessity to consider the question of who would benefit most from a possible UAV reconnaissance operation before defining a payoff function. This will be basic idea on which we will build up our formulations.

1.4 The Sensor Tasking Problem

Following the discussion in the previous section, we are now ready to define the general sensor tasking problem. For this purpose, consider a region of interest, that we denote with \mathfrak{R} , where two adversaries Team 1 (Blue team) and Team 2 (Red team) compete in a zero sum game manner. For example, such a region could be a convoy path that might come under attack as mentioned in the motivating example earlier or a general battleground. We will analyze the game between the two adversaries from the view point of the Blue team which will have only partial (imperfect) information about the Red team. One should notice that the perfect information setting could be considered as a special case of this problem, and our analysis will still be valid for such settings as well. In order to improve its partial information, Blue team is expected to deploy its mobile sensing assets, such as

UAVs with sensors, and observe the region \mathfrak{R} from an initial time of $t = t_0$ to a final time of $t = T$. Here, what we mean with the action of observing is, the utilization of mobile sensors for detecting the enemy's (Red team units) presence and also its states, such as its arms capability, health state and other qualities. Briefly speaking, the mobile sensors will be undertaking reconnaissance missions in an uncertain environment, and their observation returns will be later utilized by the Blue team for other, preceding operations. While this problem could be defined in continuous time, modelling the observation process and solving the optimal tasking problem would then become unnecessarily challenging. A more practical approach would be to work on the discrete time domain. For defining intervals on an integer set, we will use the following integer interval notation for the rest of this study.

$$[a..b] \doteq \{x \in \mathbb{N}_0 \mid a \leq x \leq b\}, \quad \mathbb{N}_0 = \{0, 1, 2, \dots\} \quad (1.9)$$

Now, we define the sensor tasking problem for future references.

Definition 1.4.1. *The Sensor Tasking Problem:*

In some physical region \mathfrak{R} where two adversaries, Blue and Red Teams, compete in a zero sum game manner, by considering all possible locations the Red team units might be present in this region, and taking Blue teams's partial information into account, decide on which of these locations Blue team's mobile sensing units should focus their sensors during the time interval of $[t_0..T]$ such that the observation returns would yield the maximum advantage (benefit) to the Blue team for the upcoming operations.

Notice that this definition is a very general problem statement, and for this reason researchers working on this problem consider subcategories of it with their own assumptions and constraints. Throughout this dissertation we will also make necessary assumptions to make the analysis feasible but still valid for the problem defined above.

1.5 An Overview of Dissertation

In chapter 2, we will develop the basic formulations of the optimal sensor tasking problem following the general problem statement we introduced in definition 1.4.1. After an initial discussion, we will start our formulations by discussing characteristics of reconnaissance operations in 2.2. Later we will develop a measure for the sensing operations. In section 2.3, we connect this measure to the decisions regarding the sensing operations. We will develop both open and closed loop formulations for the solution of the sensing problem. Later in 2.4 we give an example about the utilization of the theory in an urban operation.

In chapter 3 we will discuss the closed loop solution of the optimal sensor tasking problem in more detail. First we will develop dynamic programming iterations for the solution of the state feedback problem. Then we will show a way to utilize the dynamic programming algorithm with greater simplicity. In section 3.2 we will analyze possible extensions of this methodology that improves the computation issues of dynamic programming iterations.

In chapter 4, we will introduce two other methods to improve the computation times demanded for the solution of the dynamic programming formulations. The first one of these methods, is called Refining, and it is based on utilizing the linear programming methods to seek for redundant information produced during dynamic programming iterations. The second one is called Pruning, and it will give us approximations of the original value function within specified error bounds, while improving computation times.

Chapter 2

Observation Control Problem

In this chapter we will develop the basic formulations of the optimal sensor platform tasking problem following the general problem statement we introduced in the previous chapter, in definition 1.4.1. We start our discussion by laying out the general ideas of the proposed methodology. Later, in section 2.2 we discuss possible approaches to quantify the value of sensing operations, and define a unique payoff function as a measure of the sensing actions. In section 2.3 we complete the optimal tasking formulations by introducing methods to update the available information, and also introduce open and closed loop control formulations to solve the problem. In section 2.4 we utilize the developed theory for an urban mission scenario. This analysis will be based on an open-loop controller. We leave the results related to the closed-loop control to chapter 3 for we will exploit the structure of the payoff function during dynamic programming iterations.

2.1 Introduction

The general sensor tasking problem defined in 1.4.1 can be thought as an optimal decision-making problem, where the decisions are the focus points of the mobile sensor units and the objective function being maximized is the advantage being returned to the Blue team as a consequence of these decisions. One way to approach, and solve this problem could be done by employing the methods of optimal control theory. In this methodology, as a first step, one would need to

formulate a relevant payoff function, $\mathcal{J}(\cdot)$, which would be a quantitative measure of the *advantage* to the Blue team of the consequences of the sensor observations. Later, by maximizing this measure over the set of sensor focus points one could arrive at the optimal set for the sensors.

At this point one can notice that the necessity to reformulate a new payoff function before each sensor mission with detailed analysis of each mission's objective might become a demanding job for the mission planners. Especially for missions that are time sensitive, such as missions involving micro-UAVs which could be deployed by the combat units on the ground to detect enemy presence around them, there might not be such luxury of lengthy mission planning times. Because of this possible drawback, the proposed method might not seem like a practical way to approach the sensor tasking problem. Meanwhile, the need for reformulating the payoff function before each mission could be avoided by noticing the common objectives in mobile sensor deployments, and by grouping the sensor missions according to their mission objectives. After this grouping, by considering the common objectives of each group of sensing missions, generic payoff functions could be formulated for each group. This way, payoff functions that are defined for a specific group could be utilized for all the missions belonging to that group with slight modifications, and the burden of reformulation of payoff function before each mission could be avoided.

In this study, as mentioned earlier in 1.4, we have assumed that the mobile sensors would be deployed in a region with the presence of an adversary, and that they would be on a reconnaissance mission. Under this assumption, the generic form of $\mathcal{J}(\cdot)$ could be very well formulated considering the general nature of reconnaissance missions. By the fact that, reconnaissance based UAV missions compromise most of UAV operations, a study based on this group of missions could be very useful for the UAV research field. Similarly reconnaissance based operations involving other kinds of autonomous sensing vehicles, such as UUVs (Unmanned Underwater Vehicles), could also benefit from this new approach as well.

Once the relevant payoff function is defined, possible sensing decisions could

be evaluated by considering the outcomes of each decision. In this context, this approach have similarities to solving a sequential decision making problem with modern decision-making theory. Considering the similarities, we list the major steps in this approach below. (taken from [KK81])

1. List all possible actions.
 2. List all possible outcomes.
 3. Assess the probability of each outcome from each action.
 4. Choose the best action based on likelihood and utility of the outcome.
- $$\left. \vphantom{\begin{array}{l} 1. \\ 2. \\ 3. \\ 4. \end{array}} \right\} \quad (2.1)$$

In our analysis the possible actions will be the set of possible focus points for the mobile sensing assets, and the possible outcomes will be the observation results of the assets regarding the adversary. What will be different is that, in the last step the likelihood criterion and the utility of the outcome will be replaced by Bayes' decision rule and the payoff function $\mathcal{J}(\cdot)$, respectively. We will give greater details of these steps with an example in section 2.4. Before that, we will discuss our formulation of the generic payoff function for the sensing missions that could be put under the general group of reconnaissance missions.

2.2 Assessing the Value of Information for Reconnaissance Missions

From common knowledge, one is well aware that a reconnaissance operation is aimed at improving our knowledge about a non-perfect information setting, which could be also dynamic. We use the term setting to define the combined concept of the physical region of interest \mathfrak{R} , plus entities that are present on this region which will have a possible affect on the blue team's upcoming missions. Considering this objective, when formulating the generic payoff function for the group of reconnaissance based mobile sensing missions, mission planners should distinct the most essential possible part of the payoff function, their knowledge about this setting. For this reason, for the group of reconnaissance based mobile

sensing missions we suggest that $\mathcal{J}(\cdot)$ should be formulated as a sum of two distinct functions, $C(\cdot)$ and $\bar{J}(\cdot)$, similar to the earlier formulation we introduced in (1.1).

$$\mathcal{J}(\cdot) \doteq \mathbf{E}\left\{C(\cdot) + \bar{J}(\cdot)\right\}, \quad (2.2)$$

where the expectation will be taken over the unknown states of the mission setting, the initial states, and the state transitions. We define these two functions as:

$\bar{J}(\cdot) \doteq$ the payoff function that is directly related to the information aspect of the sensor mission

$C(\cdot) \doteq$ the cost function for the mobile sensing mission.

$C(\cdot)$, for most operations, is simply the traditional cost of the sensing action, i.e the cost of loss and maintenance of the sensing asset. Meanwhile depending on the particular mission it could also include any other cost that is not linked to the information aspect of the mission, such as penalty related to the mission completion time. It has been studied well in the literature, see for example Hilliard et al. [HSL⁺92] and for this reason, we will not be further analyzing it in our research. On the other hand, $\bar{J}(\cdot)$, the payoff related the information aspect of the mission, has been less studied, and developed. We also believe that it is the more critical part of $\mathcal{J}(\cdot)$ for the reconnaissance-based mobile sensor mission, and for this reason we will focus our attention on this part throughout the rest of this study.

The problem associated with $\bar{J}(\cdot)$ is the difficulty of formulating it in terms of available information, or better said, the difficulty of quantifying ones knowledge about the enemy through this function. This is critical for one would need an accurate measure by which the optimality level will be defined for the function $\mathcal{J}(\cdot)$.

A good candidate for $\bar{J}(\cdot)$ could be based on the entropy function, introduced by Shannon. It has been used well-before the deployment of the UAVs, for the manned reconnaissance mission,s by the aim of reducing uncertainty in given settings [Dan62], and is still considered by researchers for the same purpose [BS96], [YY10]. However, one could notice that the entropy measure is not directly

tied to the associated ongoing or future operations which will be making use of the state information. We believe that this lack of direct relation with the related operations is a big disadvantage of the entropy approach, and a similar criticism can also be found in [Whi75]. For this reason, we examine the structure of the payoff for observation in terms of the expected benefit to the operations which is to follow. Now, in order to formulate $\bar{J}(\cdot)$, at first we will assume that the observation activities of the sensors will occur prior to the operations that will be utilizing the updated information. Once we develop the observation control algorithm further below, we will discuss a case where the observation tasking is occurring in parallel with the associated operations.

While we have develop the theory for the general group of reconnaissance-based mobile sensing missions, in order to motivate the construction, we present the following example. Throughout this dissertation, we will come back to this example when necessary, and we will also simulate the developed theories on this setting. Consider the problem, depicted in Figure 2.3, where some Blue ground units will be moving through urban terrain along the dashed blue line. Pre-observation knowledge may indicate that there are Red fire-teams in some of the related buildings along their path. For example, it might be known that there is a Red fire-team in either Building 1 or Building 2 with associated probabilities. Similarly, it might be additionally known that there is one Red fire-team distributed among buildings 3–5, and also another again Red fire-team distributed among buildings 6–7 with some associated probabilities describing the likelihood of the various possible configurations. Meanwhile the Blue ground units have a single micro-UAV that they could deploy before starting to move through this region, and receive immediate observations about the possible presence of Red teams hiding in those buildings. After the UAV completes its mission, with the sensor returns pre-observation knowledge will be updated, and it will be utilized by the Blue ground units to make their own decisions when moving through the buildings. We assume that depending on their knowledge about the presence of Red fire-teams inside the buildings along their path, Blue ground units will decide on possible actions on their own, even though they will not be let to modify their

path. (Interestingly, current military doctrine does not seem to allow automated path re-planning of troop movement, based on observation returns. However, that point will be irrelevant to the construction to follow.) For example, Blue units will be given the option to lay cover fire on a particular building, or to remain “tight” during some step, meaning that they fire only if fired upon, depending on their knowledge. We assume these local actions are chosen from some finite set. We put this extra decision making layer to complete our example where UAV updated information will be utilized for another mission.

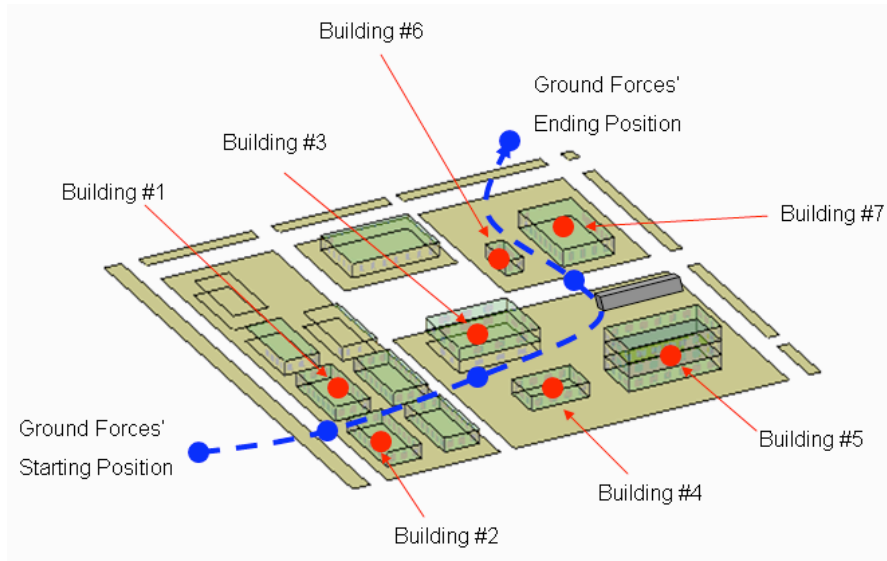


Figure 2.1: Blue COA

Now we turn our attention to develop the model in the general case. As mentioned earlier, we are assuming that the blue sensing operation will be a precursor to another succeeding blue operation. Since our theory is aimed at a general level, we consider the operation to follow as an abstract operation (as a black box), which will utilize the updated information from the sensing operations. We let ζ_t be the state of this operation at time t , and the finite set \mathcal{Z} its state space. Considering the two person game setting, we differentiate the states of the blue team from other states (such as states defining red team) defining the operation,

i.e.

$$\zeta_t \doteq \begin{bmatrix} \mathbb{H}_t \\ \mathbb{X}_t \end{bmatrix} \quad (2.3)$$

where we let $\mathbb{H}_t \in \mathcal{H}$ be the blue team's state at time t and $\mathbb{X}_t \in \mathcal{X}$ be states other than these states. Without loss of generality, we let $\mathcal{H} = [1..H]$, and similarly $\mathcal{X} = [1..X]$. (where, of course, each $c \in \mathcal{X}$ is an integer which actually indexes a specific configuration). Note that in this study, we do not assume that the non-blue states, \mathbb{X}_t , will be dynamic. For this reason, \mathbb{X}_t will be fixed, but unknown during time of the operations. For this reason we omit the time index from the variable \mathbb{X}_t , and use \mathbb{X} for the remainder of the study.

While in our argument we are assuming an abstract follow-up operation, in order to develop a feasible mathematical theory, we will assume that this operation will evolve as a controlled Markov process over the discrete time interval $\mathcal{T}^g \doteq [t_g..T^g]$. Considering the state, ζ_t , we introduced earlier one can define the state process as $\zeta : \mathcal{T}^g \rightarrow \mathcal{Z}$. Since \mathbb{X} is now only a random variable, the state transition will be based on \mathbb{H} . We suppose that at each time-step $t \in \mathcal{T}^{g-} \doteq [t_g..(T^g - 1)]$ we may select a control from finite set U_t^g . We let $\mathcal{P}_{\tilde{h},h}(\tilde{t}, u, x)$ denote the probability of transitioning from \tilde{h} to h at time-step $\tilde{t} \in \mathcal{T}^{g-}$, given control $u \in U_t^g$ and $x \in \mathcal{X}$, i.e.,

$$\mathcal{P}_{\tilde{h},h}(\tilde{t}, u, x) \doteq P(\zeta_{t+1} = \tilde{h}, \mathbb{X} = x \mid \zeta_t = h, t = \tilde{t}, u_t^g = u, \mathbb{X} = x) \quad (2.4)$$

where P denotes the probability measure. Correspondingly $\mathcal{P}(\tilde{t}, u, x)$ denotes the $H \times H$ matrix of transition probabilities. We suppose the initial state of \mathbb{H}_t is distributed according to $q_0^g \in S^H$ with components $[q_0^g]_h = P(\zeta_0 = h)$, where for any positive integer, N ,

$$S^N \doteq \left\{ q \in \mathbb{R}^N \mid q_j \in [0, 1] \forall j \in [1..N] \text{ and } \sum_{j=1}^N q_j = 1 \right\} \quad (2.5)$$

is the probability simplex. Following these definitions, stochastic dynamics for the random sequence \mathbb{H}_t could be written as

$$q_{t+1}^g = [\mathcal{P}(t, u_t^g, x)]^T q_t^g. \quad (2.6)$$

with $q_{t_g}^g = q_0^g$. We should note here that this form allows for feedback dependence on ζ_t ; for example, in some applications, it is useful to take u_t^g to be a vector of length H where the h^{th} row of \mathcal{P} depends only on the h^{th} element of u_t^g .

In the motivational example above, the state $h \in \mathcal{H}$ would correspond to a health state of the Blue ground entities, and $x \in \mathcal{X}$ would correspond to an unknown Red configuration. We might also typically take $q_0^g = I^1$ where $I_h^1 = 1$ if $h = h_0$ and $I_h^1 = 0$ if $h \neq h_0$ where state h_0 represents perfect health of all entities.

Back to the formulation of the general case, we let the payoff take the form of a terminal cost criterion, $C^g(\mathbb{H}_{T^g}, \mathbb{X})$. Let $\mathcal{U}^g \doteq U_{t_g}^g \times U_{t_g+1}^g \times \cdots \times U_{T^g-1}^g$. At the onset of the operation (i.e., after the observations), our knowledge of the likelihood of any configuration $x \in \mathcal{X}$ will be given by distribution q_0 . Note that $q_0 \in S^X$. For the operation, the expected payoff, $J^g : S^X \times \mathcal{U}^g \times S^H \rightarrow \mathbb{R}$, is given by

$$J^g(q_0, u^g; q_0^g) = \mathbf{E}[C^g(\mathbb{H}_{T^g}, \mathbb{X})], \quad (2.7)$$

where the expectation is over the initial state, the unknown aspects, and implicitly, the state transitions.

The cumulative state transition over \mathcal{T}^{g-} is given by

$$\overline{\mathcal{P}}(u^g, x) \doteq \mathcal{P}(t_g, u_{t_g}^g, x) \mathcal{P}(t_g + 1, u_{t_g+1}^g, x) \cdots \mathcal{P}(T^g - 1, u_{T^g-1}^g, x) \quad (2.8)$$

where we note that $\overline{\mathcal{P}}$ maps $\mathcal{U}^g \times \mathcal{X}$ into the space of $H \times H$ transition probability matrices. We see that the expected payoff is

$$J^g(q_0, u^g; q_0^g) = \sum_{x \in \mathcal{X}} \sum_{h \in \mathcal{H}} \left\{ [\overline{\mathcal{P}}^T(u^g, x) q_0^g]_h C^g(h, x) \right\} [q_0]_x. \quad (2.9)$$

Let

$$\begin{aligned} \gamma_x(u^g; q_0^g) &\doteq \sum_{h \in \mathcal{H}} [\overline{\mathcal{P}}^T(u^g, x) q_0^g]_h C^g(h, x) \\ &= [\widehat{C}^g(x)]^T \overline{\mathcal{P}}^T(u^g, x) q_0^g, \end{aligned}$$

where $\widehat{C}^g(x)$ is the vector of length H with components $[\widehat{C}^g(x)]_h = C^g(h, x)$. Then,

$$J^g(q_0, u^g; q_0^g) = [\gamma(u^g; q_0^g)]^T q_0 = \gamma(u^g; q_0^g) \cdot q_0. \quad (2.10)$$

The value of information q is thus

$$\begin{aligned}\bar{J}(q_0) &= \bar{J}(q_0; q_0^g) = \max_{u^g \in \mathcal{U}^g} J^g(q_0, u^g; q_0^g) \\ &= \max_{u^g \in \mathcal{U}^g} [\gamma(u^g; q_0^g) \cdot q_0].\end{aligned}\tag{2.11}$$

It is important to note that \bar{J} is a convex piecewise linear function of its argument, the probability distribution regarding the unknown aspects, q_0 . This form will be exploited in the closed-loop control analysis of this problem in Chapter 3. An example of a $\bar{J}(q)$ is depicted in Fig. 2.2.

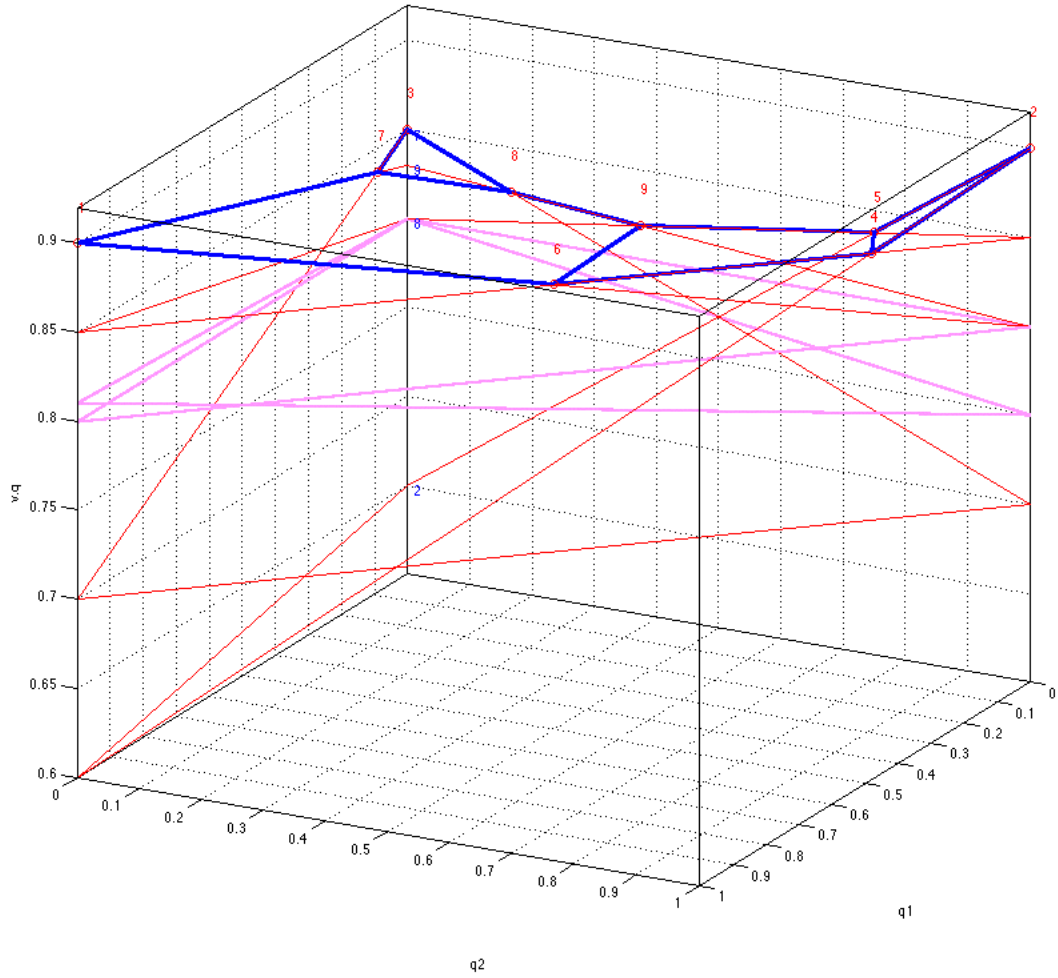


Figure 2.2: An example of $\bar{J}(q)$ (blue surface). Red planes are individual $v^i \cdot q$ defining $\bar{J}(q)$, whereas purple planes are inactive.

2.3 Observation Control Problem

In the above subsection, we obtained the form of the value of information through its effect on the associated operations. In this section, we will develop the observation tasking problem for a payoff of that form. Prior to development of this material, we again motivate the discussion with the military urban operations support example. Recall that there was some a priori information on the unknown Red configuration, and that this could be modeled as a probability distribution over some set of possible Red configurations, \mathcal{X} . Refer again to Figure 2.3. Suppose \mathcal{X} consisted of all possible Red configurations with one Red fire-team in either Building 1 or Building 2, and two additional Red fire-teams distributed across Buildings 3–7. One could, for example, have a uniform distribution over all of the possibilities for the positions of the three Red fire-teams. (A slight extension of \mathcal{X} would allow for less than three fire teams as well.) One could suppose that there were two sensing assets which could be tasked at each time-step. For example, one asset could be tasked to examine Building 2, and the other to examine Building 5. Further, one could suppose the possible observation returns were of the simple form that either a Red fire-team was spotted in the building or a Red fire-team was not spotted there. There would be probabilities of detection and missed detection (i.e., a confusion matrix). Then, based on the observation returns, one would update the probability distribution describing our knowledge, and if time allowed, re-task the sensing assets to obtain further data.

Now we develop the observation control problem in a bit more generality. We suppose the observation task will take place over a fixed, finite number of time-steps, $\mathcal{T}^o = [s..T^o]$ and also let $\mathcal{T}^{o,-} = [s..T^o - 1]$. At each $t \in \mathcal{T}^{o,-}$, the sensor(s) will make observations of a finite set of aspects of the configuration \mathcal{X} . Suppose, without loss of generality, that the set of possible sets of aspects is indexed by $\mathcal{L} = [1..L]$. That is, a state index, $l \in \mathcal{L}$ corresponds to a particular set of aspects to be observed in a time-step. The sensing assets state at time $t \in \mathcal{T}^{o,-}$ will be $\xi_t \in \mathcal{L}$. (We do not include technical complications such as sensor asset health state.) We let the observation control space be $\mathcal{U}^o = \mathcal{L}$, and take $\xi_{t+1} = u_t^o$ for all $t \in \mathcal{T}^{o,-}$. That is, we are allowing one-step transition from observing any set of aspects to

observing any other set of aspects. One could limit the possible next states to be dependent on the current state. (In the above example, this could correspond to not allowing a sensor to move from say, Building 1 to Building 7 in a single time-step.) This would only introduce additional notational and computational complexity to the current theory, and so we do not include that here. We suppose that while in state $\xi_{t+1} = u_t^o = l \in \mathcal{L}$, observation set $y_t = y \in \mathcal{Y}^l$ will be made, where $\mathcal{Y}^l = [y_1^l \dots y_{Y^l}^l]$ is finite. We suppose that one has conditional probability of observing $y \in \mathcal{Y}^l$ while observing aspects $l \in \mathcal{L}$ given $x \in \mathcal{X}$, denoted by $R_x^{y,l}$.

Let $R^{y,l}$ be the vector of length X with components $R_x^{y,l}$, and let $D^{y,l}$ be the $X \times X$ diagonal matrix with diagonal elements $D_{x,x}^{y,l} = R_x^{y,l}$. Then, given any sensing control action $u_t^o = \xi_{t+1} = l \in \mathcal{L}$ and resulting (random-variable) observation y_t , with Bayes' theorem one can write

$$q_{t+1} = \frac{1}{R^{y_t,l} \cdot q_t} D^{y_t,l} q_t \doteq \beta^{y_t,l}(q_t) = \beta^{y_t,l}(q_t; u_t^o), \quad (2.12)$$

which defines the stochastic dynamics for the state \mathbb{X} .

Now that we have completed our discussion of the value of information and the stochastic state dynamics we are ready to define the control formulations on top of these settings. We start with an open-loop controller. The payoff for information state q at any time t with observation-platform control u_t^o is

$$\mathcal{J}(s, q, u.) \doteq \mathbf{E} \{ \bar{J}(q_T) \} \quad (2.13)$$

where $u^o \doteq u_{[s, T^o-1]}^o = \{u_t^o \in \mathcal{U}^o \mid t \in \mathcal{T}^{o,-}\}$ and the propagation of the state from $q_s = q$ to q_T follows (2.12) with sequence of controls u^o . This kind of analysis would be appropriate for a concept-of-operations where incoming observational data could not be used to re-adjust the sensing-platform task plan. With such a model, the control problem reduces to an open-loop optimization problem. In particular, one solves for the value function

$$V^o(t, q) = \max_{u^o \in [\mathcal{U}^o]^{T^o-s}} \mathcal{J}(s, q, u.). \quad (2.14)$$

In the next section we give an example that solves the optimal control problem with this type of control. But before that we would like to define the state-feedback

approach. Now, let \mathcal{A}^s denote the set of non-anticipative feedback controls over $\mathcal{T}^{o,-}$. That is, we let

$$\mathcal{A}^s \doteq \left\{ \alpha_{[s, T^o-1]} : [S^N]^{T^o-s} \rightarrow [\mathcal{U}^o]^{T-s} \mid \text{if } q_r = \hat{q}_r \text{ for all } r \leq \bar{t} \text{ then } \alpha_r[q.] = \alpha_r[\hat{q}.] \text{ for all } r \leq \bar{t} \right\}$$

where $[\mathcal{U}^o]^{T^o-s}$ denotes the outer product of \mathcal{U}^o , $T-s$ times, and similarly with $[S^N]^{T-s}$. The payoff for information state $q_s = q$ and non-anticipative control $\alpha \in \mathcal{A}^s$ is

$$\mathcal{J}(s, q, \alpha.) \doteq \mathbf{E} \{ \bar{J}(q_T) \} \quad (2.15)$$

where the propagation of the state from $q_s = q$ to q_T follows (2.12) with control $u_t^o = \alpha_t[q.]$ at each time, $t \in \mathcal{T}^{o,-}$. The corresponding value function is:

$$V(s, q) = \sup_{\alpha. \in \mathcal{A}^s} \mathcal{J}(s, q, \alpha.). \quad (2.16)$$

Note that, from (2.15) and (2.16),

$$V(T, q) = \bar{J}(q), \quad (2.17)$$

and considering (2.11), one has

$$V(T, q) = \max_{u.^g \in \mathcal{U}^g} J^g(q, u.^g; q_0^g) = \max_{u.^g \in \mathcal{U}^g} [\gamma(u.^g; q_0^g) \cdot q]. \quad (2.18)$$

This formulation is particularly important for it will be utilized at the beginning of dynamic programming iterations, which are discussed in chapter 3.

2.4 Open Loop Analysis of the Urban Operation

Following our theoretical developments in the previous section, now we utilize the open-loop controller formulations to solve the optimal tasking formulation for the previously mentioned urban operation. We first define the details of this setting and present our results after.

2.4.1 Operation Setting

We suggest the reader to check Fig 2.3 to remember the physical setting for this problem where a team of blue ground units would be moving through an urban zone with a high possibility of coming under attack. In this scenario we assume that the blue team are given information that there are already 3 red fire-teams present in the region. Additionally, it is known that the first of these teams are based in either building 1 or building 2, the second team in building 2,4, or 5, and the last team in 6 or 7. Before the start of this mission, blue ground units will deploy their micro-UAV and will improve their knowledge about this setting following the observations. We need to find the optimal buildings the UAV should focus its sensors on to maximize the expected troop survivability at the end of the mission. Consistent with our earlier assumption of \mathbb{X}_t being time independent, we will assume that red units will stay on defense, and will hold their positions, that are unknown to the blue team. Also in this scenario, our other earlier assumption of having the follow-up operations evolve as a controlled Markov chain, can be interpreted as having the attrition on blue units at time t being independent of the expected attrition at other times, and we will model the payoff function accordingly.

Considering the given information we start our analysis by discretizing the whole map (\mathfrak{R}) into 3 subregions, each of which has exactly 1 enemy team. We enumerate this subregions as subregions 1, 2 and 3 according to the forward path of the ground units, i.e., starting with 1 for the initial one and 3 for the last. We form the the following sets that contain the building numbers at each subregion, $B^1 \doteq \{1, 2\}$, $B^2 \doteq \{3, 4, 5\}$, and $B^3 \doteq \{6, 7\}$. We let $B \doteq B^1 \cup B^2 \cup B^3$. For future reference, we also form vectors for each of these sets. We order the vectors considering the numbers associated with each building. For each $t \in \{1, 2, 3\}$, we let $\Delta^t \doteq [i_1, i_2, \dots, i_n]^T$ where $i_j < i_k$ for $j < k$ and $i_j \in B^t$, $i_k \in B^t$. We also define

$$\Delta \doteq \begin{bmatrix} \Delta^1 \\ \Delta^2 \\ \Delta^3 \end{bmatrix}$$

The discretized map is shown below in figure 2.3.

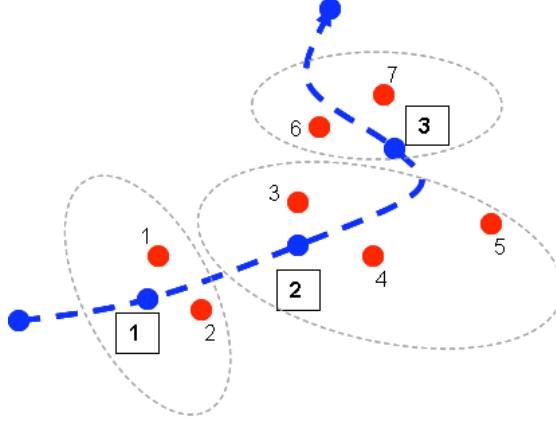


Figure 2.3: Discretized Blue Course of Action

In order to avoid multiple time scales, we also switch from real time to operation time, and in this setting we assume that one step of time will pass at each subregion. Also, without losing generality, in order to avoid extra time notations we will let the operation start at time $t = 1$. Following these, the operation time period will be $\mathcal{T}^{g-} = \{1, 2, 3\}$, and at each time the blue ground units will be at the subregion with the same number.

2.4.2 Formulation

As mentioned before, in this motivational problem, a possible state $h \in \mathcal{H}$ would correspond to the health state of the Blue entities. In this simulation in order to reduce computational complexity, we consider a small state space \mathcal{H} , for the state \mathbb{H} . We let $\mathcal{H} = \{1, 2\}$, with 1 corresponding to a killed state and 2 for a full healthy state.

As for the other state, \mathbb{X} , remember that we have earlier mentioned that it would correspond to unknown red configurations. Unlike the blue state \mathbb{H} , we will not consider health state of red units for defining \mathbb{X} . Instead, following a worst case scenario approach we will assume that the red entities have full health prior to the start of the game and will keep it throughout the operation. We let \mathbb{X} to be the unknown physical locations of red fire-teams. In this specific setting, since we have partitioned \mathfrak{R} into three subregions, we want to distinguish the red state

at one subregion from another. Following this reason, for this problem, we define \mathbb{X} as a random vector

$$\mathbb{X} = [\mathbb{X}^1 \ \mathbb{X}^2 \ \mathbb{X}^3]^T \quad (2.19)$$

For a 3 stage operation like this if we hadn't had any a priori knowledge about the location of red forces the range of \mathbb{X} would be the all possible locations of 3 red fire-teams, i.e., the set of triplets formed from the set of building numbers, $\bar{\mathcal{B}}^3$. Considering a priori knowledge, a smaller subset of $\bar{\mathcal{B}}^3$ could be defined for this purpose, i.e., $\mathbb{X} : \Omega \rightarrow \mathcal{B}^3$ where

$$\mathcal{B}^3 \doteq \{[a, b, c]^T \mid a \in B^1, b \in B^2, c \in B^3\}.$$

Furthermore, with an indexing function $f(\cdot)$ we can map enumerate all possible triplets with an index set \mathcal{N} , i.e., $f : \mathcal{B}^3 \rightarrow \mathcal{N}$ is 1-1 and onto and $\mathcal{N} \doteq \{1, 2, \dots, 12\}$. Here, $\#(B^1)\#(B^2)\#(B^3) = 12$. This way for $f(\mathbb{X})$ we can define the distribution q we defined earlier in this chapter such that $[q]_k = P(f(\mathbb{X}) = k) \ \forall k \in \mathcal{N}$. Notice that in this problem q is the joint distribution. At this point we also would like to define the probability of the presence of a red fire-team at a specific building. We define the probability of red unit presence for building $i \in B^1$ using the marginal probability formulation below.

$$p^i \doteq \sum_{j \in B^2, k \in B^3} P(\mathbb{X}^1 = i, \mathbb{X}^2 = j, \mathbb{X}^3 = k), \quad i \in B^1. \quad (2.20)$$

With a similar formulation, the probability of red presence for buildings in other subregions can be formulated as well. Notice that, following the a priori information that there is one fire-team at each subregion, these probabilities satisfy the following:

$$\begin{aligned} p^1 + p^2 &= 1 \\ p^3 + p^4 + p^5 &= 1 \\ p^6 + p^7 &= 1 \end{aligned}$$

Notice that the following vectors

$$q^1 = \begin{bmatrix} p^1 \\ p^2 \end{bmatrix}, \quad q^2 = \begin{bmatrix} p^3 \\ p^4 \\ p^5 \end{bmatrix}, \quad q^3 = \begin{bmatrix} p^6 \\ p^7 \end{bmatrix} \quad (2.21)$$

act as distributions to \mathbb{X}^1 , \mathbb{X}^2 , and \mathbb{X}^3 . We will consider these vectors in a moment.

As mentioned earlier we assume that blue ground units will utilize updated information in their own actions. For this scenario we model such actions in the following way. When blue units are passing through a subregion by considering the probability distribution of the enemy they can either lay cover on one of the buildings or hold tight (only fire if fired on). Following this model, at time $t \in \mathcal{T}^{g-}$, $u_t^g \in \bar{B}^t \doteq B^t \cup \{0\}$. Notice that we used 0 to denote the control option of remaining “tight”.

Related to ground units decisions, we will define the value of information for this game as the expected survivability through the operation in region \mathfrak{R} . Related to this function, we define the state (health) transition matrix for the blue team in the following form.

$$\mathcal{P}(t, u, x) = \begin{bmatrix} 1 & 0 \\ k(t, u, x) & 1 - k(t, u, x) \end{bmatrix} \quad (2.22)$$

where we define $k(t, u, x)$ as the probability of kill against blue units given state x , ground team action u at time t . Remember that the health transition matrix $\mathcal{P}(t, u, x)$ corresponds to attrition happening at the t^{th} stage of the operation, and also that the attrition on ground units at one stage will not be influenced by the states of the other stages. For this reason, when defining $\mathcal{P}(t, u, x)$, it is sufficient to only considering the state space of X^t , i.e., x will take values in B^t .

Notice that the heath transition matrix defined above has the computational advantage that one only needs to consider $k(u, x)$ in order to define the cumulative state transition matrix. For example, for a two step operation the cumulative state transition matrix, \bar{P} , would be,

$$\begin{aligned} \bar{P}(\{u_{t_1}^g, u_{t_2}^g\}, x) &= \begin{bmatrix} 1 & 0 \\ k(t_1, u_1^g, x) & 1 - k(t_1, u_1^g, x) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ k(t_2, u_2^g, x) & 1 - k(t_2, u_2^g, x) \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ k(t_1, u_1^g, x) + (1 - k(t_1, u_1^g, x)) k(t_2, u_2^g, x) & (1 - k(t_1, u_1^g, x)) (1 - k(t_2, u_2^g, x)) \end{bmatrix} \end{aligned}$$

In the final matrix above, the probability of blue unit survivability (healthy state ending at healthy state) after two steps is given at 2nd row 2nd column, $(1 - k(t_1, u_1^g, x)) (1 - k(t_2, u_2^g, x))$. It is, in fact, the product of 2nd row 2nd column entries of probability transition matrices for u_1^g and u_2^g . For this reason, if one wants to find the survivability of blue units, in order to reduce the computational efforts, he can only work on and store this term rather than the full matrix. For its importance we will denote this term through a vector. For this purpose, as a first step, we need to define an index sets for these vectors. We define sets, $\mathcal{I}^1 \doteq [1.. \#(\bar{B}^1)]$, $\mathcal{I}^2 \doteq [(\#(\bar{B}^1)+1).. \#(\bar{B}^1) + \#(\bar{B}^2)]$, and $\mathcal{I}^3 \doteq [\#(\bar{B}^1) + \#(\bar{B}^2).. I]$, where $I \doteq \#(\bar{B}^1) + \#(\bar{B}^2) + \#(\bar{B}^3)$. We also let,

$$\mathcal{I} \doteq \bigcup_{t \in \mathcal{T}^g} \mathcal{I}^t \quad \text{and} \quad \mathcal{I}^t \cap \mathcal{I}^{\bar{t}} = \emptyset \quad \text{for } t \neq \bar{t} \in \mathcal{T}^g. \quad (2.23)$$

Now, we define a functional $\mathcal{M}[t](\cdot) : \bar{B}^t \rightarrow \mathcal{I}^t$, 1-1 and onto, such that $\mathcal{M}[t](i) < \mathcal{M}[t](j)$ for $i < j$, $i \in B^t, j \in B^t \forall t \in \mathcal{T}^g$. Following this, we define the vector

$$v^j \doteq \begin{bmatrix} 1 - k(t, u, [\Delta^t]_1) \\ \vdots \\ 1 - k(t, u, [\Delta^t]_n) \end{bmatrix} \quad (2.24)$$

where $j = \mathcal{M}[t](u)$, $j \in \mathcal{I}^t$. Remember that Δ^t is the vector containing the building numbers for subregion t . For example

$$v^2 = \begin{bmatrix} 1 - k(1, 2, 1) \\ 1 - k(1, 2, 2) \end{bmatrix} \quad (2.25)$$

where we have let $\mathcal{M}_1[2] = 2$. To make our point more clear, we present the following diagram about formation of the index vector \mathcal{I} .

$$\mathcal{I} = \begin{bmatrix} 1 \\ \vdots \\ 10 \end{bmatrix} = \begin{bmatrix} \mathcal{I}^1 \\ \mathcal{I}^2 \\ \mathcal{I}^3 \end{bmatrix} \begin{array}{l} \longleftarrow \bar{B}^1 \\ \longleftarrow \bar{B}^2 \\ \longleftarrow \bar{B}^3 \end{array}$$

Now, suppose that the vector v^j corresponds to an action at time t , and remember that time t corresponds to subregion t . For this reason, it should be noted that this newly defined vector contains enough information to define the dynamics at subregion t . Said in a different way, the payoff of information (maximum expected survivability) at subregion t could be defined as

$$\max_{i \in \mathcal{I}^t} \{v_t^i \cdot q^t\} \quad (2.26)$$

where $j = \mathcal{M}[t](u)$, $j \in \mathcal{I}^t$ for some $u \in \bar{B}^t$. By introducing this vector in $\mathfrak{R}^{\#(B^t)}$ notice that we are avoiding the need to store 3 (2x2) matrices. Now, remember our earlier general assumption that the operations following the sensing actions would evolve as a controlled Markov process. This assumption has let us to think of the attrition on blue units on a subregion independent of other attrition on other subregions. Following this reasoning, the total payoff for this operation could be formulated as the product of expected survivability at each subregion, i.e;

$$\bar{J}(q) = \prod_{k=1}^3 \left[\max_{i \in \mathcal{I}^k} \{v_k^i \cdot q^k\} \right] \quad (2.27)$$

At this point, we also want to mention our formulation of the observation control problem for this scenario. For this purpose, we first need to determine the control space, \mathcal{U}^o for the sensing asset. Remember that in our early model in section 2.3 we let $\mathcal{U}^o = \mathcal{L}$ and \mathcal{L} was the set of possible configurations for \mathbb{X} . In this scenario since the set of possible configurations for \mathbb{X} is \mathcal{B}^3 we could consider it as a possible candidate for \mathcal{L} . Meanwhile since we assume we have single UAV available we pick the set B . We assume that the micro-UAV utilized in this scenario has the limited capability of making two observations at each state $l \in \mathcal{L}$, ‘See’ and ‘No See’, which as the names suggests correspond to seeing a red fire-team, or not seeing a red fire-team at building l , respectively. Following this assumption, we define $\mathcal{Y} \doteq \{1, 2\}$ with 1 corresponding to the observation result ‘See’ and 2 to ‘No See’. Notice that $\mathcal{Y}^l = \mathcal{Y}$, $\forall l \in \mathcal{L}$.

Now, we will update given information according to (2.12). In order to utilize this equation we need to define conditional probability vectors $R^{y,l}$ for all possible observations y and for all possible states l . Remember that we have defined $R_x^{y,l}$ as the conditional probability of observing $y \in \mathcal{Y}^l$ while focusing on $l \in \mathcal{L}$ given $x \in \mathcal{X}$. To reduce the burden of defining $R^{y,l}$ for all possible observations and all possible states, we will define the following two conditional probability variables α and β , and will define $R^{y,l}$ according to them afterwards.

$$\alpha \doteq P(y_t = 1, \text{ for } u_t^o = l \mid \mathbb{X} = [x_1 x_2 x_3]^T, \exists i \in \mathcal{T}^{g-} x_i = l) \quad (2.28)$$

$$\beta \doteq P(y_t = 2, \text{ for } u_t^o = l \mid \mathbb{X} = [x_1 x_2 x_3]^T, \nexists i \in \mathcal{T}^{g-} x_i = l) \quad (2.29)$$

Basically, expressed in words, α is the conditional probability of seeing an enemy at a building given an enemy is there, and β is not seeing an enemy at a building given an enemy is not there. We will assume that these conditional probabilities will be constant for any building (i.e., for any $l \in \mathcal{L} = B$). At this point, notice that when the micro-UAV is sent to make an observation at building l , notice that the conditional probabilities of observing an enemy or not observing an enemy only depends on the state of the subregion containing l , and is independent on other subregions' states. Following this reason when considering to update the distribution of \mathbb{X} , q , following a sensor observation at subregion t , it would be sufficient to work with the distribution of \mathbb{X}^t , q^t . This could also save us computation time by working with a smaller size vector, $R^{y,l}$. The update of q^t also follows (2.12). Then when UAV is sent to make an observation on building l , which is at subregion k , we update q^k but leave other subregional distributions unchanged. For example, for $u_t^o = 2$, and $y_t = 1$ one should update q^1 by (2.12) where

$$R^{2,1} = \begin{bmatrix} (1 - \beta) & \alpha \end{bmatrix} \quad (2.30)$$

We present our results using to this formulations in the next section.

2.4.3 Results

Before presenting our results about (2.14) the optimal sensor tasks that improved the excepted ground unit survivability the most, we also would like to

present a few figures about the ground unit actions that we have mentioned so much, and clear any possible ambiguity.

For this purpose consider the first subregion of this urban operation scenario. Assume that the health transition matrices $\mathcal{P}(t = 1, u, x)$ are defined as the following.

$$\begin{aligned} \mathcal{P}(1, 1, 1) &= \begin{bmatrix} 1 & 0 \\ 0.1 & 0.9 \end{bmatrix} & \mathcal{P}(1, 2, 1) &= \begin{bmatrix} 1 & 0 \\ 0.4 & 0.6 \end{bmatrix} & \mathcal{P}(1, 0, 1) &= \begin{bmatrix} 1 & 0 \\ 0.15 & 0.85 \end{bmatrix} \\ \mathcal{P}(1, 1, 2) &= \begin{bmatrix} 1 & 0 \\ 0.5 & 0.5 \end{bmatrix} & \mathcal{P}(1, 2, 2) &= \begin{bmatrix} 1 & 0 \\ 0.15 & 0.85 \end{bmatrix} & \mathcal{P}(1, 0, 2) &= \begin{bmatrix} 1 & 0 \\ 0.35 & 0.65 \end{bmatrix} \end{aligned}$$

Now we want to see how the ground troops make their choices as a function of q^1 . Since $q^1 = [p^1 \ p^2]^T$ and $p^2 = 1 - p^1$, it would be sufficient to draw the results with respect to p^1 on a one dimensional graph rather than two. The results are shown on figure 2.4.

As could be expected when p^1 is closer to 1, that is the case when the possibility of the red fire team being at building 1 is very likely, ground troops opt to lay cover fire on building 1. Similarly when p^1 is closer to 0, that is the case when the possibility of the red fire team being at building 1 is very unlikely, ground troops opt to lay cover fire on building 2. The worst situation is happening when p^1 is in the vicinity of 0.5, when the uncertainty is highest, and the ground units opt to stay tight.

In our analysis we didn't solve the open loop control of the sensor tasking problem for a specific distribution q but we rather discretized the whole probability simplex into smaller elements and analyzed the sensor problem on this discretized simplex. For each element in this discretized simplex we considered all possible sensor controls (buildings in this case) and all possible observation possibilities (see or no see). We updated the each distribution utilizing (2.12) and found a posterior distributions. As we considered our possible observations, y_t as random variables, we calculated their respective probabilities as well for each UAV control. Finally we found the payoff for an initial distribution q with (2.27).

The additive inverse of the value function for the open-loop case is depicted below in figure 2.5, as a function of the initial information in subregions 1 and

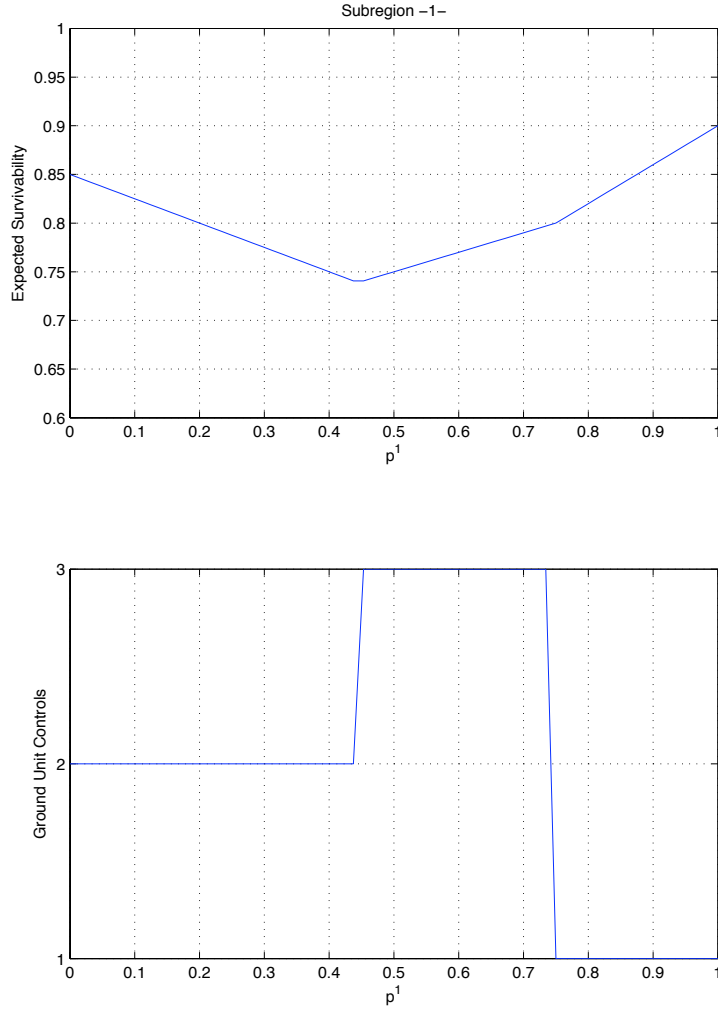


Figure 2.4: Expected ground unit survivability and unit decisions in subregion 1 as a function of p^1 .

3. (Note that the information state is minimally stored as a vector in the four-dimensional unit hypercube, and so we only display it over two components—the probabilities that there is a red entity in building 1 and the probability there is a red entity in building 6.

We compared this with a heuristically generated sensing-platform task plan, which for any specific q , might be similar to what a commander would choose. The expected payoff for the heuristic task planner is depicted in 2.6, and the percent improvement (in terms of reduced attrition) is depicted in 2.7.

Notice that after switching to the heuristic controller the expected attrition

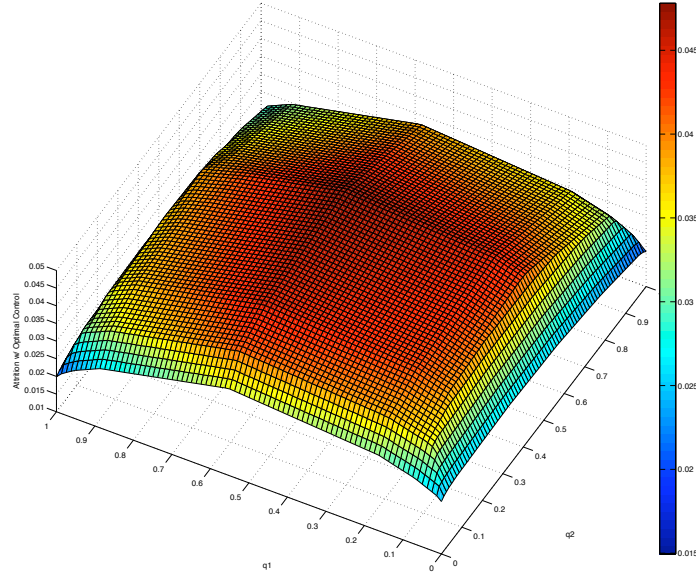


Figure 2.5: Expected attrition on ground units following optimal UAV support. (On this figure axis $q1$ corresponds to p^1 and axis $q2$ corresponds to p^6).

on blue units was increased between %0 to %40. Notice the change of attrition values around the edges.

Overall, in this scenario, the proposed sensor tasking formulation was able to reduce the expected attrition (increase the expected survivability) significantly, especially more significantly for some particular distributions. While our example was simple and had only 3 stages, the expected gains shows that it might be good alternative to automated decision making algorithms for the sensor platforms. The drawback of this approach is its incapability to incorporate the current states of the system, and its dependence on pre-operation knowledge. Still it should be considered for situation when state feedback might not be available to the controllers. We complete our discussion of the open-loop control here and proceed to the closed loop formulation.

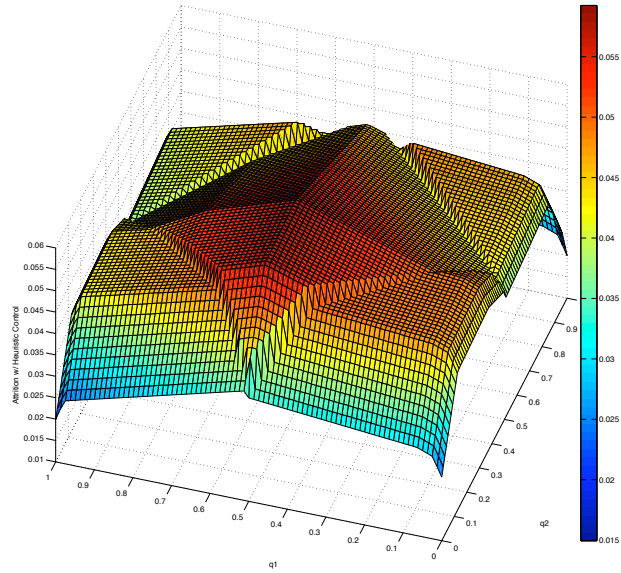


Figure 2.6: Expected attrition on ground units following heuristic UAV support. (On this figure axis $q1$ corresponds to p^1 and axis $q2$ corresponds to p^6).

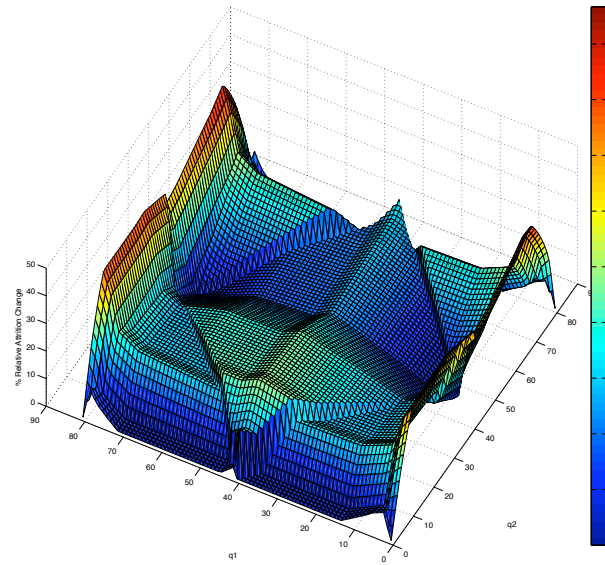


Figure 2.7: Expected percent change in attrition when switched from optimal UAV support to heuristic UAV support. (On this figure axis $q1$ corresponds to p^1 and axis $q2$ corresponds to p^6).

Chapter 3

The State Feedback Control Problem

In this chapter we analyze the optimal sensor tasking problem from the state feedback control perspective. As opposed to the open-loop control state feedback control delivers better performance by inheriting the advantages of closed-loop control. Meanwhile, the computational efforts to fully analyze, and solve the control problem of (2.16) is more challenging than the open-loop case. However, as we will show in the next section, the special linear form of (2.18) will give us an opportunity to exploit this special form in the solution of the optimal control problem, and avoid numerical challenges. We start our discussion with formulating the solution to the sensor tasking problem.

3.1 Solution of the State Feedback Control Problem with Dynamic Programming

In this section we solve the feedback control problem of (2.16) utilizing dynamic programming. In order to give a complete discussion, at first we briefly mention some of the dynamic programming results.

Proposition 3.1.1. *Principle of Optimality:*

An optimal policy has the property that whatever the initial state and initial decision

are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision. [Bel57]

The principle of optimality could be captured in a mathematical form considering the payoff function, $\mathcal{J}(\cdot)$, and the value function $V(\cdot)$ of the previous chapter. For a discrete time dynamical system, when analyzing a finite time horizon problem, the above proposition could be expressed as in the following theorem.

Theorem 3.1.2. *Principle of Optimality for Discrete-Time Fine-Time Horizon Problems:*

For the a system whose dynamics are governed as,

$$\begin{aligned}\xi_{t+1} &= f(\zeta_t, u_t, w_t) \\ \xi_t &= x\end{aligned}$$

one can formulate the value function,

$$V(t_0, x) = \inf_{\alpha \in \mathcal{A}^{t_0}} J(t_0, x, \alpha) \quad (3.1)$$

$$V(t_0, x) = \inf_{\alpha \in \mathcal{A}^{t_0}} \mathbf{E}|_{\zeta_s=x} \left[\sum_{t=t_0}^{t_1-1} \mathcal{L}(\zeta_t, \mu_t) + V(t_1, \zeta_{t_1}) \right] \quad (3.2)$$

This theorem could be put into another form that would let us solve the optimal state feedback problem.

$$V(t_o, x) = \inf_{u \in \mathcal{U}} [\mathcal{L}(x, u) + E[V(t_o + 1, f(x, u_{t_o}, w_{t_o}))]] \quad (3.3)$$

We will refer the above equation as the dynamic programming equation (which is also known as the Bellman equation), or shortly DP throughout this dissertation.

Now that we have introduced the DP formulation, we can utilize it to develop the solution to the feedback control of the optimal sensor tasking problem. The following theorem highlights this.

Theorem 3.1.3. *For $t \in \{0, 1, \dots, T-1\}$,*

$$V(t, q) = \max_{u_t^o \in \mathcal{R}} \mathbf{E}_{y_t} \{V(t+1, \beta^{y, u_t^o}(q))\} \quad (3.4)$$

where the expectation is over the set of possible observations.

Proof. The cost function $J(\cdot)$ we developed in the previous chapter only had terminal cost and no running cost, i.e, $\mathcal{L}(x, u) = 0$. Also we have defied the value function as a function of information state q rather than x . The propagation of the distribution q , was formulated in terms of the function β in (2.12). \square

Now, we indicate how the backward DP is mechanized for our own problem we formulated in (2.17). Starting from the terminal time, T ,

$$V(T, q) = \bar{J}(q).$$

Next, note that from (3.4)

$$V(t, q) = \max_{u_t^o \in \mathcal{U}} \sum_{y_t \in \mathcal{Y}^{u_t^o}} P(y_t) V(t+1, \beta^{y_t, u_t^o})$$

and further expanding this equation considering definitions of β^{y_t, u_t^o} and $P(y_t)$.

$$V(t, q) = \max_{u_t^o \in \mathcal{U}} \sum_{y_t \in \mathcal{Y}^{u_t^o}} \left\{ [R^{y_t, u_t^o} \cdot q] V \left(t+1, \frac{1}{R^{y_t, u_t^o} \cdot q} D(R^{y_t, u_t^o} q) \right) \right\} \quad (3.5)$$

Considering the formulations above one can notice that the solution of the feedback control is demanding. In order to compute $V(t, q)$, one must have $V(t+1, \cdot)$ on S^X . One way to approach and solve the feedback problem, (3.4), could be through numerical methods via discretization of the probability simplex. Especially for small scale settings, such as small building numbers etc, this type of approach might be a reasonable option for the field commander to find the optimal UAV paths. Meanwhile, one can notice that for larger values of $X = \#(\mathcal{X})$, the dynamic programming computations would become computationally infeasible when performed over the discretized probability simplex (grid-based methods), even for short time spans. To notice the numerical burden of calculations consider the following example. Let $L = 3$, the information state being defined on S^3 , and suppose that we'll be discretizing each axis into $2^5 = 32$ elements. For this setting the terms inside the bracket in (3.5) will be recalculated for a total of 6,144 times. This number is definitely unacceptable for such a low scale example, not to mention the possible numerical errors associated with discretization.

Meanwhile, the special form of $V(t, q)$ inherited from $\bar{J}(q)$ can be exploited to avoid this problem. That is, from (2.18), $V(T, q)$ takes the form

$$V(T, q) = \max_{i \in \mathcal{I}} (v^i \cdot q). \quad (3.6)$$

Here we let $\mathcal{I} = \mathcal{U}^g$ for notational simplicity in the upcoming sections, and again for the same reason use the notation v^i to represent the vectors $\gamma(u^g; q_0^g)$ in the sequel.

If we can show that this form is retained under the dynamic programming propagation, then we will be able to work with the v^i vectors instead of a discretized form of $V(t, q)$ over the probability simplex. In order to demonstrate this, we first introduce the following notation. For any set, \mathcal{I} , and positive integer M , let $\mathcal{P}^M(\mathcal{I})$ denote the set of all sequences of length M with elements from \mathcal{I} . (Note that the cardinality of $\mathcal{P}^M(\mathcal{I})$ is $(\#\mathcal{I})^M$.) Also in order to simplify our problem notation, we take $\mathcal{Y} = \{1, 2, \dots, N_y\}$, and define the general control set as $\mathcal{U} \doteq [1..N_u]$.

Theorem 3.1.4. *Suppose $V(t+1, q)$ takes the form*

$$V(t+1, q) = \max_{i \in \mathcal{I}_{t+1}} v_{t+1}^i \cdot q$$

where $\mathcal{I}_{t+1} = [1..I_{t+1}]$. Then,

$$V(t, q) = \max_{i \in \mathcal{I}_t} v_t^i \cdot q \quad (3.7)$$

where $\mathcal{I}_t = [1..I_t]$, $I_t = N_u(I_{t+1})^{N_y}$, and

$$v_t^i = \sum_{y_t \in \mathcal{Y}} D^{y_t, u_t^o} v_{t+1}^{j_{y_t}}, \quad (3.8)$$

where $(u_t^o, \{j_{y_t}\}) = \mathcal{M}^{-1}(i)$, and \mathcal{M} is a one-to-one, onto mapping from $\mathcal{U} \times \mathcal{P}^{N_y}(\mathcal{I}_{t+1}) \rightarrow \mathcal{I}_t$ (i.e., an indexing of $\mathcal{U} \times \mathcal{P}^{N_y}(\mathcal{I}_{t+1})$).

Proof. Using the assumption and Theorem 3.1.3, one can write:

$$\begin{aligned}
V(t, q) &= \max_{u_t^o \in \mathcal{U}} \sum_{y_t \in \mathcal{Y}} V(t+1, \beta^{u_t^o, y_t}(q)) Pr(y_t) \\
&= \max_{u_t^o \in \mathcal{U}} \sum_{y_t \in \mathcal{Y}} \max_{i \in \mathcal{I}_{t+1}} [v_{t+1}^i \cdot \beta^{u_t^o, y_t}(q)] Pr(y_t) \\
&= \max_{u_t^o \in \mathcal{U}} \sum_{y_t \in \mathcal{Y}} \max_{i \in \mathcal{I}_{t+1}} \left[v_{t+1}^i \cdot \frac{D^{y_t, u_t^o} q}{R^{y_t, u_t^o} \cdot q} \right] R^{y_t, u_t^o} \cdot q \\
&= \max_{u_t^o \in \mathcal{U}} \sum_{y_t \in \mathcal{Y}} \max_{i \in \mathcal{I}_{t+1}} \left[v_{t+1}^{i^T} \frac{D^{y_t, u_t^o} q}{R^{y_t, u_t^o} \cdot q} R^{y_t, u_t^o} \cdot q \right] \\
&= \max_{u_t^o \in \mathcal{U}} \sum_{y_t \in \mathcal{Y}} \max_{i \in \mathcal{I}_{t+1}} [v_{t+1}^i \cdot D^{y_t, u_t^o} q]. \tag{3.9}
\end{aligned}$$

The next step is most easily seen using the max-plus algebra notation, where we note that the max-plus algebra is the commutative semifield over $\mathbb{R} \cup \{-\infty\}$ with operations $a \oplus b = \max\{a, b\}$ and $a \otimes b = a + b$ (c.f., [FBQ92], [CG79], [KM97], [McE06]). Using this notation, we have

$$V^{o,f}(t, q) = \bigoplus_{u_t^o \in \mathcal{U}} \bigotimes_{y_t \in \mathcal{Y}} \bigoplus_{i \in \mathcal{I}_{t+1}} [v_{t+1}^i \cdot D^{y_t, u_t^o} q]$$

which, using the max-plus distributive property,

$$= \bigoplus_{u_t^o \in \mathcal{U}} \bigoplus_{\{i_{y_t}\} \in \mathcal{P}^{N_y}(\mathcal{I}_{t+1})} \bigotimes_{y_t \in \mathcal{Y}} [v_{t+1}^{i_{y_t}} \cdot D^{y_t, u_t^o} q]$$

where $\mathcal{P}^{N_y}(\mathcal{I}_{t+1}) = \{ \{i_{y_t}\}_{y_t \in \mathcal{Y}} \mid i_{y_t} \in \mathcal{I}_{t+1} \forall y_t \in \mathcal{Y} \}$. Returning to our previous notation, this is:

$$\begin{aligned}
V(t, q) &= \max_{u_t^o \in \mathcal{U}} \max_{\{i_{y_t}\} \in \mathcal{I}_{t+1}^{N_y}} \sum_{y_t \in \mathcal{Y}} [v_{t+1}^{i_{y_t}} \cdot D^{y_t, u_t^o} q] \\
&= \max_{u_t^o \in \mathcal{U}} \max_{\{i_{y_t}\} \in \mathcal{I}_{t+1}^{N_y}} \left[\sum_{y_t \in \mathcal{Y}} v_{t+1}^{i_{y_t}^T} D^{y_t, u_t^o} \right] q.
\end{aligned}$$

Note that D^{y_t, u_t^o} is symmetric, and manipulating the dot product one obtains:

$$V(t, q) = \max_{u_t^o \in \mathcal{U}} \max_{\{i_{y_t}\} \in \mathcal{I}_{t+1}^{N_y}} \left[\sum_{y_t \in \mathcal{Y}} D^{y_t, u_t^o} v_{t+1}^{i_{y_t}} \cdot q \right].$$

Now we proceed to reindex. We first define the new integer index set, $\mathcal{I}_t \doteq]1, I_t[$ where $I_t = (I_{t+1})^{N_y} N_u$ and $I_{t+1} \doteq \#\mathcal{I}_{t+1}$. This new set, \mathcal{I}_t , may be viewed as

composed of disjoint subsets, $\mathcal{I}_t^{u_t^o}$, each of which is related to a particular sensor control, $u_t^o \in \mathcal{U}$, i.e:

$$\mathcal{I}_t = \bigcup_{u_t^o \in \mathcal{U}} \mathcal{I}_t^{u_t^o} \quad \text{and} \quad \mathcal{I}_t^{u_t^o} \cap \mathcal{I}_t^{\bar{u}_t^o} = \emptyset \quad \text{for } u_t^o \neq \bar{u}_t^o \in \mathcal{U}$$

Now, we define a functional $\mathcal{M}_t[u_t^o] : \mathcal{I}_{t+1}^{N_y} \rightarrow \mathcal{I}_t^{u_t^o}$, 1-1 and onto, such that the following ordering holds. $\mathcal{M}_t[u]\{i_{y_t}\} < \mathcal{M}_t[\bar{u}]\{i_{y_t}\}$ for $u < \bar{u}$, $u \in \mathcal{U}, \bar{u} \in \mathcal{U}$. Then one can formulate $V(t, q)$ as:

$$V(t, q) = \max_{u_t^o \in \mathcal{U}} \max_{k \in \mathcal{I}_t^{u_t^o}} \{v_t^k \cdot q\}$$

where,

$$v_t^k = \sum_{y_t \in \mathcal{Y}} D^{y_t, u_t^o} v_{t+1}^{i_{y_t}} \quad \text{and} \quad k = \mathcal{M}_t[u_t^o](\{i_{y_t}\}).$$

Combining the two maxima, we find

$$V(t, q) = \max_{k \in \mathcal{I}_t} v_t^k \cdot q.$$

□

We now develop some helpful notation. For any t , let $\mathcal{V}_t \doteq \{v_t^i \mid i \in \mathcal{I}_t\}$. Then, by Theorem 3.1.4, the dynamic program can equivalently be given as

$$(\mathcal{V}_t, \mathcal{I}_t) = \mathcal{D}^{\mathcal{U}}[(\mathcal{V}_{t+1}, \mathcal{I}_{t+1})],$$

where the operator, $\mathcal{D}^{\mathcal{U}}$ is defined by the propagation (3.8). Also, we can denote the reconstruction of $V(t, \cdot)$ from the pair $(\mathcal{V}_t, \mathcal{I}_t)$ as $V(t, \cdot) = \mathcal{C}[(\mathcal{V}_t, \mathcal{I}_t)]$, where the reconstruction operator is given by (3.7).

Using Theorem 3.1.4, the numerical burden of grid-based analysis of $V(t, q)$ on the probability simplex is now avoided, and one only needs to propagate the vectors v_t^i backwards in time using (7). This can yield a significant reduction in the computation time, especially for settings where the state space is large. However, when the DP was performed for large time spans, the computation speed was observed to still be too slow relative to what would be required for real-time UAV operations. The remaining difficulty was the growth of the set \mathcal{I}_t at each

iteration. Remember that, $\mathcal{I}_t \doteq [1..I_t]$ where $I_t = (I_{t+1})^{N_y} N_u$ and $I_{t+1} \doteq \#\mathcal{I}_{t+1}$. In this equation, the growth of I_t as a power function is source of problem.

To see this issue, consider the particular example we have considered earlier on S^3 . With the newly proposed method, to find $V(T-1, q)$ from $V(T, q)$ a number of **48** v_{T-1}^i vectors needs to be found using (7). This number is no doubt a big improvement compared to **6,144** iterations earlier. However; continuing to 2^{nd} and 3^{rd} time steps, to find $V(T-2, q)$ and $V(T-3, q)$ from $V(T, q)$ first a total number of **6912** and then an incredible amount of **143,327,237** vectors would needed to be calculated and also stored in the computer memory. The following diagram shows the growth of I_t .

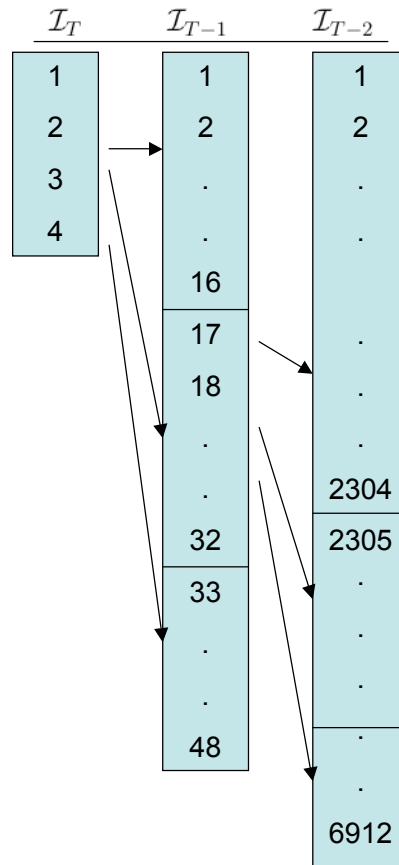


Figure 3.1: Growth of \mathcal{I}_T

We will address the means that may be used to attenuate this problem in

Chapter 4. But before that we would like find out possible extensions of 3.1.4. One particular case is the formulation we had earlier for the multi subregion case in section 2.4. Remember that in order to reduce computational complexity we had switched from the joint distribution, q , to subregional distributions, q^1 , q^2 , and q^3 and later we have formulated the payoff function as a product of maximums. In the next section, we will analyze similar multi-subregion operations, and will try determine if the form of value function could also be preserved in such cases.

3.2 Extension of Theorem 3.1.4 to Other Possible Cases

As mentioned earlier, the previous section was based on the assumption that a single mobile sensor would be deployed on a region with a single subregion. For this reason one can wonder, if a similar result to the one we mentioned in theorem 3.1.4 could be found in other possible scenarios. For this purpose we analyze two other possible cases. The first analysis considers the situation of N UAVs being deployed to a region consisting of N subregions, while the second one considers a single UAV being deployed to a region of N subregions. Before analyzing these cases, we again consider the assumption that the value of information regarding different subregions will be independent from each other. For example, in our earlier urban operation example we have defined the value of information as the expected survivability of blue ground units. For that scenario, our new assumption considers the expected survivability of blue ground units in a particular subregion independent of the expected survivability in other subregions. Also, in a similar way, we assume that the observations in different subregions will be independent of each other. This assumption is consistent with our earlier open loop formulations in section 2.4.

3.2.1 N UAVs for a Region of N Subregions

Before strating our analysis we present the following mathematical identities which will be utilized later.

Lemma 3.2.1. : *Consider two finite sets F and G , both of which are subsets of \mathfrak{R} , and have their elements indexed by index sets I_F and I_G . Then,*

$$\max_{(i,j) \in I_F \otimes I_G} \{|f_i| |g_j|\} = \max_{i \in I_F} |f_i| \max_{j \in I_G} |g_j|$$

Proof.

$$\text{Let } \bar{f} \doteq \max_{i \in I_F} |f_i| \quad \text{and} \quad \bar{g} \doteq \max_{i \in I_G} |g_i|$$

Then, since

$$\begin{aligned} |f_i| |g_j| &\leq \bar{f} \bar{g} \quad \forall (i, j) \in I_F \otimes I_G, \\ \max_{(i,j) \in I_F \otimes I_G} \{|f_i| |g_i|\} &\leq \bar{f} \bar{g}. \end{aligned}$$

Also, by the definition of the max operator,

$$\begin{aligned} \max_{(i,j) \in I_F \otimes I_G} \{|f_i| |g_i|\} &\geq |f_k| |g_l| \quad \forall (k, l) \in I_F \otimes I_G, \\ \max_{(i,j) \in I_F \otimes I_G} \{|f_i| |g_i|\} &\geq \bar{f} \bar{g}. \end{aligned}$$

Considering two inequalities,

$$\max_{(i,j) \in I_F \otimes I_G} \{|f_i| |g_i|\} = \bar{f} \bar{g} = \max_{i \in I_F} |f_i| \max_{j \in I_G} |g_j|$$

□

Corollary 3.2.2. : *Consider two finite sets F and G , both of which are subsets of $\mathfrak{R}^+ \doteq \{x \in \mathfrak{R} | x \geq 0\}$, and have their elements indexed by index sets I_F and I_G . Then,*

$$\max_{i \in I_F} \{f_i\} \max_{j \in I_G} \{g_j\} = \max_{(i,j) \in I_F \otimes I_G} \{f_i g_j\}$$

Proof. Follows the lemma above.

□

Now suppose that the operation following the sensing operations will go through K different subregions. Similar to our 3 stage problem we solved earlier, we enumerate each subregion with a positive integer from 1 to K . As an initial assumption we assume that the blue ground units will spend one time step at each subregion but later will consider the general case. Similar to the index set \mathcal{I} used earlier, we define \mathcal{I}^k to be the index set for the survivability vectors in subregion $k \in [1..K]$. Under the earlier assumption of “the value of information regarding different subregions will be independent from each other” one can write the following formulation at time T considering the distribution q .

$$V(T, q) = \prod_{k=1}^K \left[\max_{i_k \in \mathcal{I}^k} \{v^{i_k} \cdot q_k\} \right] \quad (3.10)$$

Now, by theorem 3.1.3, and the above equation can be modified into

$$\begin{aligned} V(T-1, q) &= \max_{u_{T-1}^o \in \mathcal{U}^K} \{ \mathbf{E}_{y_{T-1}} [V(T, \beta^{y_{T-1}}(q))] \}, \\ &= \max_{u_{T-1}^o \in \mathcal{U}^K} \left\{ \sum_{y_{T-1} \in \mathcal{Y}^K} [V(T, \beta^{y_{T-1}}(q)) Pr(y_{T-1})] \right\}, \\ &= \max_{u_{T-1}^o \in \mathcal{U}^K} \left\{ \sum_{y_{T-1}} \left[\prod_{k=1}^K \left[\max_{i_k \in \mathcal{I}^k} \{v^{i_k} \cdot [\beta^{y_{T-1}}(q)]_k\} \right] Pr(y_{T-1}) \right] \right\}. \end{aligned}$$

Notice that in the formulations above, $\mathcal{U}^K = \mathcal{U}^1 \otimes \mathcal{U}^2 \dots \mathcal{U}^K$. Following the assumption that observations in different subregions will be independent of each other,

$$Pr(y_{T-1}) = \prod_{k=1}^K Pr([y_{T-1}]_k) \quad \text{and} \quad [\beta^{y_{T-1}}(q)]_k = \beta^{[y_{T-1}]_k}(q_k).$$

Substituting these into the previous equation yields

$$\begin{aligned} V(T-1, q) &= \max_{u_{T-1}^o} \left\{ \sum_{y_{T-1}} \left[\prod_{k=1}^K \left[\max_{i_k \in \mathcal{I}^k} \{v^{i_k} \cdot \beta^{[y_{T-1}]_k}(q_k)\} \right] \prod_{k=1}^K Pr([y_{T-1}]_k) \right] \right\} \\ &= \max_{u_{T-1}^o} \left\{ \sum_{y_{T-1}} \left[\prod_{k=1}^K \left[\max_{i_k \in \mathcal{I}^k} \{v^{i_k} \cdot \beta^{[y_{T-1}]_k}(q_k)\} Pr([y_{T-1}]_k) \right] \right] \right\}. \end{aligned}$$

Notice that the last term $Pr([y_{T-1}]_k)$ is independent of $i_k \in \mathcal{I}^k$, so can be taken inside the max.

$$V(T-1, q) = \max_{u_{T-1}^o} \left\{ \sum_{y_{T-1}} \left[\prod_{k=1}^K \left[\max_{i_k \in \mathcal{I}^k} \{v^{i_k} \cdot \beta^{[y_{T-1}]_k}(q_k) Pr([y_{T-1}]_k)\} \right] \right] \right\}$$

where

$$\beta^{[y_{T-1}]_k}(q_k) = \frac{D(R^{[u_{T-1}^o]_k, [y_{T-1}]_k}) q_k}{R^{[u_{T-1}^o]_k, [y_{T-1}]_k} \cdot q_k} \quad (3.11)$$

and

$$Pr(y_{T-1}) = R^{[u_{T-1}^o]_k, [y_{T-1}]_k} \cdot q_k \quad (3.12)$$

After cancellation of common terms:

$$V(T-1, q) = \max_{u_{T-1}^o} \left\{ \sum_{y_{T-1} \in \mathcal{Y}^K} \left[\prod_{k=1}^K \left[\max_{i_k \in \mathcal{I}^k} \{v^{i_k} \cdot D(R^{[u_{T-1}^o]_k, [y_{T-1}]_k}) q_k\} \right] \right] \right\}$$

Now, we will expand the sum defined for \mathcal{Y}^K , and also to shorten the notation we define:

$$D^{[y_{T-1}]_k} \doteq D(R^{[u_{T-1}^o]_k, [y_{T-1}]_k})$$

Then:

$$V(T-1, q) = \max_{u_{T-1}^o} \left\{ \sum_{[y_{T-1}]_1 \in \mathcal{Y}} \dots \sum_{[y_{T-1}]_K \in \mathcal{Y}} \left[\prod_{k=1}^K \left[\max_{i_k \in \mathcal{I}^k} \{v^{i_k} \cdot D^{[y_{T-1}]_k} q_k\} \right] \right] \right\}$$

Notice that the terms inside the product are independent of $[y_{T-1}]_K$ except the K^{th} term. Taking out the K^{th} term from the common product

$$V(T-1, q) = \max_{u_{T-1}^o} \left\{ \sum_{[y_{T-1}]_1} \dots \sum_{[y_{T-1}]_{K-1}} \prod_{k=1}^{K-1} \left[\max_{i_k \in \mathcal{I}^k} \{v^{i_k} \cdot D^{[y_{T-1}]_k} q_k\} \right] \dots \sum_{[y_{T-1}]_K} \left[\max_{i_K \in \mathcal{I}^K} \{v^{i_K} \cdot D^{[y_{T-1}]_K} q_K\} \right] \right\}$$

Continuing in this fashion we get:

$$V(T-1, q) = \max_{u_{T-1}^o} \left\{ \sum_{[y_{T-1}]_1} \left[\max_{i_1 \in \mathcal{I}^1} \{v^{i_1} \cdot D^{[y_{T-1}]_1} q_1\} \right] \dots \sum_{[y_{T-1}]_K} \left[\max_{i_K \in \mathcal{I}^K} \{v^{i_K} \cdot D^{[y_{T-1}]_K} q_K\} \right] \right\}$$

Now using the Max-Plus distributivity property as used before:

$$\begin{aligned}
V(T-1, q) &= \max_{u_{T-1}^o} \left\{ \left[\max_{\{(i_1)_{[y_{T-1}]_1}\} \in [\mathcal{I}^1]^{N_y}} \sum_{[y_{T-1}]_1} \left\{ v^{(i_1)_{y_{T-1}}} \cdot D^{[y_{T-1}]_1} q_1 \right\} \right] \cdots \right. \\
&\quad \left. \max_{\{(i_K)_{[y_{T-1}]_K}\} \in [\mathcal{I}^K]^{N_y}} \sum_{[y_{T-1}]_K} \left[\left\{ v^{(i_K)_{y_{T-1}}} \cdot D^{[y_{T-1}]_K} q_K \right\} \right] \right\} \\
&= \max_{u_{T-1}^o} \left\{ \prod_{k=1}^K \max_{\{(i_k)_{[y_{T-1}]_k}\} \in [\mathcal{I}^k]^{N_y}} \left[\sum_{[y_{T-1}]_k} \left\{ D^{[y_{T-1}]_k} v^{(i_k)_{[y_{T-1}]_k}} \cdot q_k \right\} \right] \right\}
\end{aligned}$$

Again similar to the previous analysis, define: $I^k \doteq \#(\mathcal{I}^k)$.

$$I_{T-1}^k \doteq (I^k)^{N_y} * N_u^k, \quad I_{T-1} \doteq \sum_{k=1}^K (I^k)^{N_y} * N_u^k$$

Now, similar to the functionals we defined earlier, we define the functional, $M_{T-1}[[u_{T-1}^o]_k] : [\mathcal{I}^k]^{N_y} \rightarrow \mathcal{I}_{T-1}^{k, [u_{T-1}^o]_k}$, 1-1 and onto, such that the following ordering holds. $\mathcal{M}_{T-1}[[u_{T-1}^o]_k] \{(i_k)_{[y_{T-1}]_k}\} < \mathcal{M}_{T-1}[[\bar{u}_{T-1}^o]_k] \{(i_k)_{[y_{T-1}]_k}\}$ for $[u_{T-1}^o]_k < [\bar{u}_{T-1}^o]_k$, $u_{T-1}^o \in \mathcal{U}, \bar{u}_{T-1}^o \in \mathcal{U}$. Notice that $\mathcal{I}_{T-1}^{k, [u_{T-1}^o]_k}$ are partitions of \mathcal{I}_{T-1}^k for different $[u_{T-1}^o]_k$; i.e:

$$\mathcal{I}_{T-1}^k = \bigcup_{[u_{T-1}^o]_k} \mathcal{I}_{T-1}^{k, [u_{T-1}^o]_k} \quad \text{and} \quad \mathcal{I}_{T-1}^{k, [u_{T-1}^o]_k} \cap \mathcal{I}_{T-1}^{k, [\bar{u}_{T-1}^o]_k} = \emptyset \quad \text{for} \quad [u_{T-1}^o]_k \neq [\bar{u}_{T-1}^o]_k \quad (3.13)$$

Then,

$$V(T-1, q) = \max_{u_{T-1}^o} \left\{ \prod_{k=1}^K \max_{j \in \mathcal{I}_{T-1}^{k, [u_{T-1}^o]_k}} \{ v_{T-1}^j \cdot q_k \} \right\}$$

where

$$v_{T-1}^j \doteq \sum_{[y_{T-1}]_k} D^{[y_{T-1}]_k} v^{(i_k)_{[y_{T-1}]_k}} \quad \text{where} \quad j = M_{T-1}[[u_{T-1}^o]_k] \{(i_k)_{[y_{T-1}]_k}\} \quad (3.14)$$

By Corollary 3.2.2, we can interchange \prod and \max . Then for a fixed u_{T-1}^o define the set of K tuples (related to control u_{T-1}^o)

$$S_{T-1}^{K, u_{T-1}^o} \doteq \left\{ (j_1, j_2, \dots, j_K) \mid j_k \in \mathcal{I}_{T-1}^{k, [u_{T-1}^o]_k} \right\}$$

and

$$S_{T-1}^K \doteq \bigcup_{u_{T-1}^o \in \mathcal{U}^K} S_{T-1}^{K, u_{T-1}^o}$$

Then,

$$V(T-1, q) = \max_{u_{T-1}^o} \left\{ \max_{i \in S_{T-1}^{K, u_{T-1}^o}} \left\{ \prod_{k=1}^K [v_{T-1}^j \cdot q_k] \right\} \right\}$$

Combining two max's:

$$V(T-1, q) = \max_{i \in S_{T-1}^K} \left\{ \prod_{k=1}^K [v_{T-1}^j \cdot q_k] \right\}.$$

Notice that S_{T-1}^K can also be defined as

$$S_{T-1}^K = \{(j_1, j_2, \dots, j_K) \mid j_k \in \mathcal{I}_{T-1}^k\}$$

i.e., $S_{T-1}^K = \mathcal{I}_{T-1}^1 \otimes \mathcal{I}_{T-1}^2 \otimes \dots \mathcal{I}_{T-1}^K$. Then again by corollary 3.2.2, we can interchange the max and the product operators to get,

$$V(T-1, q) = \prod_{k=1}^K \max_{i \in \mathcal{I}_{T-1}^k} [v_{T-1}^j \cdot q_k].$$

This result showed that the value function retained its form of product of maximums when propagated backwards in DP from the terminal time. With a proof following the same mechanics, it can also be shown that for any $t \in \mathcal{T}^o$ —this form will be preserved. The following theorem summarizes this result.

Theorem 3.2.3. : Suppose that $V(t+1, q_k)$ takes the form

$$V(t+1, q) = \prod_{k=1}^K \max_{i \in \mathcal{I}_{t+1}^k} (v_{t+1}^i \cdot q_k)$$

Then

$$V(t, q) = \prod_{k=1}^K \max_{i \in \mathcal{I}_t^k} (v_t^i \cdot q_k)$$

where \mathcal{I}_t^k propagates according to (3.13) and the vectors, v^i , according to (3.14).

3.2.2 Single UAV for a Region of N Subregions

Now, suppose we have only a single UAV to deploy to battlefield. We need to analyze the best control decision based on this scarce resource. One can argue that it would be best for the ground units to send the sensor once to each subregion instead of sending it more than once to a subregion and leaving another unobserved. We first analyze this situation and later develop a more general framework for unconstrained UAV controls. Again starting from the terminal time T , we have

$$V(T, q) = \prod_{k=1}^K \left[\max_{i_k \in \mathcal{I}^k} \{v^{i_k} \cdot q_k\} \right]$$

Now, first using the Dynamic Programming Principle (DPP), and then using the above equation:

$$\begin{aligned} V(T-1, q) &= \max_{u_{T-1}^o} \{ \mathbf{E}_{y_{T-1} \in \mathcal{Y}} [V(T, \beta^{y_{T-1}}(q))] \} \\ &= \max_{u_{T-1}^o} \left\{ \sum_{y_{T-1}(u_{T-1}^o)} [V(T, \beta^{y_{T-1}}(q)) Pr(y_{T-1})] \right\} \\ &= \max_{u_{T-1}^o} \left\{ \sum_{y_{T-1}(u_{T-1}^o)} \left[\prod_{k=1}^K \left[\max_{i \in \tilde{I}^k} \{v^i \cdot [\beta^{y_{T-1}}(q)]_k\} \right] Pr(y_{T-1}) \right] \right\} \end{aligned}$$

Now we split the \prod into two parts, the subregion where UAV is going to go and the subregions unaffected by the UAV controls. Also:

$$\begin{aligned} [\beta^{y_{T-1}}(q)]_k &= q_k & \text{if } k \neq L(u_{T-1}^o) \\ &= \beta^{y_{T-1}}(q_k) & \text{if } k = L(u_{T-1}^o) \end{aligned}$$

where $L(u_{T-1}^o)$ is an operator that gives the subregion number for any $u_{T-1}^o \in \mathcal{U}$.

Then

$$\begin{aligned} &= \max_{u_{T-1}^o} \left\{ \sum_{y_{T-1}} \left[\prod_{\substack{k=1 \\ k \neq L(u_{T-1}^o)}}^K \left[\max_{i \in \tilde{I}^k} \{v^i \cdot q_k\} \right] \right. \right. \\ &\quad \left. \left. \max_{j \in \tilde{I}^{L(u_{T-1}^o)}} \{v^j \cdot \beta^{y_{T-1}}(q_{L(u_{T-1}^o)})\} Pr(y_{T-1}) \right] \right\} \end{aligned}$$

Again noticing that $Pr(y_{T-1})$ is independent of $j \in \tilde{I}^{L(u_{T-1}^o)}$ it can be taken inside the max. Also to shorten the notation lets define:

$$q_L \doteq (q_{L(u_{T-1}^o)})$$

Now using previously defined formula for $\beta^{y_{T-1}} q_L$:

$$\begin{aligned} &= \max_{u_{T-1}^o} \left\{ \sum_{y_{T-1}} \left[\prod_{\substack{k=1 \\ k \neq L(u_{T-1}^o)}}^K \left[\max_{i \in \tilde{I}^k} \{v^i \cdot q_k\} \right] \right. \right. \\ &\quad \left. \left. \max_{j \in \tilde{I}^{L(u_{T-1}^o)}} \left\{ v^j \cdot \frac{D^{y_{T-1}} q_L}{R^{y_{T-1}} \cdot q_L} R^{y_{T-1}} \cdot q_L \right\} \right] \right\} \\ &= \max_{u_{T-1}^o} \left\{ \sum_{y_{T-1}} \left[\prod_{\substack{k=1 \\ k \neq L(u_{T-1}^o)}}^K \left[\max_{i \in \tilde{I}^k} \{v^i \cdot q_k\} \right] \max_{j \in \tilde{I}^{L(u_{T-1}^o)}} \{v^j \cdot D^{y_{T-1}} q_L\} \right] \right\} \end{aligned}$$

Whole \prod term is independent of y_{T-1} :

$$= \max_{u_{T-1}^o} \left\{ \prod_{\substack{k=1 \\ k \neq L(u_{T-1}^o)}}^K \left[\max_{i \in \tilde{I}^k} \{v^i \cdot q_k\} \right] \left[\sum_{y_{T-1}} \max_{j \in \tilde{I}^{L(u_{T-1}^o)}} \{v^j \cdot D^{y_{T-1}} q_L\} \right] \right\}$$

Now, interchanging \sum and max as done before:

$$\begin{aligned} &= \max_{u_{T-1}^o} \left\{ \prod_{\substack{k=1 \\ k \neq L(u_{T-1}^o)}}^K \left[\max_{i \in \tilde{I}^k} \{v^i \cdot q_k\} \right] \right. \\ &\quad \left. \max_{\{j_{y_{T-1}}\} \in [\tilde{I}^{L(u_{T-1}^o)}]^2} \left[\sum_{y_{T-1}} \{v^{j_{y_{T-1}}} \cdot D^{y_{T-1}} q_L\} \right] \right\} \end{aligned}$$

$$= \max_{u_{T-1}^o} \left\{ \prod_{\substack{k=1 \\ k \neq L(u_{T-1}^o)}}^K \left[\max_{i \in \tilde{I}^k} \{v^i \cdot q_k\} \right] \max_{\{j_{y_{T-1}}\} \in [\tilde{I}^{L(u_{T-1}^o)}]^2} \left[\sum_{y_{T-1}} \{D^{y_{T-1}} v^{j_{y_{T-1}}}\} \cdot q_L \right] \right\}$$

Similar to previous analysis, now define: $I^k \doteq \#(\tilde{I}^k)$.

$$I_{N,T-1}^k \doteq (I^k)^2 * b^k, \quad I_{N,T-1} \doteq \sum_{k=1}^K (I^k)^2 * b^k,$$

$$\tilde{I}_{N,T-1}^k \doteq \{I_T + 1, \dots, I_T + I_{N,T-1}\}, \quad \tilde{I}_{T-1}^k \doteq \tilde{I}_T^k \cup \tilde{I}_{N,T-1}^k$$

define: The functional, $M_{T-1}[u_{T-1}^o] : [\tilde{I}^{L(u_{T-1}^o)}]^2 \rightarrow \tilde{I}_{N,T-1}^{L(u_{T-1}^o), u_{T-1}^o}$, 1-1 and onto, where $\tilde{I}_{N,T-1}^{L(u_{T-1}^o), u_{T-1}^o}$ are partitions of $\tilde{I}_{N,T-1}^{L(u_{T-1}^o)}$ for different $u_{T-1}^o \in L(u_{T-1}^o)$; i.e:

$$\tilde{I}_{N,T-1}^{L(u_{T-1}^o)} = \bigcup_{u_{T-1}^o \in L(u_{T-1}^o)} \tilde{I}_{N,T-1}^{L(u_{T-1}^o), u_{T-1}^o} \quad \text{and}$$

$$\tilde{I}_{N,T-1}^{L(u_{T-1}^o), u_{T-1}^o} \cap \tilde{I}_{N,T-1}^{L(u_{T-1}^o), \bar{u}_{T-1}^o} = \emptyset \quad \text{for } u_{T-1}^o \neq \bar{u}_{T-1}^o$$

Then:

$$= \max_{u_{T-1}^o} \left\{ \prod_{\substack{k=1 \\ k \neq L(u_{T-1}^o)}}^K \left[\max_{i \in \tilde{I}^k} \{v_{T-1}^i \cdot q_k\} \right] \max_{r \in \tilde{I}_{N,T-1}^{L(u_{T-1}^o), u_{T-1}^o}} [v_{T-1}^r \cdot q_L] \right\}$$

where

$$v_{T-1}^r \doteq \sum_{y_{T-1}} D^{y_{T-1}} v^{j_{y_{T-1}}} \quad \text{where, } r = M_{T-1}[u_{T-1}^o](j_1, j_2) \text{ for } j_1, j_2 \in \tilde{I}^{L(u_{T-1}^o)}$$

$$v_{T-1}^s \doteq v^s \quad \text{for } s \notin \tilde{I}^{L(u_{T-1}^o)}$$

By Corrollary 3.2.2, we can interchange \prod and \max . Then for a fixed u_{T-1}^o define:

$$S_T^{K-1/L(u_{T-1}^o)} \doteq \{(i_1, i_2, \dots, i_{K-1}) \mid i_k \in \tilde{I}^k \quad \forall k < L(u_{T-1}^o) \quad \text{and} \\ i_k \in \tilde{I}^{k+1} \quad \forall k \geq L(u_{T-1}^o)\}$$

and similarly

$$[\bar{q}]_k \doteq q_k \quad \forall k < L(u_{T-1}^o) \quad \text{and}$$

$$[\bar{q}]_k \doteq q_{k+1} \quad \forall k \geq L(u_{T-1}^o)$$

Then:

$$= \max_{u_{T-1}^o} \left\{ \max_{j \in S_T^{K-1/L(u_{T-1}^o)}} \prod_{k=1}^{K-1} [v_{T-1}^{j_k} \cdot \bar{q}_k] \max_{r \in \tilde{I}_{N,T-1}^{L(u_{T-1}^o), u_{T-1}^o}} [v_{T-1}^r \cdot q_L] \right\}$$

Combining two max's with defining a new set:

$$S_{T-1}^{K, u_{T-1}^o} = \{(i_1, i_2, \dots, i_K) \mid i_k \in \tilde{I}^k \quad \forall k \neq L(u_{T-1}^o), \quad \text{and} \\ i_{L(u_{T-1}^o)} \in \tilde{I}_{N,T-1}^{L(u_{T-1}^o), u_{T-1}^o}\}$$

Then:

$$= \max_{u_{T-1}^o} \left\{ \max_{i \in S_{T-1}^{K, u_{T-1}^o}} \prod_{k=1}^K [v_{T-1}^{i_k} \cdot q_k] \right\}$$

Now, define:

$$S_{T-1}^K \doteq \bigcup_{u_{T-1}^o} S_{T-1}^{K, u_{T-1}^o}$$

Then:

$$V^{o,f}(T-1, q) = \max_{i \in S_{T-1}^K} \prod_{k=1}^K [v_{T-1}^{i_k} \cdot q_k]$$

If written in terms of subregional indices, the set S_{T-1}^K is:

$$S_{T-1}^K = \left\{ (i_1, i_2, \dots, i_K) \mid i_k \in \tilde{I}_{T-1}^k \quad \text{and} \quad \sum_{k=1}^K N(i_k) = 1 \right\}$$

where $N(j)$ = number of UAV visits related to index j .

Up to this point the constraint and unconstrained UAV control analysis are the

same. Now lets have a look at constrained UAV control (i.e: UAV visiting each subregion once)

Constrained UAV analysis:

Analysing time $t=T-2$ with the Dynamic Programming Principle (DPP), and considering the equation above:

$$\begin{aligned}
 V^{o,f}(T-2, q) &= \max_{\substack{u_{T-2}^o \\ L(u_{T-2}^0) \neq L(u_{T-1}^o)}} \left\{ E_{y_{T-2}(u_{T-2}^o) \in \mathcal{Y}} [V^{o,f}(T-1, \beta^{y_{T-2}}(q))] \right\} \\
 &= \max_{u_{T-2}^o} \left\{ \sum_{y_{T-2}} [V^{o,f}(T, \beta^{y_{T-2}}(q)) Pr(y_{T-2})] \right\} \\
 &= \max_{u_{T-2}^o} \left\{ \sum_{y_{T-2}} \left[\max_{i \in S_{T-1}^K} \left\{ \prod_{k=1}^K [v_{T-1}^{i_k} \cdot [\beta^{y_{T-2}}(q)]_k] \right\} \right] Pr(y_{T-2}) \right\}
 \end{aligned}$$

Now dividing the \prod with first noticing again:

$$\begin{aligned}
 [\beta^{y_{T-2}}(q)]_k &= q_k & \text{if } k \neq L(u_{T-2}^o) \\
 &= \beta^{y_{T-2}}(q_k) & \text{if } k = L(u_{T-2}^o)
 \end{aligned}$$

Then

$$= \max_{u_{T-2}^o} \left\{ \sum_{y_{T-2}} \left[\max_{i \in S_{T-1}^K} \left\{ \prod_{\substack{k=1 \\ k \neq L(u_{T-2}^0)}}^K [v_{T-1}^{i_k} \cdot q_k] \left[v_{T-1}^{i_{L(u_{T-2}^o)}} \cdot \beta^{y_{T-2}}(q_{L(u_{T-2}^o)}) \right] \right\} \right] Pr(y_{T-2}) \right\}$$

And now using lemma (1) we divide the $\max_{i \in S_{T-1}^K}$ into parts that are and not related to u_{T-2}^o and also again to shorten the notation lets define:

$$q_L \doteq (q_{L(u_{T-2}^o)})$$

for a fixed u_{T-2}^o define:

$$S_{T-1}^{K-1/L(u_{T-2}^o)} \doteq \left\{ (i_1, i_2, \dots, i_{K-1}) \mid \begin{array}{l} i_k \in \tilde{I}_{T-1}^k \quad \forall k < L(u_{T-2}^o) \quad \text{and} \\ i_k \in \tilde{I}_{T-1}^{k+1} \quad \forall k \geq L(u_{T-2}^o) \quad \text{and} \\ \sum_{k=1}^{K-1} N(i_k) = 1 \end{array} \right\}$$

and similarly

$$[\bar{q}]_k \doteq q_k \quad \forall k < L(u_{T-2}^o) \quad \text{and}$$

$$[\bar{q}]_k \doteq q_{k+1} \quad \forall k \geq L(u_{T-2}^o)$$

Then:

$$= \max_{u_{T-2}^o} \left\{ \sum_{y_{T-2}} \left[\max_{i \in S_{T-1}^{K-1/L(u_{T-2}^o)}} \left\{ \prod_{k=1}^{K-1} [v_{T-1}^{i_k} \cdot \bar{q}_k] \right\} \right. \right. \\ \left. \left. \max_{j \in \tilde{I}^{L(u_{T-2}^o)}} [v_{T-1}^j \cdot \beta^{y_{T-2}}(q_L)] \Pr(y_{T-2}) \right] \right\}$$

Again noticing that $\Pr(y_{T-2})$ is independent of $j \in \tilde{I}^{L(u_{T-2}^o)}$ it can be taken inside the max. Also notice that for $j \in \tilde{I}^{L(u_{T-2}^o)}$, $v_{T-1}^j = v^j$.

$$= \max_{u_{T-2}^o} \left\{ \sum_{y_{T-2}} \left[\max_{i \in S_{T-1}^{K-1/L(u_{T-2}^o)}} \left\{ \prod_{k=1}^{K-1} [v_{T-1}^{i_k} \cdot \bar{q}_k] \right\} \right. \right. \\ \left. \left. \max_{j \in \tilde{I}^{L(u_{T-2}^o)}} \left[v^j \cdot \frac{D^{y_{T-2}} q_L}{R^{y_{T-2}} \cdot q_L} R^{y_{T-2}} \cdot q_L \right] \right] \right\}$$

$$= \max_{u_{T-2}^o} \left\{ \sum_{y_{T-2}} \left[\max_{i \in S_{T-1}^{K-1/L(u_{T-2}^o)}} \left\{ \prod_{k=1}^{K-1} [v_{T-1}^{i_k} \cdot \bar{q}_k] \right\} \max_{j \in \tilde{I}^{L(u_{T-2}^o)}} [v^j \cdot D^{y_{T-2}} q_L] \right] \right\}$$

But the set $S_{T-1}^{K-1/L(u_{T-2}^o)}$ is independent of y_{T-2} :

$$= \max_{u_{T-2}^o} \left\{ \max_{i \in S_{T-1}^{K-1/L(u_{T-2}^o)}} \left\{ \prod_{k=1}^{K-1} [v_{T-1}^{i_k} \cdot \bar{q}_k] \right\} \sum_{y_{T-2}} \left[\max_{j \in \tilde{I}^{L(u_{T-2}^o)}} \{v^j \cdot D^{y_{T-2}} q_L\} \right] \right\}$$

Now using the Max-Plus Distributivity:

$$\begin{aligned}
&= \max_{u_{T-2}^o} \left\{ \max_{i \in S_{T-1}^{K-1/L(u_{T-2}^o)}} \left\{ \prod_{k=1}^{K-1} [v_{T-1}^{i_k} \cdot \bar{q}_k] \right\} \right. \\
&\quad \left. \max_{\{j_{y_{T-2}}\} \in [\tilde{I}^{L(u_{T-2}^o)}]^2} \left[\sum_{y_{T-2}} \{v^{j_{y_{T-2}}} \cdot D^{y_{T-2}} q_L\} \right] \right\} \\
&= \max_{u_{T-2}^o} \left\{ \max_{i \in S_{T-1}^{K-1/L(u_{T-2}^o)}} \left\{ \prod_{k=1}^{K-1} [v_{T-1}^{i_k} \cdot \bar{q}_k] \right\} \right. \\
&\quad \left. \max_{\{j_{y_{T-2}}\} \in [\tilde{I}^{L(u_{T-2}^o)}]^2} \left[\sum_{y_{T-2}} \{D^{y_{T-2}} v^{j_{y_{T-2}}} \cdot q_L\} \right] \right\}
\end{aligned}$$

But, remember that the previously defined functional: $M_{T-1}[u] : [\tilde{I}^{L(u)}]^2 \rightarrow \tilde{I}_{N,T-1}^{L(u),u}$, 1-1 and onto.

Using $M_{T-1}[u]$ again as $M_{T-1}[u_{T-2}^o]$:

$$= \max_{u_{T-2}^o} \left\{ \max_{i \in S_{T-1}^{K-1/L(u_{T-2}^o)}} \left\{ \prod_{k=1}^{K-1} [v_{T-1}^{i_k} \cdot \bar{q}_k] \right\} \max_{r \in \tilde{I}_{N,T-1}^{L(u_{T-2}^o), u_{T-2}^o}} \left[\sum_{y_{T-2}} \{v_{T-1}^r \cdot q_L\} \right] \right\}$$

where

$$v_{T-1}^r \doteq \sum_{y_{T-2}} D^{y_{T-2}} v^{j_{y_{T-2}}} \quad \text{where, } r = M_{T-1}[u_{T-2}^o](j_1, j_2) \text{ for } j_1, j_2 \in \tilde{I}^{L(u_{T-2}^o)}$$

One should note that this value of v_{T-1}^r is already computed in the previous step.

Now combining two max's with defining a new set:

$$\begin{aligned}
S_{T-2}^{K, u_{T-2}^o} &= \{(i_1, i_2, \dots, i_K) \mid i_k \in \tilde{I}_{T-1}^k, \sum_{\substack{k=1 \\ k \neq L(u_{T-2}^o)}}^K N(i_k) = 1, \forall k \neq L(u_{T-2}^o), \\
&\quad \text{and } i_{L(u_{T-2}^o)} \in \tilde{I}_{N,T-1}^{L(u_{T-2}^o), u_{T-2}^o}\}
\end{aligned}$$

Then:

$$V^{o,f}(T-2, q) = \max_{u_{T-2}^o} \left\{ \max_{i \in S_{T-2}^{K, u_{T-2}^o}} \prod_{k=1}^K [v_{T-1}^{i_k} \cdot q_k] \right\}$$

Now, define:

$$S_{T-2}^K \doteq \bigcup_{u_{T-2}^o} S_{T-2}^{K, u_{T-2}^o}$$

Then:

$$V^{o,f}(T-2, q) = \max_{i \in S_{T-2}^K} \prod_{k=1}^K [v_{T-1}^{i_k} \cdot q_k]$$

If written in terms of subregional indices, the set S_{T-2}^K is: (similar to S_{T-2}^K)

$$S_{T-2}^K = \left\{ (i_1, i_2, \dots, i_K) \mid i_k \in \tilde{I}_{T-1}^k \quad \text{and} \quad \sum_{k=1}^K N(i_k) = 2 \right\}$$

where $N(j)$ = number of UAV visits related to index j .

Now, generalizing this idea to any time $t = T - n$:

Theorem 3.2.4. : Suppose that $V^{o,f}(t+1, q_k)$ takes the form

$$V^{o,f}(t+1, q) = \max_{i \in S_{t+1}^K} \left\{ \prod_{k=1}^K (v_{t+1}^i \cdot q_k) \right\}$$

Then

$$V^{o,f}(t, q) = \max_{i \in S_t^K} \left\{ \prod_{k=1}^K (v_t^i \cdot q_k) \right\}$$

where

$$S_t^K = S_{T-n}^K = \left\{ (i_1, \dots, i_K) \mid i_k \in \tilde{I}_{T-1}^k \quad \text{and} \quad \sum_{k=1}^K N(i_k) = n \right\}$$

and

$$v_t^k = v_{T-1}^k \quad \forall t \in [T-1, T-n]$$

Unconstrained Analysis:

We start analysis from $t = (T-1)^{th}$ step in the previous section. Recall:

$$V^{o,f}(T-1, q) = \max_{i \in S_{T-1}^K} \prod_{k=1}^K [v_{T-1}^{i_k} \cdot q_k]$$

with

$$S_{T-1}^K = \left\{ (i_1, i_2, \dots, i_K) \mid i_k \in \tilde{I}_{T-1}^k \quad \text{and} \quad \sum_{k=1}^K N(i_k) = 1 \right\}$$

where $N(j)$ = number of UAV visits related to index j .

Now the UAV controller does not have a constraint of sending the UAV to another subregion. Thus we can no longer assume: $L(u_{T-2}^o) \neq L(u_{T-1}^o)$.

A similar analysis yields:

$$\begin{aligned} V^{o,f}(T-2, q) &= \max_{u_{T-2}^o} \left\{ E_{y_{T-2}(u_{T-2}^o) \in \mathcal{Y}} [V^{o,f}(T-1, \beta^{y_{T-2}}(q))] \right\} \\ &= \max_{u_{T-2}^o} \left\{ \sum_{y_{T-2}} [V^{o,f}(T, \beta^{y_{T-2}}(q)) Pr(y_{T-2})] \right\} \\ &= \max_{u_{T-2}^o} \left\{ \sum_{y_{T-2}} \left[\max_{i \in S_{T-1}^K} \left\{ \prod_{k=1}^K [v_{T-1}^{i_k} \cdot [\beta^{y_{T-2}}(q)]_k] \right\} \right] Pr(y_{T-2}) \right\} \end{aligned}$$

And splitting the \prod :

$$= \max_{u_{T-2}^o} \left\{ \sum_{y_{T-2}} \left[\max_{i \in S_{T-1}^K} \left\{ \prod_{\substack{k=1 \\ k \neq L(u_{T-2}^o)}}^K [v_{T-1}^{i_k} \cdot q_k] \left[v_{T-1}^{i_{L(u_{T-2}^o)}} \cdot \beta^{y_{T-2}}(q_{L(u_{T-2}^o)}) \right] \right\} \right] Pr(y_{T-2}) \right\}$$

However, unlike the previous case $\max_{i \in S_{T-1}^K}$ can not be divided into two separate entities. The value chosen for $j \in \tilde{I}_{T-1}^{L(u_{T-2}^o)}$ is going to affect what is left for the remaining subregions. The main reason for this is that UAV can either go back to the previous subregion or to a new one. Depending on this choice, indices left to other regions are affected.

Dividing the max we get (writing only the terms after max):

$$\max_{j \in \tilde{I}_{T-1}^{L(u_{T-2}^o)}} \left\{ \max_{i \in S_{T-1}^{K-1/L(u_{T-2}^o),j}} \left\{ \prod_{k=1}^{K-1} [v_{T-1}^{i_k} \cdot \bar{q}_k] [v_{T-1}^j \cdot \beta^{y_{T-2}}(q_L)] \right\} \right\} Pr(y_{T-2})$$

where similar to previous definitions

$$\begin{aligned} S_{T-1}^{K-1/L(u_{T-2}^o),j} \doteq & \{(i_1, i_2, \dots, i_{K-1}) \mid i_k \in \tilde{I}_{T-1}^k \quad \forall k < L(u_{T-2}^o) \quad \text{and} \\ & i_k \in \tilde{I}_{T-1}^{k+1} \quad \forall k \geq L(u_{T-2}^o) \quad \text{and} \\ & \sum_{k=1}^{K-1} N(i_k) + N(j) = 1\} \end{aligned}$$

and similarly

$$[\bar{q}]_k \doteq q_k \quad \forall k < L(u_{T-2}^o) \quad \text{and}$$

$$[\bar{q}]_k \doteq q_{k+1} \quad \forall k \geq L(u_{T-2}^o)$$

Like the previous analysis $Pr(y_{T-2})$ is independent of both sets that define max and thus can be taken inside to give us:

$$\max_{j \in \tilde{I}_{T-1}^{L(u_{T-2}^o)}} \left\{ \max_{i \in S_{T-1}^{K-1/L(u_{T-2}^o),j}} \left\{ \prod_{k=1}^{K-1} [v_{T-1}^{i_k} \cdot \bar{q}_k] [v_{T-1}^j \cdot D^{y_{T-2}} q_L] \right\} \right\}$$

Then, $V^{o,f}(T-2, q)$ would become:

$$= \max_{u_{T-2}^o} \left\{ \sum_{y_{T-2}} \left[\max_{j \in \tilde{I}_{T-1}^{L(u_{T-2}^o)}} \left\{ [v_{T-1}^j \cdot D^{y_{T-2}} q_L] \max_{i \in S_{T-1}^{K-1/L(u_{T-2}^o),j}} \left\{ \prod_{k=1}^{K-1} [v_{T-1}^{i_k} \cdot \bar{q}_k] \right\} \right\} \right] \right\}$$

Using Max-Plus trick:

$$\begin{aligned} = \max_{u_{T-2}^o} & \left\{ \max_{\{j_{y_{T-2}}\} \in [\tilde{I}_{T-1}^{L(u_{T-2}^o)}]^2} \left\{ \sum_{y_{T-2}} [v_{T-1}^{j_{y_{T-2}}} \cdot D^{y_{T-2}} q_L] \right. \right. \\ & \left. \left. \max_{i \in S_{T-1}^{K-1/L(u_{T-2}^o),j_{y_{T-2}}} \left\{ \prod_{k=1}^{K-1} [v_{T-1}^{i_k} \cdot \bar{q}_k] \right\} \right\} \right\} \end{aligned}$$

Unlike previous case, now y_{T-2} is affecting both UAV sent region and other regions.
(No separation) !

PROBLEM !!!

One can take the UAV related term inside the max,:

$$= \max_{u_{T-2}^o} \left\{ \max_{\{j_{y_{T-2}}\} \in [\tilde{I}_{T-1}^{L(u_{T-2}^o)}]^2} \left\{ \sum_{y_{T-2}} \max_{i \in S_{T-1}^{K-1/L(u_{T-2}^o), j_{y_{T-2}}}} \left[v_{T-1}^{j_{y_{T-2}}} \cdot D^{y_{T-2}} q_L \right] \prod_{k=1}^{K-1} [v_{T-1}^{i_k} \cdot \bar{q}_k] \right\} \right\}$$

Now combining the product with defining a new set: (similar to before):

$$S_{T-2}^{K, u_{T-2}^o, j_{y_{T-2}}} = \{(i_1, i_2, \dots, i_K) \mid i_k \in \tilde{I}_{T-1}^k \forall k \neq L(u_{T-2}^o), i_{L(u_{T-2}^o)} = j_{y_{T-2}},$$

$$\text{and } \sum_{\substack{k=1 \\ k \neq L(u_{T-2}^o)}}^K N(i_k) + N(j_{y_{T-2}}) = 1, \}$$

$$= \max_{u_{T-2}^o} \left\{ \max_{\{j_{y_{T-2}}\} \in [\tilde{I}_{T-1}^{L(u_{T-2}^o)}]^2} \left\{ \sum_{y_{T-2}} \max_{i \in S_{T-1}^{K-1/L(u_{T-2}^o), j_{y_{T-2}}}} \prod_{k=1}^K [\hat{v}_{T-1}^{i_k, j_{y_{T-2}}} \cdot q_k] \right\} \right\}$$

where

$$\begin{aligned} \hat{v}_{T-1}^{i_k, j_{y_{T-2}}} &\doteq v_{T-1}^{i_k} && \text{if } k \neq L(u_{T-2}^o) \\ &\doteq D^{y_{T-2}} v_{T-1}^{j_{y_{T-2}}} && \text{if } k = L(u_{T-2}^o) \end{aligned}$$

Now one can use the Max-Plus trick on the last max too:

$$= \max_{u_{T-2}^o} \left\{ \max_{\{j_{y_{T-2}}\} \in [\tilde{I}_{T-1}^{L(u_{T-2}^o)}]^2} \left\{ \max_{\{i_{y_{T-2}}\} \in [S_{T-1}^{K-1/L(u_{T-2}^o), j_{y_{T-2}}}]^2} \left\{ \sum_{y_{T-2}} \prod_{k=1}^K [\hat{v}_{T-1}^{i_k, j_{y_{T-2}}} \cdot q_k] \right\} \right\} \right\}$$

By combining max's, this can be put into a final form of:

$$V^{o,f}(T-2, q) = \max_{i \in S_{T-2}^K} \left\{ \sum_{y_{T-2}} \prod_{k=1}^K [v_{T-2}^{[i_{y_{T-2}]}_k} \cdot q_k] \right\}$$

This finding shows that for a the case of K subregion, single UAV case the value function does not retain its form during backward DP iterations. For this reason, for this kind of settings grid-based methods would still be the only option to solve the optimal tasking problem.

Chapter 4

Refining and Pruning Methods

4.1 Introduction

In this chapter, we introduce two approaches to increase the computation speed of the new methodology defined in theorem 3.1.4. Briefly, the first one, “refining”, will involve the elimination of inactive members of the set \mathcal{I}_t , i.e., the elimination of those that nowhere achieve the maximum. On the other hand, “pruning” will refer judicious elimination of active elements of \mathcal{I}_t .

4.2 Refining the Index Set, \mathcal{I}_t

When backward dynamic programming was employed to obtain the new vectors, v_t , out of the set of \mathcal{V}_{t+1} , it was noticed that the newly generated set of \mathcal{V}_t was not generally the minimal set containing the necessary information at time t . Some vectors in set \mathcal{V}_t were inactive (suboptimal) in the simplex and thus they never influenced the supremum. That is, an inactive vector yields hyperplanes which are everywhere below the supremum of the other hyperplanes; an example can be seen in Fig. 2.2 where purple hyperplanes are inactive.

These inactive vectors hold precious computer memory, and moreover during dynamic programming iterations, since every vector in \mathcal{V}_t is propagated, additional new vectors are produced from them. From this observation, one might wonder whether the progeny of such vectors (through the dynamic program

propagation) would also remain inactive during subsequent steps. If so, then any inactive vector can instantly be eliminated from \mathcal{V}_t at the first time of non-influence, and in this way the growth of the size of \mathcal{V}_t would be slower, thereby greatly speeding computation. The following theorem indicates this propagation of inactivity.

Theorem 4.2.1. *Let \mathcal{R}_{t+1} be the refined subset of \mathcal{I}_{t+1} , i.e.:*

$$\mathcal{R}_{t+1} \doteq \{i \in \mathcal{I}_{t+1} \mid \exists q \in S^N, v_{t+1}^i \cdot q > v_{t+1}^j \cdot q \quad \forall j \in \mathcal{I}_{t+1} \setminus \{i\}\},$$

and let the corresponding refined vectors be $\mathcal{V}_{t+1}^R = \{v_{t+1}^i \in \mathcal{V}_{t+1} \mid i \in \mathcal{R}_{t+1}\}$. Let $(\hat{\mathcal{V}}_t, \hat{\mathcal{I}}_t) = \mathcal{D}^U[(\mathcal{V}_{t+1}^R, \mathcal{R}_{t+1})]$ i.e., the backward propagation of the refined set of vectors. Then,

$$\mathcal{C}[(\hat{\mathcal{V}}_t, \hat{\mathcal{I}}_t)] = \mathcal{C}[(\mathcal{V}_t, \mathcal{I}_t)].$$

In other words, it is sufficient to work with $\hat{\mathcal{V}}_t$, so we only need to propagate \mathcal{V}_{t+1}^R .

Proof. For the sake of presentation we define a slightly different notation for the mapping \mathcal{M} . For each $u^o \in \mathcal{U}$ and $\{i_{y_t}\} \in \mathcal{P}^{N_y}(\mathcal{I}_{t+1})$, we let $M_t[u_t^o](\{i_{y_t}\}) = \mathcal{M}(u^o, \{j_{y_t}\})$. Now, consider $j \in \mathcal{I}_t$ but $j \notin \hat{\mathcal{I}}_t$. Then by the definition of \mathcal{I}_t , $\exists u_t^o \in \mathcal{U}$, and $\{i_{y_t}\} \in (\mathcal{I}_{t+1})^{N_y}$ such that $M_t[u_t^o](\{i_{y_t}\}) = j$. Then for $q \in S^N$ following the formulation 3.1.4 one can write:

$$\begin{aligned} v_t^j \cdot q &= \left(\sum_{y_t} D(R^{u_t^o, y_t}) v_{t+1}^{\{i_{y_t}\}} \right) \cdot q = \sum_{y_t} \left(D(R^{u_t^o, y_t}) v_{t+1}^{\{i_{y_t}\}} \cdot q \right) \\ &= \sum_{y_t} \left(\left[v_{t+1}^{\{i_{y_t}\}} \right]^T D^T(R^{u_t^o, y_t}) q \right) \\ &= \sum_{y_t} \left(\left[v_{t+1}^{\{i_{y_t}\}} \right]^T D(R^{u_t^o, y_t}) q \right) \\ &= \sum_{y_t} \left(v_{t+1}^{\{i_{y_t}\}} \cdot D(R^{u_t^o, y_t}) q \right) \\ &= \sum_{y_t} \left(v_{t+1}^{\{i_{y_t}\}} \cdot \hat{q}^{u_t^o, y_t} R^{u_t^o, y_t} \cdot q \right), \\ &\text{where } \hat{q}^{u_t^o, y_t} \doteq \frac{D(R^{u_t^o, y_t}) q}{R^{u_t^o, y_t} \cdot q} = \beta^{u_t^o, y_t}(q) \end{aligned}$$

Here, following the properties of the mapping $\beta^{u_t^o, y_t}(\cdot)$, it should be noticed that $\hat{q}^{u_t^o, y_t} \in S^N$. Then for $\hat{q}^{u_t^o, y_t}$, by the definition of \mathcal{R}_{t+1} , and $\{i_{y_t}\} \in (\mathcal{I}_{t+1})^{N_y}$, $\exists k_{y_t} \in \mathcal{R}_{t+1}$ we can write:

$$v_{t+1}^{k_{y_t}} \cdot \hat{q}^{u_t^o, y_t} \geq v_{t+1}^{i_{y_t}} \cdot \hat{q}^{u_t^o, y_t} \quad \forall i_{y_t} \in \mathcal{I}_{t+1}$$

which implies:

$$v_t^j \cdot q \leq \sum_{y_t} \left(v_{t+1}^{\{k_{y_t}\}} \cdot \hat{q}^{y_t} R^{u_t^o, y_t} \cdot q \right)$$

Following previous steps this time backwards we can write:

$$v_t^j \cdot q \leq \sum_{y_t} \left(D(R^{u_t^o, y_t}) v_{t+1}^{\{k_{y_t}\}} \right) \cdot q \quad (4.1)$$

Now, similar to $M_t[u_t^o]$, define a functional $\hat{M}_t[u_t^o] : \mathcal{R}_{t+1}^{N_y} \rightarrow \hat{\mathcal{I}}_t^{u_t^o}$, 1-1 and onto, such that the following ordering holds. $\hat{M}_t[u_t^o](\{k_{y_t}\}) < \hat{M}_t[\bar{u}_t^o](\{k_{y_t}\})$ for $u_t^o < \bar{u}_t^o$, $u_t^o \in \mathcal{U}, \bar{u}_t^o \in \mathcal{U}$. Notice that $\hat{\mathcal{I}}_t^{u_t^o}$ are partitions of $\hat{\mathcal{I}}_t$ for different u_t^o ; i.e:

$$\hat{\mathcal{I}}_t = \bigcup_{u_t^o} \hat{\mathcal{I}}_t^{u_t^o} \quad \text{and} \quad \hat{\mathcal{I}}_t^{u_t^o} \cap \hat{\mathcal{I}}_t^{\bar{u}_t^o} = \emptyset \quad \text{for} \quad u_t^o \neq \bar{u}_t^o$$

Then following (4.1)

$$v_t^j \cdot q \leq v_t^r \cdot q,$$

where,

$$v_t^r \doteq \sum_{y_t} D(\mathbf{R}^{u_t^o, y_t}) v_{t+1}^{\{k_{y_t}\}} \quad \text{and} \quad r = \hat{M}_t[u_t^o](\{k_{y_t}\}), r \in \hat{\mathcal{I}}_t^{u_t^o}$$

This analysis shows that, for any $j \in \mathcal{I}_t \setminus \hat{\mathcal{I}}_t, \exists r \in \hat{\mathcal{I}}_t$ (defined from $\{k_{y_t}\} \in \mathcal{R}_{t+1}^{N_y}$) such that $v_t^j \cdot q \leq v_t^r \cdot q$. Thus it is sufficient to propagate \mathcal{R}_{t+1} rather than the complete set \mathcal{I}_{t+1} . \square

Graphically, the propagation of the sets should be done as shown below:

$$\begin{array}{ccc} (\mathcal{V}_{t+1}, \mathcal{I}_{t+1}) & \xrightarrow{REFINE} & (\mathcal{V}_{t+1}^R, \mathcal{R}_{t+1}) \\ & & \downarrow DP \\ (\mathcal{V}_t, \mathcal{I}_t) \doteq & (\hat{\mathcal{V}}_t, \hat{\mathcal{I}}_t) & \xrightarrow{REFINE} (\mathcal{V}_t^R, \mathcal{R}_t) \end{array}$$

Now that we have established that suboptimal vectors do not influence the subsequent value functions, the only thing that remains is to develop a method to identify these inactive vectors in \mathcal{V}_t , and then form the refined vector set \mathcal{V}_t^R by excluding them. Following Theorem 4.2.1, where we defined the refined subsets \mathcal{R}_t of \mathcal{I}_t and \mathcal{V}_t^R of \mathcal{V}_t , we know that a vector v_t^i must be excluded from \mathcal{V}_t^R if there exists no $q \in S^N$ such that $v_t^i \cdot q > v_t^j \cdot q \forall j \in \mathcal{I}_t \setminus \{i\}$. This is equivalent to saying that, v_t^i is an inactive vector if the difference, $v_t^i \cdot q - v_t^j \cdot q < 0 \forall j \in \mathcal{I}_t \setminus \{i\}$ and at any $q \in S^N$. Since this difference is negative for any $q \in S^N$, we notice that taking the maximum of this difference over S^N would still yield a negative number. For this reason, the identification of the inactive vectors can be formulated as an optimization problem over the simplex, S^N . Considering the linear nature of the problem we use the well documented linear programming algorithm.

Now, suppose that a number of vectors are already present in the set \mathcal{V}_t , and we want to determine particularly whether the vector v_t^i is an active member of this set, i.e., if it is influencing the supremum, $V(t, q)$. The following optimization formulation formulates the inactive vector identification problem as a linear programming problem following the ideas developed in the previous paragraph. It formulates the objective function as the difference between the dot product, $v_t^i \cdot q$, and the surface z , which is defined in the constraint equation by the other vectors in \mathcal{V}_t . Other constraints defining the simplex, S^N , appear as well.

$$\left. \begin{aligned} & \max_{q \in S^N} : v_t^i \cdot q - z \\ & \text{subject to : } v_t^j \cdot q - z \leq 0, \quad \forall j \in \mathcal{I}_t - \{i\} \\ & q_k \geq 0, \quad \forall k \in]1, N[\\ & z \geq 0 \\ & \sum q_k = 1. \end{aligned} \right\} \doteq \psi^i \quad (4.2)$$

Equivalently, the above formulation can be expressed in the all-inequality form:

$$-\psi^i = \left\{ \begin{array}{l} \min_x : c \cdot x \\ \text{subject to : } Ax \geq 0, \end{array} \right. \quad (4.3)$$

where

$$x = [q^T \ z]^T, \quad c = c^i \doteq [-v_t^{iT} \ 1]^T \quad (4.4)$$

and

$$A = A^{\mathcal{I}_t, i} \doteq \begin{bmatrix} -a^{1^T} \\ -a^{2^T} \\ \vdots \\ -a^{i-1^T} \\ -a^{i+1^T} \\ \vdots \\ -a^{I_t^T} \\ \mathbf{I}_{(N+1)} \\ b^T \\ -b^T \end{bmatrix}, \quad \text{with} \quad a^j = \begin{bmatrix} [v_t^j]_1 \\ \vdots \\ [v_t^j]_N \\ -1 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 0 \end{bmatrix}$$

If this last optimization scheme (4.3) results in a negative value (positive in the first scheme) then the v_t^i vector should be retained (as it contributes to $V(t, q)$). Otherwise, it should be eliminated. Repeating this scheme for each vector $v_t^i, i \in \mathcal{I}_t$, one could get the minimal set \mathcal{R}_t which still would yield $V(t, q)$. We complete the refining method by giving the Refining algorithm below.

Definition 4.2.2. *Refining Algorithm:*

Suppose that at some time $t \in \mathcal{T}^g$ we have the set of vectors, \mathcal{V}_t and the corresponding index set $\mathcal{I}_t = \{1, \dots, I_t\}$ that define $V(t, q)$. In order to eliminate the inactive vectors in \mathcal{V}_t the following pseudo-code is employed.

Step 0: Let $i = 1$

Loop over, $j = 1 : I_t$

Step 1: Form the matrix $A = A^{\mathcal{I}_t, i}$, and solve (4.3).

Step 2: If $\psi^i \leq 0$: v_t^j is not an active vector.

Erase i^{th} row of A

Else: v_t^j is an active vector.

$$i = i + 1$$

End loop

Once the loop ends, the set of vectors active, \mathcal{V}_t^R , can be extracted from the final form of the matrix A . The k^{th} vector in set \mathcal{V}_t^R will be, $v_t^k = A(k, 1 : N)$

The reader should notice the extreme growth of the size of \mathcal{I}_t in Theorem 3.1.4. Overall, the refining of \mathcal{I}_t gave us a significant boost to computation speeds by reducing this growth. Simulations were done with randomly created \mathcal{V}_T sets and for a sensor tasking problem defined again on S^2 , it was found that the refining algorithm reduced the size of the set \mathcal{I}_{T-1} to an average 1/3 of its original size. The sizes of subsequent sets were even reduced by higher factors. By the end of the third iteration (with observation and refining at each step) the size of \mathcal{I}_{T-3} was reduced by an average factor of more than **30,000**.

Besides this improvement, it was noticed that the simplex method was also giving us a quantitative value about the contribution of each individual vector in $(\mathcal{V}_t, \mathcal{R}_t)$ to our analysis of $V(t, q)$, the quantity $\max_q(v_t^j \cdot q - z)$. Exploiting this value, one can think about eliminating vectors with very small contributions to $V(t, q)$ to keep the size of \mathcal{I}_t more manageable. This idea forms the basis for the next analysis.

4.3 Pruning the Refined Set, \mathcal{R}_t

At a given time $t \in \{0, 1, \dots, T-1\}$, after refining the original set of vectors, \mathcal{V}_t , we ended up with the refined set of vectors, \mathcal{V}_t^R , which consisted of the vectors that were active through the simplex S .

At that time, it was quickly noticed that some of the vectors in \mathcal{V}_t defining $V(t, q)$ were having very small contributions to $V(t, q)$. What we mean is that the value of $V(t, q)$ would have changed very small and/or in a very small portion of the simplex S if such vectors were taken out of \mathcal{V}_t^R .

In order to further improve the computation speed without losing accuracy in our analysis, these vectors might be omitted carefully. To identify which vectors

to eliminate from the refined set, \mathcal{R}_t , we first need to define an error function at time t , ϵ_t , over the probability simplex, S^N , for the pruning analysis. Two candidates chosen for this purpose were the functions defined by L_∞ and L_1 norms. We first present our results with the L_∞ norm and later with the L_1 norm.

Definition 4.3.1. L_∞ based Error Function, ϵ_∞ :

Suppose that at some time $t \in \{0, 1, \dots, T-1\}$, a certain number of vectors were pruned out of the refined set, \mathcal{R}_t , leaving us with the remaining set, \mathcal{P}_t which has a cardinality of $\#(\mathcal{P}_t) = p$. We define the error, ϵ_∞ , occurring by omitting these vectors out of the refined set, using the L_∞ norm as:

$$\epsilon_\infty(\mathcal{P}_t, t) \doteq \|V(t, q) - W^{\mathcal{P}_t}(t, q)\|_\infty = \max_{q \in S^N} \{V(t, q) - W^{\mathcal{P}_t}(t, q)\} \quad (4.5)$$

where $V(t, q)$ is the value function as defined in (2.16), and $W^{\mathcal{P}_t}(t, q)$, value function after pruning, is defined as:

$$W^{\mathcal{P}_t}(t, q) \doteq \max_{i \in \mathcal{P}_t} \{v_t^i \cdot q\}. \quad (4.6)$$

Since, $\epsilon_\infty(\mathcal{P}_t, t)$ is dependent on the choice of \mathcal{P}_t , the set that results in the minimum pruning error should be chosen at the end to yield the optimal pruning error, $\epsilon_\infty^o(t)$. Let $2^{\mathcal{R}}$ denote the power set of \mathcal{R} , i.e., the algebra consisting of all subsets of \mathcal{R} . Then we can write:

$$\epsilon_\infty^o(t) \doteq \min_{\mathcal{P}_t \in 2^{\mathcal{R}}} \epsilon_\infty(\mathcal{P}_t, t). \quad (4.7)$$

Following this definitions, we now look for a method to calculate $\epsilon_\infty^o(t)$, and find \mathcal{P}_t^o . Since $\epsilon_\infty(\mathcal{P}_t, t)$ is defined as the maximum over the simplex S^N , calculation of this error can be considered as an optimization problem over that simplex. For this reason, an error formulation utilizing the linear programming formulation, similar to (4.2), can be formulated over the probability simplex, S^N , and later the well documented simplex algorithm can again be utilized to solve this problem as well. Now, as mentioned in the previous paragraph, at time t we assume that a number of vectors were pruned out of the refined set, \mathcal{R}_t , leaving

us the remaining set, \mathcal{P}_t . In order to reduce notational complexity, here we define the set \mathcal{T}_t , as $\mathcal{T}_t \doteq \mathcal{R}_t \setminus \mathcal{P}_t$. Following the same ideas utilized in the refining formulations, we can formulate the error induced by omitting the vectors v_t^i with $i \in \mathcal{T}_t$, and working only with the vectors v_t^j , $j \in \mathcal{P}_t$ as:

$$\epsilon_\infty(\mathcal{P}_t, t) \doteq \max_{(q,z) \in \mathcal{S}^N \otimes \mathbb{R}} \left[\max_{i \in \mathcal{T}_t} (v_t^i \cdot q) - z \right] \quad (4.8)$$

$$\left. \begin{aligned} &\text{subject to : } v_t^j \cdot q - z \leq 0, \quad \forall j \in \mathcal{P}_t \\ &q_k \geq 0, \quad \forall k \in]1, N[\\ &z \geq 0 \\ &\sum q_k = 1 \end{aligned} \right\} \quad (4.9)$$

Notice that in the above equations z is defined by \mathcal{P}_t , and for this reason could be taken inside the parentheses in (4.8):

$$\epsilon_\infty(\mathcal{P}_t, t) = \max_{(q,z) \in \mathcal{S}^N \otimes \mathbb{R}} \left[\max_{i \in \mathcal{T}_t} (v_t^i \cdot q - z) \right].$$

The function inside the parentheses is uniformly continuous in both q and z . Considering this property, we can interchange the order of the max terms:

$$\epsilon_\infty(\mathcal{P}_t, t) = \max_{i \in \mathcal{T}_t} \left[\max_{(q,z) \in \mathcal{S}^N \otimes \mathbb{R}} (v_t^i \cdot q - z) \right].$$

To complete our discussion, for $i \in \mathcal{T}_t$ we define:

$$\epsilon_\infty^i(\mathcal{P}_t, t) \doteq \max_{q \in \mathcal{S}^N} (v_t^i \cdot q) - z \quad (4.10)$$

$$\left. \begin{aligned} &\text{subject to : } v_t^j \cdot q - z \leq 0, \quad \forall j \in \mathcal{P}_t \\ &q_k \geq 0, \quad \forall k \in]1, N[\\ &z \geq 0 \\ &\sum q_k = 1. \end{aligned} \right\} \quad (4.11)$$

With the formulation above $\epsilon_\infty^i(t)$ can be computed by the utilization of the simplex algorithm. Following this one can compute $\epsilon_\infty(\mathcal{P}_t, t)$ as:

$$\epsilon_\infty(\mathcal{P}_t, t) = \max_{i \in \mathcal{T}_t} \epsilon_\infty^i(\mathcal{P}_t, t). \quad (4.12)$$

At this point, one might wonder about the existence of a possible error bound during DP iterations. The following theorem highlights the boundedness property of the error, $\epsilon_\infty(t)$.

Theorem 4.3.2. *Let $V(t+1, q)$ and $W^{\mathcal{P}_t}(t+1, q)$ be the functions defined above for some pruned set \mathcal{P}_t . If*

$$\epsilon_\infty(\mathcal{P}_t, t+1) = \| V(t+1, q) - W(t+1, q) \|_\infty = \epsilon$$

then

$$\epsilon_\infty(\mathcal{P}_t, t) = \| V(t, q) - W(t, q) \|_\infty \leq \epsilon$$

Proof. By the definition of $V(t+1, q)$ and $W^{\mathcal{P}_t}(t+1, q)$,

$$V(t, q) = \max_{u_t} \left\{ \sum_{y_t} V(t+1, \beta^{u_t, y_t}(q)) P(y_t) \right\}$$

and

$$W^{\mathcal{P}_t}(t, q) = \max_{u_t} \left\{ \sum_{y_t} W^{\mathcal{P}_t}(t+1, \beta^{u_t, y_t}(q)) P(y_t) \right\}.$$

Consequently,

$$V(t, q) - W^{\mathcal{P}_t}(t, q) = \max_{u_t} \left\{ \sum_{y_t} [V(t+1, \beta^{u_t, y_t}(q)) - W^{\mathcal{P}_t}(t+1, \beta^{u_t, y_t}(q))] P(y_t) \right\}$$

with

$$\beta^{u_t, y_t}(q) = \frac{D(R^{u_t, y_t}) q}{R^{u_t, y_t} \cdot q} \in S^N, \quad \forall q \in S.$$

Thus,

$$V(t+1, \beta^{u_t, y_t}(q)) - W^{\mathcal{P}_t}(t+1, \beta^{u_t, y_t}(q)) \leq \epsilon \quad (4.13)$$

which yields:

$$\| V(t, q) - W^{\mathcal{P}_t}(t, q) \|_\infty \leq \max_{u_t} \left\{ \sum_{y_t} \epsilon P(y_t) \right\} = \epsilon \quad \square$$

□

Corollary 4.3.3. *If for some $t \in \{0, \dots, T\}$, $\epsilon_\infty^o(t) = \epsilon$, then for all $0 \leq \tau \leq t$, one has $\epsilon_\infty^o(\tau) \leq \epsilon$.*

Proof. Notice that theorem 4.3.2 was valid for any pruned set \mathcal{P}_t . For this reason, the inequality holds for the optimal pruned error as well. By induction on theorem 4.3.2 we complete the proof. \square

This corollary states that the error induced by pruning a set of vectors out of the analysis at time t , would not grow during subsequent steps in the DP. The easy use of linear programming and the boundedness of the pruning error make the error function defined by the L_∞ norm look like a perfect candidate for measuring the error in pruning. However, ϵ_∞ also possesses two disadvantages. First, and most importantly, it was found that pruning was not optimal for approximating $V(t, q)$ with a smaller set using the L_∞ norm. That is, the optimal set of, say \bar{n} , vectors for approximating $V(t, \cdot)$ (where $\bar{n} < \#\mathcal{I}_t = \#\mathcal{V}_t$) may not consist of a subset of the elements of \mathcal{V}_t . The following example shows this situation.

Here, we present a counter example to show that when approximating the function $\max_i(v^i \cdot q)$, $i \in \mathcal{R}$, with the ϵ_∞ based pruning method defined in (??), one may not end up with the optimal pruned set that would minimize ϵ_∞ .

Consider the two-dimensional simplex, S^2 , and let $q \in S^2$ be:

$$q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} q_1 \\ 1 - q_1 \end{bmatrix}, \text{ with } q_1 \in [0, 1]. \quad (4.14)$$

On this simplex, S^2 , consider 3 vectors comprising the set \mathcal{R} :

$$v^1 = \begin{bmatrix} 0.95 \\ 0.35 \end{bmatrix}, \quad v^2 = \begin{bmatrix} 0.75 \\ 0.75 \end{bmatrix}, \quad v^3 = \begin{bmatrix} 0.35 \\ 0.95 \end{bmatrix}$$

Suppose that we need to prune out one of these vectors considering the ϵ_∞ criteria (pruning out the vector that will yield the least error based on L_∞ norm). For a better visualization of the situation, the vectors above are plotted in Fig. 4.1 (as solid lines). Following the definition of ϵ_∞^i in (4.10) one finds:

$$\epsilon_\infty^1 = 0.20, \quad \epsilon_\infty^2 = 0.10, \quad \epsilon_\infty^3 = 0.20$$

Considering these numbers, one would decide to prune out v^2 so that the remaining vectors, v^1 and v^3 would be approximating to the original piecewise function $\max_{i \in \mathcal{R}}(v^i \cdot q)$ with the least error, ε_∞ . Meanwhile, if we analyze approximations other than pruning, we can notice that the following vectors (shown as dashed lines in Fig. 4.1) would yield the optimal approximation, with an error of $\varepsilon_\infty = 0.0667$:

$$v^4 = \begin{bmatrix} 0.8833 \\ 0.4833 \end{bmatrix}, \quad v^5 = \begin{bmatrix} 0.4833 \\ 0.8833 \end{bmatrix}$$

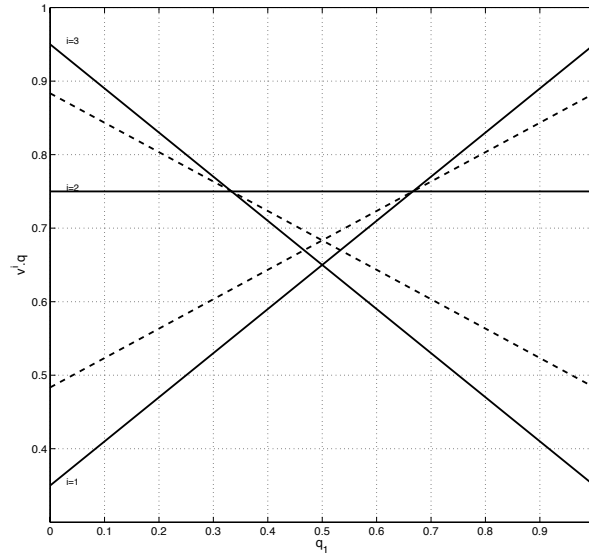


Figure 4.1: A counter example for pruning with ε_∞ . Rather than any 2 of the original 3 vectors (solid lines), vectors with dashed lines comprise the optimal set approximating $\max_i(v^i \cdot q)$

Second, the authors are concerned that using the L_∞ norm might not be an accurate way to measure the pruning error. To see such an inaccurate pruning situation consider a set of v_t^i 's where some of these vectors having a huge contribution to $V(t, q)$ by means of the infinity norm only in a very small region of the simplex, S^N . Then according to L_∞ pruning algorithm, these vectors may still be stored in the memory not required to be pruned.

Because of these drawbacks an error function based on the L_1 norm, $\epsilon_1(\mathcal{P}_t, t)$, was also considered for analyzing a pruned set \mathcal{P}_t . We define $\epsilon_1(\mathcal{P}_t, t)$

as below:

$$\epsilon_1(\mathcal{P}_t, t) \doteq \int_{S^N} V(t, q) - W^{\mathcal{P}_t}(t, q) dq \quad (4.15)$$

where $V(t, q)$ and $W_t^{\mathcal{P}}(t, q)$ are as defined earlier. Similar to $\epsilon_\infty^o(t)$ we denote the optimal pruning error according to L_1 analysis with $\epsilon_1^o(t)$.

The main advantage of using $\epsilon_1^o(t)$ was highlighted in [McE09], where it was proven that an error function based on the L_1 norm would be convex, and moreover, when approximating a set of functions with another smaller set of functions the optimal reduced complexity representation would be comprised of a subset of the original set of functions. That is, with an error metric based on the L_1 norm, pruning does, in fact, yield the optimal solution. This is the superiority of the L_1 norm over the L_∞ norm. Having an optimal set of pruned vectors over the refined set \mathcal{R}_t we are encouraged us to use $\epsilon_1^o(t)$ over $\epsilon_\infty^1(t)$. However, contrary to these fine properties $\epsilon_1^o(t)$ was not found to posses the boundedness property of $\epsilon_\infty(t)$ during DP. A counterexample is given below.

Here, we present an example to highlight the fact that ϵ_1^o pruning error might grow during DP iterations. For this purpose, again on S^2 as defined in (4.14), we consider a refined set $\mathcal{R}_{t+1} = \{1, 3\}$ with:

$$v_{t+1}^1 = \begin{bmatrix} 0.95 \\ 0.25 \end{bmatrix} \quad \text{and} \quad v_{t+1}^3 = \begin{bmatrix} 0.7 \\ 0.65 \end{bmatrix}$$

Suppose that out of these 2 vectors we need to prune the vector that will result to the the minimal pruning error, $\epsilon_1^o(t+1)$. Similar to $\epsilon_\infty^i(t)$ defined earlier, we define the pruning error induced by pruning a vector i from the refined set, \mathcal{R}_t , but this time based on the L_1 norm as:

$$\epsilon_1^i(t+1) \doteq \int_{S^{N_u}} V(t+1, q) - V_i^-(t+1, q) dq$$

where $V_i^-(t+1, q) = \max_{i \in \mathcal{I}_{t+1} \setminus \{i\}} (v^i \cdot q)$. Now, since we are trying to prune out one single vector that would lead to least error in our analysis, the error $\epsilon_1^o(t+1)$ would be:

$$\epsilon_1^o(t+1) = \min_{i \in \mathcal{R}_{t+1}} \epsilon_1^i(t+1)$$

Following the definition of $\epsilon_1^i(t+1)$ above one can find,

$$\epsilon_1^1(t+1) = 0.0481 \quad \text{and} \quad \epsilon_1^3(t+1) = 0.1231$$

Then, $\epsilon_1(t+1) = 0.0481$, and vector 1 should be pruned out to give us the pruned set, $P_{t+1} = \{3\}$. Now, we analyze $V(t, q)$ and $W(t, q)$ backwards in time. Remember that:

$$V(t, q) = \max_{u_t^o} \left\{ \sum_{y_t} \max_{i \in \mathcal{R}_{t+1}} (v_{t+1}^i \cdot D^{y_t, u_t^o} q) \right\}$$

Since the problem is developed on S^2 , a physical system consisting of two buildings, one of which having an enemy, could be considered as a real life application of this problem. We enumerate the buildings as Building 1 and Building 2, and thus $u_t^o \in \{1, 2\}$. We also assume that the UAV's observation on a building could result in either a detection (detecting an opposing force) which we denote by 1, or a non-detection which we denote by 2. With this, $y_t \in \{1, 2\}$, as well. For these sets, we use the following D^{y_t, u_t^o} matrices:

$$\begin{aligned} D^{1,1} &= \begin{bmatrix} 0.95 & 0 \\ 0 & 0.1 \end{bmatrix}, & D^{2,1} &= \begin{bmatrix} 0.05 & 0 \\ 0 & 0.9 \end{bmatrix} & (u = 1) \\ D^{1,2} &= \begin{bmatrix} 0.1 & 0 \\ 0 & 0.95 \end{bmatrix}, & D^{2,2} &= \begin{bmatrix} 0.9 & 0 \\ 0 & 0.05 \end{bmatrix} & (u = 2) \end{aligned}$$

Performing the numerics, one can find the piecewise linear functions defining $V(t, q)$:

$$V(t, q_1) = \begin{cases} 0.65 + 0.05q_1 & \text{if } 0 \leq q_1 < 0.0816, \\ 0.63 + 0.295q_1 & \text{if } 0.0816 \leq q_1 < 0.6154, \\ 0.61 + 0.3275q_1 & \text{if } 0.6154 \leq q_1 < 0.9383, \\ 0.25 + 0.70q_1 & \text{if } 0.9383 \leq q_1 \leq 1, \end{cases}$$

and following similar calculations one can find,

$$W(t, q) = \max_{u_t^o} \left\{ \sum_{y_t} \max_{i \in P_{t+1}} (v_{t+1}^i \cdot D^{y_t, u_t^o} q) \right\}$$

as:

$$W(t, q_1) = 0.65 + 0.05q_1.$$

Integrating the difference between $V(t, q)$ and $W(t, q)$ one finds:

$$\begin{aligned}\epsilon_1(t) &= \int_{S^{N_u}} V(t, q) - W(t, q) dq \\ &= \int_0^1 V(t, q_1) - W(t, q_1) dq_1 = 0.1058\end{aligned}$$

Since $\epsilon_1(t+1) > \epsilon_1(t)$, we notice that the error propagation of DP is not bounded with the L_1 based error metric.

Because of this unboundedness one might be worried about error growth during the propagation process. However, even though $\epsilon_1(t)$ might grow during subsequent steps in DP, an upper bound for the error growth is always maintained because of the relationship between the L_∞ and L_1 norms. The following theorem highlights this fact.

Theorem 4.3.4. *Suppose at some time $t \in \{0, \dots, T\}$, $\epsilon_\infty(t) = \epsilon$. Then during DP, for any $0 \leq \tau \leq t$, $\epsilon_1(\tau) \leq \epsilon$.*

Proof. Following the definition of $\epsilon_1(\tau)$ and $\epsilon_\infty(\tau)$ we can write:

$$\begin{aligned}\epsilon_1(\tau) &= \int_{S^N} V(\tau, q) - W(\tau, q) dq \leq \int_{S^N} \max_q \{V(\tau, q) - W(\tau, q)\} dq \\ &= \int_{S^N} \epsilon_\infty(\tau) dq = \epsilon_\infty(\tau)\end{aligned}$$

Now, from Corollary 4.3.3, we have $\epsilon_\infty(\tau) \leq \epsilon_\infty(t) = \epsilon$ for any $0 \leq \tau \leq t$. Combining these, one finds that $\epsilon_1(\tau) \leq \epsilon$ \square

Although this theorem eases our worries about unbounded growth of $\epsilon_1(t)$, it can be easily seen that it is actually a conservative bound. Here, one needs to trade off between computation speed and error bounding carefully.

Another issue that needs attention is the numerical computation methods needed to calculate $\epsilon_1(t)$. Unlike $\epsilon_\infty(t)$ one cannot use linear programming, and numerical integration methods are required for complex piecewise functions that define $V(t, q)$. For probability simplexes with dimensions lower than four, volume

calculation algorithms not subject to the curse-of-dimensionality can be developed with the help of visual aids. For higher dimensions, the construction of such volume computation algorithms appears to be a difficult task. Although a promising approach to find the volume of such higher dimensional problems can be found in [ST01].

Meanwhile, both ϵ_∞^o and ϵ_1^o based approaches suffer from the computational cost of the search for the optimal pruned set. In order to find an optimal pruned set $\mathcal{P} \doteq \mathcal{P}_t$ at a given time t , one needs to analyze all possible subset combinations of an original set $\mathcal{R} \doteq \mathcal{R}_t$, and for large sets considering all such combinations with numerically calculating relative pruning errors for each subset would reduce the computational advantages of pruning. Especially, the cost of such search will noticeably be demanding for the L_1 pruning approach considering the mentioned issues in the previous paragraph. In order to avoid such computational burden one might resort to the well-known “Stingy” or the “Greedy” algorithms (c.f., [NW78]). Briefly, the Stingy algorithm starts with the refined set \mathcal{R} , and at each iteration removes the vector that would lead to the smallest pruning error from this set, leading finally to pruned set \mathcal{P} . Greedy on the opposite, starts with an empty set, and at each iteration adds the vector that would give the largest increase in value to the newly forming set, leading finally to the pruned set \mathcal{P} . Before introducing the pseudo-code of both algorithms we introduce functions, $W_j^-(t, q)$ and $W_j^+(t, q)$, similar to $W^{\mathcal{P}-t}(t, q)$ defined earlier. Given $t \in]0, T - 1[$, $q \in S^X$ and $\mathcal{P} \subset \mathcal{R}$, let

$$W_j^-(t, q) \doteq \max_{i \in \mathcal{P}, i \neq j} (v_t^i \cdot q), \quad \forall j \in \mathcal{P} \quad (4.16)$$

$$W_j^+(t, q) \doteq \max_{i \in \mathcal{P} \cup \{j\}} (v_t^i \cdot q), \quad \forall j \in \mathcal{T} = \mathcal{R}_t \setminus \mathcal{P}. \quad (4.17)$$

Also, we will let $\bar{\epsilon}$ be a predefined bound on the allowable pruning error, and \bar{n} be a predefined maximum number of elements of the set \mathcal{P} . We include both $\bar{\epsilon}$, and \bar{n} as possible stopping criteria for the algorithms. However, one would typically choose $\bar{\epsilon} = 0$ or $\bar{n} = \infty$, depending on the criteria preferred. With this new notation the pseudo-code for the stingy and greedy algorithms for the L_1 norm approach is as follows:

Stingy Algorithm:

Step 0: Starting with set \mathcal{R} , initialize $\mathcal{P} = \mathcal{R}$.

Step 1: If: $\#\mathcal{P} \leq \bar{n}$, stop.

Else: Select: $k \in \mathcal{P}$ such that:

$$k = \operatorname{argmin}_{j \in \mathcal{P}} \int_{S^N} [W(t, q) - W_j^-(t, q)] dq$$

(with ties settled arbitrarily).

Step 2: If: $\int_{S^N} [V(t, q) - W_k^-(t, q)] dq \geq \bar{\epsilon}$, stop.

Else: Update: $\mathcal{P} = \mathcal{P} \setminus \{k\}$. Return to Step 1.

Greedy Algorithm:

Step 0: Starting with the empty set, \emptyset , initialize $\mathcal{P} = \emptyset$.

Step 1: If: $\#\mathcal{P} \geq \bar{n}$, stop.

Else: Select: $k \in \mathcal{R} \setminus \mathcal{P}$ such that:

$$k = \operatorname{argmax}_{j \in \mathcal{R} \setminus \mathcal{P}} \int_{S^N} [W_j^+(t, q) - W(t, q)] dq$$

(with ties settled arbitrarily).

Update $\mathcal{P} = \mathcal{P} \cup \{k\}$.

Step 2: If: $\int_{S^N} [V(t, q) - W(t, q)] dq \leq \bar{\epsilon}$, stop.

Else: Return to Step 1.

The pseudo-codes for stingy and greedy algorithms could also be utilized for the L_∞ based pruning with k being defined as the following:

$$k = \operatorname{argmin}_{j \in \mathcal{P}} \left\{ \max_{q \in S} (W(., q) - W_j^-(., q)) \right\} \quad \text{for Stingy, and}$$

$$k = \operatorname{argmax}_{j \in \mathcal{P}} \left\{ \max_{q \in S} (W(., q) - W_j^+(., q)) \right\} \quad \text{for Greedy}$$

In practice when $\#\mathcal{R}$ is large, the conditions involving $\bar{\epsilon}$ might be omitted and both algorithms could be run considering only the conditions on \bar{n} . Also it could be noticed that the Stingy Algorithm should be preferred when $\bar{n} \geq \#\mathcal{R}/2$ and Greedy should be preferred otherwise. Unfortunately, although this methods would avoid the computational burden of an optimal set search, their resulting values would be suboptimal. We briefly present an example to point out that the

computationally fast Greedy Search algorithm may not yield optimal solutions for the L_∞ and L_1 based pruning methods

Again on the two-dimensional simplex, S^2 , consider 3 vectors comprising the set \mathcal{R} :

$$v^1 = \begin{bmatrix} 0.95 \\ 0.35 \end{bmatrix}, \quad v^2 = \begin{bmatrix} 0.75 \\ 0.75 \end{bmatrix}, \quad v^3 = \begin{bmatrix} 0.35 \\ 0.95 \end{bmatrix}$$

If one needs to prune this set \mathcal{R} to \mathcal{P} with cardinality of 1 using the Greedy Algorithm, he would get the pruned set $\mathcal{P} = \{1\}$ following both L_∞ and L_1 approaches with pruning errors, $\epsilon_\infty = 0.6$ and $\epsilon_1 = 0.1667$ respectively. On the other hand, by considering all other pruning options one would find that the optimal pruned set would be $\mathcal{P}^o = \{2\}$ with optimal errors $\epsilon_\infty^o = 0.2$ and $\epsilon_1^o = 0.0667$.

Given that this approach does not yield the optimal pruned set, one must be concerned with the deviation from the result the optimal pruned set would yield. Here, by referring to known results regarding optimization of submodular functions (c.f., Nemhauser and Wolsey [NW78]), we at least obtain a bound on the error induced by our suboptimal pruning using the greedy algorithm. Note that submodular functions are mappings from a class of sets to the real line, which meet a condition given in the proof below. In order to the error bound for the greedy algorithm, first, we must identify the submodular set functions that could be used for the pruning process when greedy is utilized. Following our discussion above defining the greedy algorithm, and the methodology developed in [NW78], one might first think about using set-based error functions, similar to error functions in (4.8) and (4.15), for this purpose. Specifically, one might take

$$\delta_{t,\infty}(Z) \doteq \max_{q \in S^X} \left\{ V(t, q) - \max_{i \in Z} (v^i \cdot q) \right\}, \quad Z \subset \mathcal{R}$$

and

$$\delta_{t,1}(Z) \doteq \int_{q \in S^X} \left\{ V(t, q) - \max_{i \in Z} (v^i \cdot q) \right\}, \quad Z \subset \mathcal{R}.$$

It should be noticed easily that both functions could easily been incorporated into the pruning algorithms and thus would be great choices. However, it

was noticed that neither functions carried the suboptimality property. Because of this reason, we turn our attention to similarly defined functions:

$$f_\infty(Z) \doteq \max_{q \in S^X} \left\{ \max_{i \in Z} (v^i \cdot q) \right\}, \text{ and} \quad (4.18)$$

$$f_1(Z) \doteq \int_{q \in S^X} \left\{ \max_{i \in Z} (v^i \cdot q) \right\}, \quad Z \subset \mathcal{R}. \quad (4.19)$$

By the nature of the max operator, f_∞ as defined above will not be useful for Stingy and Greedy algorithms. Because of this, f_∞ will not be employed for pruning practices. Below we show the submodularity property of both functions.

Theorem 4.3.5. *Functions f_∞ and f_1 defined in (4.18) and (4.19) are both submodular.*

Proof. We must show that f_∞ and f_1 satisfy the submodularity property, i.e., that

$$f_\eta(A) + f_\eta(B) \geq f_\eta(A \cup B) + f_\eta(A \cap B) \quad \forall A, B \subseteq \mathcal{R}$$

for $\eta \in \{1, \infty\}$. We begin with f_∞ . Given $A, B \subseteq \mathcal{R}$ one has:

$$\begin{aligned} f_\infty(A \cup B) &= \max_{q \in S} \left\{ \max_{i \in A \cup B} (v_t^i \cdot q) \right\} \\ &= \max \left\{ \max_{q \in S} \left\{ \max_{i \in A} (v_t^i \cdot q) \right\}, \max_{q \in S} \left\{ \max_{i \in B} (v_t^i \cdot q) \right\} \right\} \\ &= \max \{f_\infty(A), f_\infty(B)\}. \end{aligned} \quad (4.20)$$

Without loss of generality let $f_\infty(A) \geq f_\infty(B)$. Then, (4.20) implies

$$f_\infty(A \cup B) = f_\infty(A). \quad (4.21)$$

On the other hand, since $(A \cap B) \subseteq B$

$$\begin{aligned} f_\infty(A \cap B) &= \max_{q \in S} \left\{ \max_{i \in A \cap B} (v_t^i \cdot q) \right\} \leq \max_{q \in S} \left\{ \max_{i \in B} (v_t^i \cdot q) \right\} \\ &= f_\infty(B) \end{aligned} \quad (4.22)$$

Combining (4.21) and (4.22), one has

$$f_\infty(A \cup B) + f_\infty(A \cap B) \leq f_\infty(A) + f_\infty(B), \quad (4.23)$$

which implies that suboptimality holds for f_∞ .

Now we turn to $f_1(\cdot)$. Given $A, B \subset \mathcal{R}$, we first partition the probability simplex, S , with the following 4 disjoint sets:

$$S_A \doteq \{q \in S \mid \exists i \in (A \setminus B), v^i \cdot q > v^j \cdot q, \forall j \in B\}, \quad (4.24)$$

$$S_B \doteq \{q \in S \mid \exists i \in (B \setminus A), v^i \cdot q > v^j \cdot q, \forall j \in A\}, \quad (4.25)$$

$$S_I \doteq \{q \in S \mid \exists i \in (A \cap B), v^i \cdot q > v^j \cdot q, \forall j \in (A \setminus B) \cup (B \setminus A)\}, \quad (4.26)$$

and

$$S_K \doteq S \setminus (S_A \cup S_B \cup S_I). \quad (4.27)$$

Notice that:

$$S = S_A \cup S_B \cup S_I \cup S_K, \quad \text{and} \quad S_A \cap S_B \cap S_I \cap S_K = \emptyset. \quad (4.28)$$

Since S_K is defined in terms of S_A , S_B and S_I , we first give an equivalent formulation for S_K by considering the definitions of the other sets.

Lemma 4.3.6. *S_K defined in (4.27) is equivalent to the following set:*

$$\begin{aligned} S_K = & \left\{ q \in S \mid \max_{i \in A \setminus B} (v^i \cdot q) = \max_{j \in B \setminus A} (v^j \cdot q) \geq \max_{k \in A \cap B} (v^k \cdot q) \right\} \\ & \cup \left\{ q \in S \mid \max_{i \in A \setminus B} (v^i \cdot q) = \max_{j \in A \cap B} (v^j \cdot q) \geq \max_{k \in B \setminus A} (v^k \cdot q) \right\} \\ & \cup \left\{ q \in S \mid \max_{i \in B \setminus A} (v^i \cdot q) = \max_{j \in A \cap B} (v^j \cdot q) \geq \max_{k \in A \setminus B} (v^k \cdot q) \right\} \end{aligned}$$

Proof. First, we introduce the following notations for ease of notations.

$$\bar{A} \doteq A \setminus B, \quad \bar{B} \doteq B \setminus A, \quad \text{and} \quad AB \doteq A \cap B \quad (4.29)$$

Note that, the definitions of S_A , S_B , and S_I in (4.24), (4.25) and (4.26) are equivalent to the following formulations.

$$S_A = \left\{ q \in S \mid \max_{i \in A} (v^i \cdot q) > \max_{j \in B} (v^j \cdot q) \right\}, \quad (4.30)$$

$$S_B = \left\{ q \in S \mid \max_{i \in \bar{B}} (v^i \cdot q) > \max_{j \in A} (v^j \cdot q) \right\}, \quad (4.31)$$

$$S_I = \left\{ q \in S \mid \max_{i \in AB} (v^i \cdot q) > \max_{j \in \bar{A} \cup \bar{B}} (v^j \cdot q) \right\}. \quad (4.32)$$

On the other hand by using the set identity $A \setminus B = A \cap B'$ and other basic set theory formulations, the set S_K from (4.27) could be written as:

$$S_K = S \cap (S'_A \cap S'_B \cap S'_I) = (S \cap S'_I) \cap (S'_A \cap S'_B) = S'_I \cap (S'_A \cap S'_B) \quad (4.33)$$

since S is the universal set. Following (4.30):

$$S'_A = \left\{ q \in S \mid \max_{i \in \bar{A}} (v^i \cdot q) \leq \max_{j \in B} (v^j \cdot q) \right\} = \left\{ q \in S \mid \max_{i \in B} (v^i \cdot q) \geq \max_{j \in \bar{A}} (v^j \cdot q) \right\}$$

Meanwhile, since $B = \bar{B} \cup AB$, we can write S'_A as the union of sets SA_1 and SA_2 defined below:

$$S'_A = \underbrace{\left\{ q \in S \mid \max_{i \in \bar{B}} (v^i \cdot q) \geq \max_{j \in \bar{A}} (v^j \cdot q) \right\}}_{\doteq SA_1} \cup \underbrace{\left\{ q \in S \mid \max_{i \in AB} (v^i \cdot q) \geq \max_{j \in \bar{A}} (v^j \cdot q) \right\}}_{\doteq SA_2}$$

In a similar fashion by considering (4.31) and using the relation $A = \bar{A} \cup AB$:

$$\begin{aligned} S'_B &= \left\{ q \in S \mid \max_{i \in \bar{B}} (v^i \cdot q) \leq \max_{j \in A} (v^j \cdot q) \right\} = \left\{ q \in S \mid \max_{i \in \bar{A}} (v^i \cdot q) \geq \max_{j \in \bar{B}} (v^j \cdot q) \right\} \\ &= \underbrace{\left\{ q \in S \mid \max_{i \in \bar{A}} (v^i \cdot q) \geq \max_{j \in \bar{B}} (v^j \cdot q) \right\}}_{\doteq SB_1} \cup \underbrace{\left\{ q \in S \mid \max_{i \in AB} (v^i \cdot q) \geq \max_{j \in \bar{B}} (v^j \cdot q) \right\}}_{\doteq SB_2}, \end{aligned}$$

and by considering (4.32):

$$\begin{aligned} S'_I &= \left\{ q \in S \mid \max_{i \in AB} (v^i \cdot q) \leq \max_{j \in \bar{A} \cup \bar{B}} (v^j \cdot q) \right\} = \left\{ q \in S \mid \max_{i \in \bar{A} \cup \bar{B}} (v^i \cdot q) \geq \max_{j \in AB} (v^j \cdot q) \right\} \\ &= \underbrace{\left\{ q \in S \mid \max_{i \in \bar{A}} (v^i \cdot q) \geq \max_{j \in AB} (v^j \cdot q) \right\}}_{\doteq SI_1} \cup \underbrace{\left\{ q \in S \mid \max_{i \in \bar{B}} (v^i \cdot q) \geq \max_{j \in AB} (v^j \cdot q) \right\}}_{\doteq SI_2}. \end{aligned}$$

Recalling (4.33), we have: $S_K = S'_I \cap (S'_A \cap S'_B) = (S'_I \cap S'_A) \cap S'_B$. Expanding the first term on the right by using the newly defined sets above:

$$\begin{aligned} (S'_I \cap S'_A) &= (SI_1 \cup SI_2) \cap (SA_1 \cup SA_2) \\ &= (SI_1 \cap SA_1) \cup (SI_1 \cap SA_2) \cup (SI_2 \cap SA_1) \cup (SI_2 \cap SA_2). \end{aligned}$$

Now let:

$$\begin{aligned} S_1 &\doteq (SI_1 \cap SA_1) = \left\{ q \in S \mid \max_{i \in \bar{B}}(v^i \cdot q) \geq \max_{j \in \bar{A}}(v^j \cdot q) \geq \max_{j \in AB}(v^j \cdot q) \right\} \\ S_2 &\doteq (SI_1 \cap SA_2) = \left\{ q \in S \mid \max_{i \in \bar{A}}(v^i \cdot q) = \max_{j \in AB}(v^j \cdot q) \right\} \\ S_3 &\doteq (SI_2 \cap SA_1) = \left\{ q \in S \mid \max_{i \in \bar{B}}(v^i \cdot q) \geq \max_{j \in AB}(v^j \cdot q), \max_{i \in \bar{B}}(v^i \cdot q) \geq \max_{j \in \bar{A}}(v^j \cdot q) \right\} \\ S_4 &\doteq (SI_2 \cap SA_2) = \left\{ q \in S \mid \max_{i \in \bar{B}}(v^i \cdot q) \geq \max_{j \in AB}(v^j \cdot q) \geq \max_{j \in \bar{A}}(v^j \cdot q) \right\}. \end{aligned}$$

Then, for the final phase:

$$\begin{aligned} S_K &= (S_1 \cup S_2 \cup S_3 \cup S_4) \cap S'_B \\ &= (S_1 \cap S'_B) \cup (S_2 \cap S'_B) \cup (S_3 \cap S'_B) \cup (S_4 \cap S'_B) \\ &= (S_1 \cap (SB_1 \cup SB_2)) \cup (S_2 \cap (SB_1 \cup SB_2)) \cup (S_3 \cap (SB_1 \cup SB_2)) \\ &\quad \cup (S_4 \cap (SB_1 \cup SB_2)) \\ &= \bigcup_{i=1}^4 \bigcup_{j=1}^2 (S_i \cap SB_j) \end{aligned} \tag{4.34}$$

where,

$$(S_1 \cap SB_1) = \left\{ q \in S \mid \max_{i \in \bar{B}}(v^i \cdot q) = \max_{j \in \bar{A}}(v^j \cdot q) \geq \max_{j \in AB}(v^j \cdot q) \right\} \tag{4.35}$$

$$(S_1 \cap SB_2) = \left\{ q \in S \mid \max_{i \in \bar{B}}(v^i \cdot q) = \max_{j \in \bar{A}}(v^j \cdot q) = \max_{j \in AB}(v^j \cdot q) \right\} \tag{4.36}$$

$$(S_2 \cap SB_1) = \left\{ q \in S \mid \max_{i \in \bar{A}}(v^i \cdot q) = \max_{j \in AB}(v^j \cdot q) \geq \max_{j \in \bar{B}}(v^j \cdot q) \right\} \tag{4.37}$$

$$(S_2 \cap SB_2) = \left\{ q \in S \mid \max_{i \in \bar{A}}(v^i \cdot q) = \max_{j \in AB}(v^j \cdot q) \geq \max_{j \in \bar{B}}(v^j \cdot q) \right\} \tag{4.38}$$

$$(S_3 \cap SB_1) = \left\{ q \in S \mid \max_{i \in \bar{A}}(v^i \cdot q) \geq \max_{i \in \bar{B}}(v^i \cdot q) \geq \max_{j \in AB}(v^j \cdot q), \right. \quad (4.39)$$

$$\left. \max_{i \in \bar{B}}(v^i \cdot q) = \max_{j \in \bar{A}}(v^j \cdot q) \right\} \quad (4.40)$$

$$(S_3 \cap SB_2) = \left\{ q \in S \mid \max_{i \in \bar{B}}(v^i \cdot q) = \max_{j \in AB}(v^j \cdot q), \right. \quad (4.41)$$

$$\left. \max_{i \in AB}(v^i \cdot q) \geq \max_{i \in \bar{B}}(v^i \cdot q) \geq \max_{j \in \bar{A}}(v^j \cdot q) \right\} \quad (4.42)$$

$$(S_4 \cap SB_1) = \left\{ q \in S \mid \max_{i \in \bar{B}}(v^i \cdot q) = \max_{j \in AB}(v^j \cdot q) = \max_{j \in \bar{A}}(v^j \cdot q) \right\} \quad (4.43)$$

$$(S_4 \cap SB_2) = \left\{ q \in S \mid \max_{i \in \bar{B}}(v^i \cdot q) = \max_{j \in AB}(v^j \cdot q) \geq \max_{j \in \bar{A}}(v^j \cdot q) \right\} \quad (4.44)$$

Note that (4.40) and (4.42) could be further reduced to:

$$(S_3 \cap SB_1) = \left\{ q \in S \mid \max_{i \in \bar{A}}(v^i \cdot q) = \max_{i \in \bar{B}}(v^i \cdot q) \geq \max_{j \in AB}(v^j \cdot q) \right\} \quad (4.45)$$

$$(S_3 \cap SB_2) = \left\{ q \in S \mid \max_{i \in AB}(v^i \cdot q) = \max_{i \in \bar{B}}(v^i \cdot q) \geq \max_{j \in \bar{A}}(v^j \cdot q) \right\} \quad (4.46)$$

Also,

$$(S_1 \cap SB_1) \cup (S_1 \cap SB_2) = \left\{ q \in S \mid \max_{i \in \bar{B}}(v^i \cdot q) = \max_{j \in \bar{A}}(v^j \cdot q) \geq \max_{j \in AB}(v^j \cdot q) \right\} \quad (4.47)$$

$$(S_2 \cap SB_1) \cup (S_2 \cap SB_2) = \left\{ q \in S \mid \max_{i \in \bar{A}}(v^i \cdot q) = \max_{j \in AB}(v^j \cdot q) \geq \max_{j \in \bar{B}}(v^j \cdot q) \right\} \quad (4.48)$$

$$(S_4 \cap SB_1) \cup (S_4 \cap SB_2) = \left\{ q \in S \mid \max_{i \in \bar{B}}(v^i \cdot q) = \max_{j \in AB}(v^j \cdot q) \geq \max_{j \in \bar{A}}(v^j \cdot q) \right\} \quad (4.49)$$

Now, substituting (4.45) through (4.49) into (4.34) one gets:

$$\begin{aligned}
S_K = & \left\{ q \in S \mid \max_{i \in A} (v^i \cdot q) = \max_{j \in B} (v^j \cdot q) \geq \max_{j \in AB} (v^j \cdot q) \right\} \\
& \cup \left\{ q \in S \mid \max_{i \in A} (v^i \cdot q) = \max_{j \in AB} (v^j \cdot q) \geq \max_{j \in B} (v^j \cdot q) \right\} \\
& \cup \left\{ q \in S \mid \max_{i \in B} (v^i \cdot q) = \max_{j \in AB} (v^j \cdot q) \geq \max_{j \in A} (v^j \cdot q) \right\}.
\end{aligned}$$

□

Now that we have defined the set S_K in terms of sets A and B , we can continue with the formulation of $f_1(A)$ and $f_1(B)$. Following the partition of the set S in (4.28), one can formulate $f_1(A)$ as the following:

$$\begin{aligned}
f_1(A) &= \int_S \max_{i \in A} (v^i \cdot q) dS \\
&= \int_{S_A} \max_{i \in A} (v^i \cdot q) dS + \int_{S_B} \max_{i \in A} (v^i \cdot q) dS + \int_{S_I} \max_{i \in A} (v^i \cdot q) dS + \int_{S_K} \max_{i \in A} (v^i \cdot q) dS
\end{aligned} \tag{4.50}$$

and

$$\begin{aligned}
f_1(B) &= \int_S \max_{i \in B} (v^i \cdot q) dS \\
&= \int_{S_A} \max_{i \in B} (v^i \cdot q) dS + \int_{S_B} \max_{i \in B} (v^i \cdot q) dS + \int_{S_I} \max_{i \in B} (v^i \cdot q) dS + \int_{S_K} \max_{i \in B} (v^i \cdot q) dS
\end{aligned} \tag{4.51}$$

Lemma 4.3.7. *In (4.50) $\int_{S_K} \max_{i \in A} (v^i \cdot q) dS = 0$, and in (4.51) $\int_{S_K} \max_{i \in B} (v^i \cdot q) dS = 0$.*

Proof. In Lemma 4.3.6 consider the sets whose union define S_K . All q belonging to the first one of these sets satisfy: $\max_{i \in A \setminus B} (v^i \cdot q) = \max_{j \in B \setminus A} (v^j \cdot q)$, and notice that $A \setminus B$ and $B \setminus A$ are disjoint sets. For this reason, for an n dimensional probability simplex the set of q that satisfies this equality defines an $n - 2$ dimensional hyperplane. Similarly, q belonging to the other sets also define an $n - 2$ dimensional hyperplane. The integral defined over all this lower dimensional hyperplanes becomes zero (an example of this situation is a line integral evaluated at a countable number of points). □

Also, notice that because of the way we have defined S_A in (4.24) and S_I in (4.26) we can write:

$$\int_{S_A} \max_{i \in A} (v^i \cdot q) dS = \int_{S_A} \max_{i \in A \setminus B} (v^i \cdot q) dS, \quad \text{and} \quad (4.52)$$

$$\int_{S_I} \max_{i \in A} (v^i \cdot q) dS = \int_{S_I} \max_{i \in A \cap B} (v^i \cdot q) dS \quad (4.53)$$

Then, following Lemma 4.3.7 and identities (4.52) and (4.53), (4.50) could be reduced to:

$$f_1(A) = \int_S \max_{i \in A} (v^i \cdot q) dS = \int_{S_A} \max_{i \in A \setminus B} (v^i \cdot q) dS + \int_{S_B} \max_{i \in A} (v^i \cdot q) dS + \int_{S_I} \max_{i \in A \cap B} (v^i \cdot q) dS \quad (4.54)$$

Similarly, because of the way we have defined S_B in (4.25) and S_I in (4.26) we can write:

$$\int_{S_B} \max_{i \in B} (v^i \cdot q) dS = \int_{S_B} \max_{i \in B \setminus A} (v^i \cdot q) dS, \quad \text{and} \quad (4.55)$$

$$\int_{S_I} \max_{i \in B} (v^i \cdot q) dS = \int_{S_I} \max_{i \in A \cap B} (v^i \cdot q) dS \quad (4.56)$$

Then, following Lemma 4.3.7 and identities (4.55) and (4.56), (4.51) could be reduced to:

$$f_1(B) = \int_S \max_{i \in B} (v^i \cdot q) dS = \int_{S_A} \max_{i \in B} (v^i \cdot q) dS + \int_{S_B} \max_{i \in B \setminus A} (v^i \cdot q) dS + \int_{S_I} \max_{i \in A \cap B} (v^i \cdot q) dS \quad (4.57)$$

Considering (4.54) and (4.57):

$$\begin{aligned} f_1(A) + f_1(B) &= \int_{S_A} \max_{i \in A \setminus B} (v^i \cdot q) dS + \int_{S_B} \max_{i \in A} (v^i \cdot q) dS + \int_{S_A} \max_{i \in B} (v^i \cdot q) dS \\ &\quad + \int_{S_B} \max_{i \in B \setminus A} (v^i \cdot q) dS + 2 \int_{S_I} \max_{i \in A \cap B} (v^i \cdot q) dS \end{aligned} \quad (4.58)$$

Meanwhile;

$$\begin{aligned} f_1(A \cup B) &= \int_S \max_{i \in A \cup B} (v^i \cdot q) dS = \int_{S_A} \max_{i \in A \cup B} (v^i \cdot q) dS + \int_{S_B} \max_{i \in A \cup B} (v^i \cdot q) dS \\ &\quad + \int_{S_I} \max_{i \in A \cup B} (v^i \cdot q) dS \end{aligned} \quad (4.59)$$

,and following definitions of S_A in (4.24), S_B in (4.25) and S_I in (4.26) we have:

$$\int_{S_A} \max_{i \in A \cup B} (v^i \cdot q) dS = \int_{S_A} \max_{i \in A \setminus B} (v^i \cdot q) dS, \quad (4.60)$$

$$\int_{S_B} \max_{i \in A \cup B} (v^i \cdot q) dS = \int_{S_B} \max_{i \in B \setminus A} (v^i \cdot q) dS, \quad (4.61)$$

$$\int_{S_I} \max_{i \in A \cup B} (v^i \cdot q) dS = \int_{S_I} \max_{i \in A \cap B} (v^i \cdot q) dS \quad (4.62)$$

With Lemma 4.3.7 and identities (4.60), (4.61), and (4.62), (4.59) could be reduced to:

$$f_1(A \cup B) = \int_{S_A} \max_{i \in A \setminus B} (v^i \cdot q) dS + \int_{S_B} \max_{i \in B \setminus A} (v^i \cdot q) dS + \int_{S_I} \max_{i \in A \cap B} (v^i \cdot q) dS \quad (4.63)$$

On the other hand,

$$f_1(A \cap B) = \int_S \max_{i \in A \cap B} (v^i \cdot q) dS = \int_{S_A} \max_{i \in A \cap B} (v^i \cdot q) dS + \int_{S_B} \max_{i \in A \cap B} (v^i \cdot q) dS + \int_{S_I} \max_{i \in A \cap B} (v^i \cdot q) dS \quad (4.64)$$

Summing (4.63) with (4.64):

$$f_1(A \cup B) + f_1(A \cap B) = \int_{S_A} \max_{i \in A \setminus B} (v^i \cdot q) dS + \int_{S_B} \max_{i \in B \setminus A} (v^i \cdot q) dS + \int_{S_A} \max_{i \in A \cap B} (v^i \cdot q) dS + \int_{S_B} \max_{i \in A \cap B} (v^i \cdot q) dS + 2 \int_{S_I} \max_{i \in A \cap B} (v^i \cdot q) dS \quad (4.65)$$

Comparing (18 & 22) one can notice that some of the terms are common in both equations. Leaving out those common terms, suboptimality condition would be equivalent to the following inequality:

$$\begin{aligned}
f_1(A) + f_2(B) &\geq f_1(A \cup B) + f_1(A \cap B) \\
&\Downarrow \\
\int_{S_B} \max_{i \in A} (v^i \cdot q) dS + \int_{S_A} \max_{i \in B} (v^i \cdot q) dS &\geq \int_{S_A} \max_{i \in A \cap B} (v^i \cdot q) dS + \int_{S_B} \max_{i \in A \cap B} (v^i \cdot q) dS
\end{aligned} \tag{4.66}$$

Since $(A \cap B) \subseteq A$ and $(A \cap B) \subseteq B$, the equivalent inequality always holds proving the submodularity of $f_1(\cdot)$ and $f_\infty(\cdot)$. \square

Although f_1 function inherits the submodularity property, it isn't a suitable candidate for the greedy algorithm based pruning, and for this reason we omit it in our analysis. Now that we have proved the submodularity of the function f_1 , we are ready to give a bound on the degree of suboptimality related to the greedy algorithm for finding the pruned set. If the algorithms were run without any limit on $\bar{\epsilon}$, and only having limitations on \bar{n} , then by the results on optimization of submodular functions of [NW78], one can guarantee bounds on the greedy algorithm results. In that study, the authors demonstrate that the greedy algorithm cannot result in a value of less than $(e - 1)/e \approx 0.63$ of the value the optimal pruned set would yield. This immediately implies the same bound on the suboptimality of our $W(\cdot, q)$. Of course this bound is quite conservative, as it is completely general. Meanwhile it can still be considered as a way to keep track of the error following greedy algorithm.

Another way to address the error bound could be utilization of the L_∞ error after the greedy and stingy algorithms. Since ϵ_∞ could be easily calculated utilizing the linear programming methods, one could calculate the error resulting from such approaches based on L_∞ . Since ϵ_∞ was found to define an upper bound on ϵ_1 , one could find a better error bound after such analysis.

Chapter 5

Conclusion and Future Work

The main theme of this research was to develop a new methodology for optimally tasking the sensing assets deployed for information gathering in uncertain and adversarial environments. The dissertation can be divided into three parts although all are very related to each other; development of a new measure for reconnaissance missions, open loop and state feedback control approaches to the optimal sensor tasking problem that utilizes this new measure, and numerical methods to improve computation times of the proposed approach.

Development of this new measure is based on the observation that although uncertainty should be eliminated during reconnaissance missions, the trade-off of where to eliminate the uncertainty depends on how uncertainty is going to affect the operations that will follow-up the reconnaissance missions. For this reason, the new measure for the reconnaissance missions is based on the benefit the team who is doing the reconnaissance is receiving. A Markovian type model is developed for an upcoming operation that would succeed the reconnaissance mission. A payoff function is defined for this operation such that it was dependent on the available information.

This payoff was related to the payoff of the sensing operations and the optimal control options for the sensors was found. An open loop analysis yielded significant increases for an analyzed scenario. Closed loop analysis was subject to curse of dimensionality. First a max-plus analysis was used to avoid grid based analysis. Although it provided the us the opportunity to avoid grid based methods,

it was again suffering from long computation times.

Computation times were improved using refining and pruning methodologies. Refining methods utilized linear programming to detect inactive objects stored and progressed during DP iterations. Pruning methods gave us an option to find approximated solutions with faster computation times.

Bibliography

- [Bel57] Richard E. Bellman. *Dynamic Programming*. Princeton University Press, 1957.
- [Ble10] Ariel Bleicher. Eyes in the sky that see too much. *IEEE Spectrum*, 47(10):16, 2010.
- [Boy03] Walter J. Boyne. *The Influence of Air Power upon History*. Pelican Publishing Company, 2003.
- [BS96] Donald Barr and Todd E. Sherrill. Measuring information gain in tactical operations. Technical report, Operations Research Center USMA, West Point, NY 10996, July 1996.
- [Buc99] John Buckley. *Air Power in The Age of Total War*. Indiana University Press, 1999.
- [CG79] R.A. Cuninghame-Green. *Minimax Algebra, Lecture Notes in Economics and Mathematical Systems 166*. Springer, 1979.
- [Clo02] Bruce Clough. Unmanned aerial vehicles: Autonomous control challenges, a researchers perspective. In *Cooperative Control and Optimization*, volume 66, pages 35–52. 2002.
- [Dan62] John M. Danskin. A theory of reconnaissance: I. *Operations Research*, 10(3):pp. 285–299, 1962.
- [DO05] John G. Drew and OTHERS. *Unmanned Aerial Vehicle End-to-End Support Considerations*. RAND Corporation, 2005.
- [FBQ92] G.J. Olsder F.L. Baccelli, G. Cohen and J.-P. Quadrat. *Synchronization and Linearity*. John Wiley, 1992.
- [HL05] Frederick S. Hillier and Gerald J. Lieberman. *Introduction to Operatins Research*. McGraw Hill, 2005.

- [HSL⁺92] Michael R. Hilliard, Rajendra S. Solanki, Cheng Liu, Ingrid K. Busch, Glen Harrison, and Ronald D. Kraemer. Scheduling the operation desert storm airlift: An advanced automated scheduling support system. *Interfaces*, 22(1):131–146, 1992.
- [KK81] Ik-Whan Kwon and Joe H. Kim. Sequential decision-making process under conditions of uncertainty in drug analysis. *Journal of Chemical Education*, 58(6):462–464, 1981.
- [KM97] V.N. Kolokoltsov and V.P. Maslov. *Idempotent Analysis and Its Applications*. Kluwer, 1997.
- [KO07] Alexander Kott and Michael Ownby. Tools for real-time anticipation of enemy actions in tactical ground operations. In *10th International Command and Control Research and Technology Symposium*, 2007.
- [McE06] W.M. McEneaney. *Max-Plus Methods for Nonlinear Control and Estimation*. Birkhauser, 2006.
- [McE09] W.M. McEneaney. Complexity reduction, cornices and pruning. In *Proc. of the International Conference on Tropical and Idempotent Mathematics*, 2009.
- [NW78] G.L. Nemhauser and L. A. Wolsey. An analysis of approximations for maximizing submodular set functions-i. *Mathematical Programming*, 14(1):265–294, 1978.
- [PC98] M. Pachter and P.R. Chandler. Challenges of autonomous control. *Control Systems Magazine, IEEE*, 18(4):92–97, August 1998.
- [ST01] A. S. Sheynin and A. V. Tuzikov. Formulae for polytope volume and surface moments. In *Proceedings of International Conference on Image Processing*, 2001.
- [Whi75] D. J. White. Entropy and decision. *Operational Research Quarterly (1970-1977)*, 26(1):pp. 15–23, 1975.
- [YY10] Xiaolei Yu and Hongmin Yu. A novel low-altitude reconnaissance strategy for smart UAVs: active perception and chaotic navigation. *Transactions of the Institute of Measurement and Control*, 2010.