5-2013

Surveillance Planning against Smart Insurgents in Complex Terrain

Nabil Lehlou
University of Arkansas, Fayetteville

SURVEILLANCE PLANNING AGAINST SMART INSURGENTS IN COMPLEX TERRAIN

SURVEILLANCE PLANNING AGAINST SMART INSURGENTS IN COMPLEX TERRAIN

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Industrial Engineering

By

Nabil Lehlou
University of Arkansas
Bachelor of Science in Computer Science, 2007
University of Arkansas
Master of Science in Industrial Engineering, 2008

May 2013
University of Arkansas

ABSTRACT

This study is concerned with finding a way to solve a surveillance system allocation problem based on the need to consider intelligent insurgency that takes place in a complex geographical environment. Although this effort can be generalized to other situations, it is particularly geared towards protecting military outposts in foreign lands. The technological assets that are assumed available include stare-devices, such as tower-cameras and aerostats, as well as manned and unmanned aerial systems. Since acquiring these assets depends on the ability to control and monitor them on the target terrain, their operations on the geo-location of interest ought to be evaluated. Such an assessment has to also consider the risks associated with the environmental advantages that are accessible to a smart adversary. Failure to consider these aspects might render the forces vulnerable to surprise attacks. The problem of this study is formulated as follows: *given a complex terrain and a smart adversary, what types of surveillance systems, and how many entities of each kind, does a military outpost need to adequately monitor its surrounding environment?* To answer this question, an analytical framework is developed and structured as a series of problems that are solved in a comprehensive and realistic fashion. This includes digitizing the terrain into a grid of cell objects, identifying high-risk spots, generating flight tours, and assigning the appropriate surveillance system to the right route or area. Optimization tools are employed to empower the framework in enforcing constraints—such as fuel/battery endurance, flying assets at adequate altitudes, and respecting the climbing/diving rate limits of the aerial vehicles—and optimizing certain mission objectives—e.g. revisiting critical regions in a timely manner, minimizing manning requirements, and maximizing sensor-captured image quality. The framework is embedded in a software application that supports a friendly user interface, which includes the visualization of maps, tours, and related statistics. The final product is expected to support designing surveillance plans for remote military outposts and making critical decisions in a more reliable manner.

This dissertation is approved for recommendation
to the Graduate Council.

Dissertation Director:

_____
Dr. Nebil Buyurgan

Dissertation Committee:

_____
Dr. Ed Pohl

_____
Dr. Ronald Rardin

_____
Dr. Randall Steeb

DISSERTATION PUBLICATION RELEASE

I hereby authorize the University of Arkansas Libraries to duplicate this dissertation when needed for research and/or scholarship.

Agreed _____
Nabil Lehlou

Refused _____
Nabil Lehlou

ACKNOWLEDGEMENTS

The completion of my doctoral studies would not have been possible without the dedication and support of several people. I would like to first and foremost thank my graduate advisor, Dr. Nebil Buyurgan, who is also the chairman of my Ph.D. dissertation as well as my Master's thesis. He has not only been an excellent scholastic guide and mentor, but a dear friend of mine as well. He has always challenged me to think hard and reevaluate my analytical methods, provided me with insight and advice beyond school and academia, and shared with me his personal memories and family moments.

I would like to give my warmest thanks to my benign committee members. Thanks to Dr. Ed Pohl for being a great teacher; I have learned from him valuable and tremendously useful concepts and practices in the fields of reliability, non-linear programming, and project management. Thanks to Dr. Rardin for sharing his stupendous research experience with his students and graduate assistants, and for letting me be part of the Center for Innovation in Healthcare Logistics; working under his direct leadership was both enjoyable and enlightening. Many thanks to Dr. Randall Steeb for he provided me with one of the greatest learning experiences. He bestowed upon me invaluable guidance and expertise in the subject matter of this research effort, and showed great patience and talent in addressing the issues I have raised throughout this journey of solving a hard problem. He promptly answered all my questions to a great degree of satisfaction and was the core source of my understanding of the surveillance topic.

Besides Randall Steeb, I would like to extend my thanks to RAND Corporation, and to Bruce Held in particular, for giving me the honorable chance to intern at and work with the RAND Force Protection team on a great research topic. I thank Kenneth Horn for his hospitality and kind accommodating supervision; he made sure I was not lacking any resource or access during my internship. I thank Brian Chow, who inspired my approach of modeling terrain and long-range attacks, Lewis Jamison, who

helped me understand how to carry out airborne surveillance that targets smart insurgents, and Matt Boyer, who provided me with terrain and weather data. Without this insightful group of people, I could not have developed and validated my analytical framework and reached the point where I am today.

I thank the University of Arkansas for its genuine support and dedicated service during both my undergraduate and graduate studies. At this institution, I learned how to think like an entrepreneur, act as a leader, solve problems like a scientist, and build systems like an engineer. Also, special thanks to my other professors who helped inspire me and had a positive impact on my life, including Dr. Amy Apon, Dr. Gordon Beavers, Dr. Richard Cassady, Dr. Justin Chimka, Dr. John Lusth, Dr. Carol Reeves, Dr. Manuel Rossetti, Dr. Greg Starling, Dr. Craig Thompson, Dr. Janet Vassilev, and Dr. John White. In addition, I thank Michael Freeman, Audra Johnston, Cynthia Smith, Veronikha Salazar, and the rest of the International Students and Scholars Office staff for providing international students with a warm community through its ICT program, assisting us financially through the FIES scholarships, and ensuring that our immigration statuses have always been in line with the most recent governmental requirements.

Sincere thanks to all my friends and family members for their love, prayers, and encouragement. Special thanks to my mother in law, Dr. Maria Dadi, for her exceptional motivating words, and to my mom and dad for their loving support. Immense thanks to my beloved and caring wife, Khaoula Charahbili Harouchi, for her trust in me; she has been outstandingly understanding and supportive throughout this difficult journey. Regardless of how homesick she feels or how busy she gets with school and work, she always ensures that I am productive, healthy, and well nurtured. Furthermore, she never seizes from smiling and hoping for the best; she is my true source of energy!

Lastly, thank you God for life, health, energy, the ability to persevere, and the opportunity to meet wonderful and inspiring people that have contributed to my endeavor through teaching, motivation, and leadership by example.

TABLE OF CONTENTS

Abstract ii
Dissertation Publication Release iv
Acknowledgements v
Table of Contents viii
List of Figures x
List of Tables xii
List of Acronyms xiii
Chapter 1: Introduction 1
1. Introduction 1
2. Motivation 2
3. Problem Description 4
Chapter 2: Background 8
1. Technologies 8
2. Literature Review 13
2.1. Space Representation and Assessment 14
2.2. Counter-Insurgency 17
2.3. UAV Employment and Mission Planning 19
2.4. Modeling and Algorithms 23
2.5. Conclusion and Contribution 25
Chapter 3: Methodology 27
1. Concept 27
2. Analytical Framework 30
3. Performance Measures 34
Chapter 4: Terrain Risk Assessment 40
1. Background 40
1.1. Fuzzy Set Theory 40
1.2. Risk Terrain Modeling 41
2. Risk-Factors 44
2.1. Foliage Camouflage 45
2.2. Proximity to Population 46
2.3. Long-Range Attacks 48
2.4. Accessibility to Regions 51
3. Composite Map 56
Chapter 5: Tour Generation 58
1. Concept 58
2. Network Reduction 63
3. Tour Construction 66
3.1. TSP Heuristic 66
3.2. SPP Grid Reduction 70

4. Altitude Optimization 73
5. Tour Segmentation 79
Chapter 6: System Allocation 84
1. Tactical Systems 86
1.1. Building the Solution 86
1.2. Refining the Solution 88
1.3. Scaling the Solution 92
2. Stationary Systems 94
2.1. Problem Complexity 97
2.2. Solution Process 99
2.3. Coverage and Overlap Computations 101
2.4. Sensor Placement 111
2.5. Sensor Placement with Aerostat 115
3. Strategic Systems 117
Chapter 7: Case Study 121
1. Background 121
2. Analysis 124
2.1. Input Data 125
2.2. Terrain Risk Assessment 128
2.3. Tour Generation 133
2.4. Resource Allocation 137
3. Conclusion 140
4. Dissertation Summary 142
References 147
Appendix 151
1. AMPL Model for TSP 151
2. Pseudo-Code for TSP Heuristic 151
3. AMPL Model for SPP 152
4. AMPL Model for OAP 152
Example of Input Data: 153
5. Pseudo-Code for the Tour Segmentation Algorithm 153
6. AMPL Model for TBSCP 154
7. AMPL Model for TBVAP 154
Example of Input Data: 155
8. Pseudo-Code for the SSPP Algorithm 155

LIST OF FIGURES

Figure 1: A surveillance tower 9
Figure 2: An LTA Vehicle (airship) 9
Figure 3: The RQ-11 Raven 10
Figure 4: The ScanEagle 10
Figure 5: The RQ-7 Shadow 11
Figure 6: The MQ-1 Predator 11
Figure 7: The Constant Hawk 12
Figure 8: A region modeled through a TIN 15
Figure 9: Dividing the 3D space into a voxel-grid 16
Figure 10: The three-way assignment model in a surveillance portfolio scheme 27
Figure 11: The high-level architecture of the analytical framework 31
Figure 12: A geometric rep-resentation of the angle θ 35
Figure 13: A geometric representation of the slope coverage model for a cell 36
Figure 14: The geometric situation for Case 1 (i.e. $d_A + h_{AB} < \text{minOA}$) 38
Figure 15: The geometric situation for Case 2 (i.e. $h_{AB} < d_A$) 38
Figure 16: The geometric situation for Case 3 (i.e. $h_{AB} < \text{minOA}$ and $d_A < \text{minOA}$) 38
Figure 17: The geometric situation for Case 4 (i.e. $\text{minOA} < h_{AB}$) 38
Figure 18: Process of adding up risk-factors to obtain the composite map 42
Figure 19: A composite map that incorporates the three risk-factors for the Irvington, NJ study 42
Figure 20: A grid setup where the outpost and the target cell j are in the western half of the grid 46
Figure 21: An example of a terrain grid with four populated cells 47
Figure 22: Illustration of the three ideal conditions and their associated parameters 48
Figure 23: Two examples of finding intermediary cells 49
Figure 24: An Illustration of a cell causing obstruction 50
Figure 25: The fuzzy membership function in terms of elevation difference h_{jO} (in meters) 51
Figure 26: The fuzzy membership function F_β given that $\beta_o = 45^o$ and $\beta_{max} = 60^o$ 52
Figure 27: Trafficability assessment and categorization for dry terrain 53
Figure 28: Trafficability assessment and categorization for moist terrain 54
Figure 29: Trafficability assessment and categorization for a full-saturation terrain 54
Figure 30: The fuzzy membership function in terms of trafficability t_j 55
Figure 31: The process used in terrain evaluation in order to generate the composite map 56
Figure 32: The network infrastructure obtained from transforming the 2D digitization grid 58
Figure 33: Synthetic flight tours planned for the RQ-7 Shadow (no infeasible spots) 59
Figure 34: Synthetic flight tours planned for the RQ-11 Raven (two infeasible spots) 59
Figure 35: The sequence of steps to generate (sub)tours for a UAV 62
Figure 36: Three network infrastructures associated with three different UAVs 64
Figure 37: The reduced network for the RQ-7 Shadow in a 5x5 km^2 Wanat region 65
Figure 38: Dividing the grid into two subgrids in order to solve two smaller TSPs 67
Figure 39: The process of removing the breakpoints and connecting the two subtours 68

Figure 40: A subgrid comprising the origin, destination, and shortest path 72
Figure 41: A subgrid comprising the origin and destination, but not the shortest path 72
Figure 42: The geometric situation at a given cell 75
Figure 43: Example mimicking a hotspot 78
Figure 44: A snapshot during the process of expanding a subtour by the finishing algorithm 81
Figure 45: The computation process of the traveled distance for inspection (D_{ins}) 82
Figure 46: Subtour input data to the Set Covering Problem 87
Figure 47: A TBSCP solution in which one hotspot (in red) requires a high revisit-frequency 89
Figure 48: An example of five subtours that are assigned five UAVs 91
Figure 49: Process of building the surveillance plan for tactical systems 93
Figure 50: A digitized map of an outpost where each grid cell is assigned a stationary sensor 95
Figure 51: A way to envelop the irregular perimeter of the outpost with a surveillance tape 95
Figure 52: A solution example with multiple sensor ranges and allowance for lack of coverage 96
Figure 53: Steps of the solution process to select the best stationary surveillance alternative 99
Figure 54: The representation of the LTA's sensor FOV for three different levels of surveillance 102
Figure 55: The FOV representation for a tower camera at three different levels of surveillance 103
Figure 56: Triming the circular shape of the tower-camera's coverage to obtain a square 104
Figure 57: Three cases for positioning the squared shape and L-shape and their levels for overlap 104
Figure 58: Two overlapped sensor ranges positioned over a segment of the virtual wall 106
Figure 59: The four scenarios of overlap with regards to the width w_{ij} and x-coordinate x_{ij} 107
Figure 60: A solution for a SSPP instance where four sensors suffice to provide apt coverage 112
Figure 61: A solution for a SSPP instance where a layer of FOVs suffices to provide apt coverage 113
Figure 62: A solution for a SSPP instance where two layers of FOVs are needed for apt coverage 113
Figure 63: The transition from a solution for A1 to a solution for A2 116
Figure 64: The geometric situation of two overlapping circular ranges with r_1 and r_2 as radii 116
Figure 65: A circular surface area with a spiral flight pattern 118
Figure 66: A rectangular surface area with a serpentine flight patterns 118
Figure 67: A surveillance plan for ten regions serviced by three strategic vehicles 119
Figure 68: The proposed layout of COP Kalher 122
Figure 69: A view of the Wanat village from the north 123
Figure 70: The 50x50 Digitized terrain of the Wanat region 126
Figure 71: Visualization of average altitude data (m) 127
Figure 72: Visualization of slope data (deg) 127
Figure 73: Visualization of wind-levels data (m/s) 127
Figure 74: Visualization of the risk-factor associated with foliage camouflage 128
Figure 75: Visualization of the risk-factor associated with proximity to population 128
Figure 76: Visualization of the risk-factor associated with long-range attacks 129
Figure 77: Visualization of the risk-factor associated with accessibility to regions 129
Figure 78: Profile 1's composite map 130
Figure 79: Profile 2's composite map 130
Figure 80: Profile 3's composite map 130
Figure 81: Light surveillance with risk>0.6 yields 57 hotspots 131

Figure 82: Moderate surveillance with risk>0.55 yields 99 hotspots 131
Figure 83: Heavy surveillance with risk>0.5 yields 195 hotspots 131
Figure 84: Light surveillance with risk>0.6 yields 68 hotspots..... 131
Figure 85: Moderate surveillance with risk>0.55 yields 107 hotspots 131
Figure 86: Heavy surveillance with risk>0.5 yields 150 hotspots 131
Figure 87: Light surveillance with risk>0.7 yields 72 hotspots..... 131
Figure 88: Moderate surveillance with risk>0.65 yields 99 hotspots 131
Figure 89: Heavy surveillance with risk>0.6 yields 242 hotspots 131
Figure 90: Raven's feasible tour to visit the reachable hotspots..... 134
Figure 91: Raven's feasible tour paths (orange), links (cyan), and other traversable edges (black) 134
Figure 92: Raven's OA and the terrain elevation (ASL) based on 127 datapoints (in meters) 134
Figure 93: Shadow's feasible tour to visit the reachable hotspots..... 135
Figure 94: Shadow's feasible tour paths (orange), links (cyan), and other traversable edges (black) 135
Figure 95: Shadow's OA and the terrain elevation (ASL) based on 274 datapoints (in meters) 135
Figure 96: ScanEagle's feasible tour to visit the reachable hotspots 136
Figure 97: ScanEagle's feasible tour paths (orange), links (cyan), and other traversable edges (black).. 136
Figure 98: ScanEagle's OA and the terrain elevation (ASL) based on 315 datapoints (in meters) 136
Figure 99: The layout of sensor FOVs (in blue) on the 50-meter-thick virtual wall (in green) 140

LIST OF TABLES

Table 1: Platforms and their associated motion-based attributes 12
Table 2: Platforms and their associated attributes that are not motion-based..... 13
Table 3: Sensors and their associated attributes..... 13
Table 4: Legend for Figure 27, Figure 28, and Figure 29..... 55
Table 5: Types of involved constraints and their stage of integration 60
Table 6: A comparison of the heuristic against CPLEX in terms of solution quality and runtime 70
Table 7: Results of running the OAP model on the input data of the above example 79
Table 8: The factors involved in determining the appropriate analysis for a certain type of assets..... 86
Table 9: Sensitivity analysis on the revisit-time with the reduction of needed assets as an outcome..... 92
Table 10: Parameter values for the Wanat case study..... 128

LIST OF ACRONYMS

AI	Artificial Intelligence
AAF	Anti-Afghan Forces
ANA	Afghan National Army
CAD	Computer-Aided Design
CC	Coverage Contribution
COIN	Counter-Insurgency
COP	Combat Outpost
FOB	Forward Operating Base
FOV	Field-of-View
FST	Fuzzy Set Theory
GMTI	Ground Moving Target Indicator
IED	Improvised Explosive Device
IP	Integer Programming
ISR	Intelligence, Surveillance, and Reconnaissance
LAP	Linear Assignment Problem
LRA	Long-Range Attacks
LTA	Lighter-than-Air
MCP	Maximal Covering Problem
MIP	Mixed Integer Program
MP	Mathematical Programming
NAI	Named Area of Interest
O&M	Operations and Maintenance
OA	Operating Altitude
OAP	Operating Altitude Problem
OP	Observation Post
OR	Operations Research
SCP	Set Covering Problem
SAR	Surveillance Approach Radar
SIGINT	Signals Intelligence
SPP	Shortest Path Problem
SSPP	Stationary System Positioning (or Placement) Problem
SAM	Surface-to-Air Missile
STK	Systems Tool Kit
TSP	Traveling Salesman Problem
TBSCP	Time-Based Set Covering Problem
TBVAP	Time-Based Vehicle Assignment Problem
UAV	Unmanned Aerial Vehicle
UAS	Unmanned Aerial System

CHAPTER 1: INTRODUCTION

1. Introduction

When the U.S. armed forces deploy troops in foreign lands, they use *Forwarding Operating Bases* (FOBs) as military settlements to support their operational, medical, logistical, and usual daily needs (Miles, 2006; The Washington Times, 2008). FOBs represent the main stronghold (in a certain region) where most of the soldiers, equipment, and ammunitions are present. They are usually well protected against direct nearby attacks through fortified walls and obstacles, security patrols, and sophisticated surveillance assets. The military also uses temporary bases, called outposts, in remote regions to attain wider area protection. These outposts can be categorized as: (1) a *Combat Operating Post* (COP), which lasts from weeks to months, and normally, has the size of a platoon (25-60) or a company (70-250); or (2) an *Observation Post* (OP), which stays in theater for days and usually has a size of a squadron (8-15), sometimes a platoon.

However, once on foreign territory, the troops face the challenge of being unfamiliar with the geographical settings of the theaters as well as the culture and intellect of the inhabitants. To be more specific, outposts are sometimes located in disadvantageous regions where the terrain is rough, the villagers sympathize with the insurgents, and the transportation network is distant. This renders insurgent activities hard to predict, and sometimes, difficult to encounter. In fact, the U.S. Army has indeed sustained damage from well-orchestrated attacks on COP/OPs, such as the case of COPs Keiting (McCullough, 2010) and Kahler (Ross, 2010). Although the minimal protection of the outpost may have been a contributor to the failed defense, it is believed that the structure of the geographical location and the fight against swift insurgents have also played a significant role (Dixon, 2009).

To prevent such incidents, the U.S. army is interested in investing in surveillance systems in general (which, for the sake of this research effort, can be understood as platforms—either free-flying or

stationary—that are equipped with sensors), and in unmanned aerial vehicles (UAVs) in particular, to provide adequate protection to outposts (UAS-COE, 2010). First, airborne systems usually present an attractive solution as they offer *full-motion video* (FMV) long-distance monitoring and wide area covering without being noticed. Second, there is a subtle driver for using tactical UAVs as they are small enough to be organic to the outposts, and hence, are able to mitigate the length of the *Air Tasking Order* (ATO) cycle[1], a process criticized for its inefficiency by Winkler (2006) and Downs (2007). The strategy of long-distance surveillance can give enough time to provide early warnings, and even timely backups, upon the detection and identification of pre-insurgency activities. This scheme is even more appealing in hostile theaters where aerial technology is not available to the adversary.

The challenge of implementing this strategy is the existence of a number of constraining factors. For instance, mountains and elevated lands restrict the flight of some unmanned aerial systems (UASs). Also, harsh weather has the potential to decrease sensor visibility and cause flight disruption. Furthermore, there is a tradeoff between covering a large area and providing good image quality for detection and identification. Therefore, the selection from a wide variety of sensors and platforms, each with a different set of limitation and capabilities, can become a cumbersome task for the person responsible for designing the surveillance plan—the surveillance architect.

2. Motivation

This work is motivated by the importance and the degree of complexity of the target surveillance planning problem as well as the inefficiency of the currently-used methods to solve it. The problem addressed in this study, although great in magnitude, is simple to understand: *given a complex*

[1] An ATO cycle is the process of allotting resources to a mission involving an air campaign.

terrain and a smart adversary, what types of surveillance systems, and how many entities of each kind, does an outpost need to adequately monitor its surrounding environment?

On the one hand, finding the best combinations of UASs and other systems (ground and manned systems) to provide persistent surveillance to outposts is a critical task. This process however depends on several external aspects such as understanding the geographical and meteorological aspects of the theater, the potential routes to be taken by the insurgents, predicting enemy attack strategies, and applying tactical expertise in the surveillance plan design. Failure to consider these points might render the U.S. force vulnerable to surprise attacks.

On the other hand, the performance of an adequate study of a certain site requires multiple skillful (and costly) personnel as well as a long period of time to achieve the objective. Sophisticated simulation tools are typically used to integrate and investigate the behavior of insurgents, villagers, and soldiers under different weather circumstances on impressively practical representations of the terrain. Furthermore, both weapons and paths are incorporated as variables in the equation to render the simulated model even more realistic. Because of its complexity, the model can require several months to be developed, tested, and applied on different settings; not to forget that each simulation run usually lasts for days. It is also very possible that many situations and plan attacks can be ignored due to time and cost constraints. For more information on work and trends on simulation and artificial intelligence (AI), see Sidran's (2007) index.

Instead of using simulation, this research effort proposes the use of optimization and constraint satisfaction tools, as mathematical and algorithmic methods are momentously faster. Moreover, those can be integrated in a software program to automate a great part of the analysis, rendering the process even quicker. This gives researchers and surveillance architects more time to test the reliability of the surveillance portfolio by applying a wide range of attack vignettes through sensitivity analysis and similar

techniques. Yet, unlike simulation, which is a direct mimicry of the real world (with a certain degree of details), the use of mathematical and algorithmic methods require robust structuring of the problem. Such a task can be complex as constraints and objective functions have to be formulated and linked in a simplified and nonlinear fashion in order to achieve a practical runtime.

The contribution of this study is concerned with constructing an analytical framework to provide an answer to the question established above, a question that presents a set of challenges that have yet to be adequately addressed in the literature. In Chapter 2 Section 2.5, this contribution is elaborated upon after a review of previous work in the relevant fields. Also, the result of this research is integrated in a software decision-support tool that permits the surveillance architect to build and evaluate surveillance plans in an efficient manner. To achieve these goals, four high-level steps are followed: (1) acquiring the necessary knowledge and data to investigate and understand the quantitative aspects of the problem; (2) framing the problem to establish the foundation for analytical modeling; (3) constructing an analytical framework to solve the problem and analyze the solution given any set of data inputs; and (4) developing a software program[2] to provide an easy interface to the decision-support tool.

3. Problem Description

In this section, the implications that are generated as a result of trying to answer the research question are discussed. The first direct implication is that there is a need to acquire technological assets that are necessary to appropriately surveil the long-distance surroundings of the outpost. Before

[2] *Denidyzer* (a detection and identification analyzer) is a Java software application designed to help surveillance architects build and analyze surveillance portfolios. To download it or learn more about it, visit http://comp.uark.edu/~nlehlou/denidyzer/ or contact the author at nlehlou@gmail.com.

performing such an acquisition, it is wise to identify the right system to service the right region because of the existence of technological and environmental constraints.

Firstly, from a technological point of view, there are limits on the capabilities of surveillance systems. Sensors ought to be assigned to the proper platform as there are payload and compatibility constraints. Additionally, each airborne vehicle has specific maximum and minimum altitudes that are not to be crossed. It has explicit maximum climb- and dive-rates (rates for the altitude increase and decrease, respectively) that it cannot exceed. It has limits associated with crosswind resistance and endurance, depending on its size and battery or fuel consumption. Also, when flying above a population, the aerial device should be above a certain distance from the ground, called the *dB-distance* (or decibel-distance), from which its noise can be heard, and hence reveal its presence.

Secondly, depending on the type of surveillance mission, an airborne device can be categorized as a *strategic* or a *tactical* system. Strategic assets, also named *Wide-Area Surveillance* (WAS) systems, can fly at high altitudes and swipe wide areas with a wide *field of view* (FOV), but their image resolution is usually low; this may restrict their use to WAS search and detection specifically. For identification, which requires higher resolution, the platform's operating altitude (OA) can be reduced, or the onboard sensors may be zoomed on the target; these two options are not always possible though. Instead, these systems can use cross-cueing, which is the passing of information to another sensor. Such devices are usually onboard a tactical asset that is dedicated to identification and classification missions. This is due to the fact that tactical UAVs are small and able to fly at low altitudes, and thus can capture good quality images. This segregation of strategic and tactical systems mitigates a certain degree of the aforementioned tradeoff between the large-swath wide-FOV and the accurate narrow-FOV. In this study, the task of monitoring routes is assigned to tactical assets while their strategic counterparts are assigned to service wide areas. Nevertheless, the aspect of event-based cross-cueing is considered out of scope; instead, it is recommended to have an extra tactical system dedicated to this type of missions.

Beside these free-flying platforms, stationary systems—mainly camera-towers and aerostats, which provide persistent monitoring, are also considered along with the necessity of being anchored near the outpost due to the need for their protection and safety.

Thirdly, the target terrain should be evaluated from the perspective of smart insurgents in order to identify regions with high risks. Before doing so however, space has to first be modeled and associated with parameters of interest such as altitudes, degree of foliage, and proximity to population. Next, risk-factors ought to be identified, assessed, and incorporated in the model formulation. Such factors include spots around the outpost that have high altitudes and line-of-sight of the base; this situation creates a potential for the presence of long-range firearms. Also, trails and trafficable routes provide a fast way for armed vehicles to approach the outpost. Identifying such risks helps know which areas and routes should be monitored, and at which frequency (or *revisit-time*). This further generates the need for consideration of a forth aspect: *given a set of high-risk regions, what is the optimal flight path that visits all the desired spots, provided the technological limits of each platform?* This issue is rendered even more complex with the incorporation of all the three spatial dimensions. This is an important piece as the altitudes of the flight have to be known beforehand in order to achieve the appropriate image resolution.

Finally, assigning uniquely identified assets to areas and routes is necessary and needs special attention. This is important since the analysis of both the terrain and the technological systems only reveals what type of asset is useful on the theater; it does not inform us about the required number of each class. Thus, time has to be a factor in the equation in order to determine the number of identical systems that can be utilized concurrently. The resource allocation should then be time-based, and ought to further reflect the fact that the manpower qualified to operate the platforms and monitor the sensors' output is very limited in outposts. In addition, the inconvenience caused by scheduling

constraints has to also be considered; this includes the unavailability of non-organic assets due to refueling or maintenance.

In summary, the question of interest turns out to be a complex problem, and solving it by integrating all aspects into one model is extremely challenging, if not impossible. Subsequently, the procedure followed in this research effort is based on constructing an architectural framework that segregates the main problem into realistically solvable sub-problems. To be precise, the solution process is divided into three phases: terrain evaluation, tour generation, and resource allocation. Additionally, a fourth stage is added to the procedure in order to compute measures of interest for the surveillance architect as this helps assess the potential performance of the surveillance plan.

CHAPTER 2: BACKGROUND

1. Technologies

The military has always been interested in developing or purchasing technological systems that would increase its deployment and missions' success. A great part of these systems is focused around the protection of the soldier, and they range from special suites to communication architectures in theater, to surveillance assets. Of a particular interest to this study are surveillance systems, which can be categorized under different groups, depending on their attributes and utilization.

A surveillance system can be decomposed into different components: (1) the platform, (2) the mission package (which includes sensors), (3) the human element, (4) the control element, (5) the display, (6) the communication architecture, and (7) the associated cycle logistics. For more information on these aspects, see the report of the Army's Unmanned Aircraft Systems Center of Excellence (UAS-COE) (2010). In this study, focus is placed on the first and part of the second one, whereas the availability of the rest in theater is assumed.

As far as the platform is concerned, it can be (1) stationary, such as towers, or (2) mobile, such as fixed-wings and rotary-wings vehicles, or (3) both, such as lighter-than-air (LTA) platforms, which can be either tethered or free-flying. Below, each type is discussed in more details:

- Towers and fix-mounts, which can be inexpensive and easily deployed, can serve as the platform for a stare- (or persistent surveillance) system. However, it can only hold small sensors, which tend to have a shorter range for coverage. Another limitation is that it needs to be protected as it cannot flee from threats and does not carry weapons; therefore, it has to be located close to the base or outpost.

Figure 1: A surveillance tower
Source: http://www.carnetdevol.org/actualite-ballon/aerostat/usArmy.html

Figure 2: An LTA Vehicle (airship)
Source: http://www.militaryaerospace.com/articles/2011/05/tcom-to-provide-five.html

- LTA Platforms, as shown in Figure 2, are vehicles that use a light gas, such us helium, to stay aloft. LTAs take different sizes, from balloons to airships, with each size having a specific payload. In military, the typical name for stationary LTAs is *aerostats*, whereas its mobile counterpart is called *airships*. Although they are more expensive than towers, LTAs can yield less costly operations and maintenance (O&M) than winged vehicles, while still supporting heavy lifts. Their relatively larger payload enables the use of more sophisticated sensors at higher altitudes, thusly, covering large areas. Their drawback is vulnerability to firearms and long-range attacks as well as certain weather conditions. For more information about LTAs and their military use, see Bolkcom (2004).
- Winged platforms can be grouped under several overlapping categories. First, they can be either rotary-wings, such as helicopters, or fixed-wings, such as commercial airplanes. Second, they may be either manned or unmanned (UAV); and for the latter, they can be either remotely controlled or autonomous. Finally, they can be categorized, depending on their use and attributes, under the following classes (UAS-COE, 2010):
  - Class 1: This type of airborne devices is small-sized, highly portable, and easily launched by hand. They fly at very low altitudes, carry small sensors (e.g. EO, IR, SAR), and provide

accurate situational awareness. However, their endurance is very low (1-3 hours). A good example of this group is the RQ-11 Raven, which can be seen in Figure 3.

Figure 3: The RQ-11 Raven
Source: http://www.murdoconline.net/archives/date/2010/01/page/2

- Class 2: This sort of aerial vehicles is medium-sized, catapult-launched, and relatively easy to transport. They fly at low altitudes, carry small sensors, but enjoy a bit more power and endurance than their counterpart from class 1. A good example of this group is the ScanEagle, which is illustrated in Figure 4.

Figure 4: The ScanEagle
Source: http://www.milavia.net/news/archive.php?2006-13

- Class 3: This kind of airborne platforms is larger and heavier than the previous two, weighing between 25kg and 600kg. They fly at medium altitudes with high speed and do not usually require improved runways for launch. They can carry more sensors as well as small weapons

and other devices (e.g. communication relays and moving target indicators). However, they might sometimes experience a decrease in endurance (in comparison to class 2 vehicles) due to the payload increase. Also, their transportation usually needs logistical support. A good example of this group is the RQ-7 Shadow, which is shown in Figure 5.

Figure 5: The RQ-7 Shadow
Source: http://www.militaryfactory.com/aircraft/detail.asp?aircraft_id=326

Figure 6: The MQ-1 Predator
Source: http://www.vectorsite.net/twuav_13.html

- Class 4: This type of aerial devices enjoys more power and payload capacity than their aforementioned counterparts without compromising endurance. They are also able to fly at medium and high altitudes and carry a variety of assets including weapons and automated identification systems. They do however require improved runways for launch and recovery,

and they usually have a large logistical footprint. A good example of this group is the MQ-1 Predator, which can be seen in Figure 6.

- Class 5: These aircrafts are the largest available platforms. They fly at medium and high altitudes with the greatest airspeed, range, and endurance. They carry a wide variety of devices, ranging from wide-area sensors and radars to supplies and heavy weapons. They can cover very large areas, but their resolution and identification accuracy are usually low. Other drawbacks of their use include their significant logistical footprint as well as their requirement of improved areas of launch and recovery. A good example of this group is the Constant Hawk, which is shown in Figure 7.

Figure 7: The Constant Hawk
Source: http://defense-update.com/features/2009/sept/1709009_awapss.html

A sample of the platforms and their associated data, including the minimum and maximum operating altitudes (minOA & maxOA) can be found in Table 1 and Table 2:

Table 1: Platforms and their associated motion-based attributes

Platform	Communication Range (m)	MaxOA (ASL) (m)	MinOA (AGL) (m)	Cruise Velocity (m/s)	Downtime (s)	dB-Distance (AGL) (m)
Tower	n/a	20000	30	0	0	n/a
Aerostat	n/a	4570	457	0	0	n/a
Raven	10000	4250	9	18	300	152
ScanEagle	100000	5950	60	25	900	152
Shadow	50000	4500	152	40	3600	213
Grey Eagle	n/a	7620	1524	46	10800	609
Constant Hawk	n/a	6090	2438	90	10800	1524

Table 2: Platforms and their associated attributes that are not motion-based

Platform	Operators Count	Payload (kg)	Endurance (h)	Wind Resistance (m/s)	Alpha Angle (deg)
Tower	0	4.53	n/a	25	~0
Aerostat	1	90.71	n/a	20	90
Raven	1	0.18	1.5	8	45
ScanEagle	1	4.53	24	16	45
Shadow	2	45.35	7	11	45
Grey Eagle	3	362.87	30	20	45
Constant Hawk	4	680.38	7	20	45

From the other end of the spectrum, sensors represent the other aspect of aerial systems that is of special interest. These devices differ from each other through a set of capabilities and parameters, some of which can be found in Table 3:

Table 3: Sensors and their associated attributes

Sensor	Weight (kg)	Slant-Range to Detect Vehicle (from Ground Level) (m)
Raven IR	0.1814	640
micro-12	3.4019	10800
MX-12	18.1437	17600
MX-15	54.431	18400
AWAPSS-IR	317.5147	6000

Other sensors that can be employed include SAR, GMTI, SIGINT, and radars, all of which can be modeled as a set of parameters. Note that SIGINT (signals intelligent) platforms can also provide monitored radio and telephone communications.

2. Literature Review

In this section, some of the related topics that have been addressed in the literature are discussed. Such topics are grouped under four major categories: (1) the importance of understanding and modeling the terrain for good situational awareness, (2) the dynamic nature of interacting with an adaptive and smart adversary, (3) the issues pertaining to employing UAVs when planning and executing

missions, and (4) the different modeling techniques used by researchers to provide solutions for paths generation and resource allocation.

2.1. Space Representation and Assessment

Military targeting in general, and military surveillance in particular, are greatly dependent on the theater's geographical environment. Therefore, the spatial structure ought to be well modeled and analyzed. In fact, such a study becomes even more important when terrain is complex and mountainous as visibility and aircraft flight can greatly be affected. We define *complex terrain* as any geographical location that causes complexity or hindrance in terms of operations or field of view, such as sloppy regions and harsh weather (e.g. wind and dust). In this section, the methodologies used in the literature for space representation are listed, and some of the work performed on terrain assessment is mentioned.

As far as space representation is concerned, whether it is modeling sea, terrain, routes, or flight paths, there are four main techniques:

- The first one is based on a set of nodes or waypoints that are distributed across the target region. Although simplistic, this way of representing terrain is used by several studies such as Secrest (2001), Sheu et al. (2005), Kilby et al. (2007), and Balcik et al. (2008). In fact, in cases where the surface of the earth is flat (e.g. sea) or has a known infrastructure (e.g. roads), it makes perfect sense to represent space as a set of points of interest. For instance, Kilby et al. (2007) employs this strategy in structuring the problem in order to plan maritime surveillance for the Australian nautical areas.
- The second technique utilizes continuous representation of space, which requires the use of calculus and differential methods as well as curvatures and nonlinear measures. Studies that favor this strategy include Novy (2001), Zabarankin et al. (2002), and Ahmadzadeh et al. (2008).

The issue with this methodology is that it can be a challenge when modeling, and whose complexity increases even further when constraints are added (Royset et al., 2009). For instance, Ahmadzadeh et al. (2008) uses continuous representation, but soon reaches the point where the model has to be discretized before being solved.

- The third technique is based on the use of *Triangulated Irregular Networks* (TINs) (see Peucker et al., 1980), which is a type of data structures most commonly used in geographic information systems (GIS). For illustration, Figure 8 depicts an example of a region that is modeled through a TIN. The advantage of such technique is that along the capability of well representing 3D surfaces, it can yield "adaptive resolution, support hierarchical methods, [and] have generally lower storage requirements" (Fok, 2005, p. 2). Its disadvantages include its difficulty to implement as part of a solution, let alone the case where the appropriate (highly detailed) data is hard or impossible to obtain.

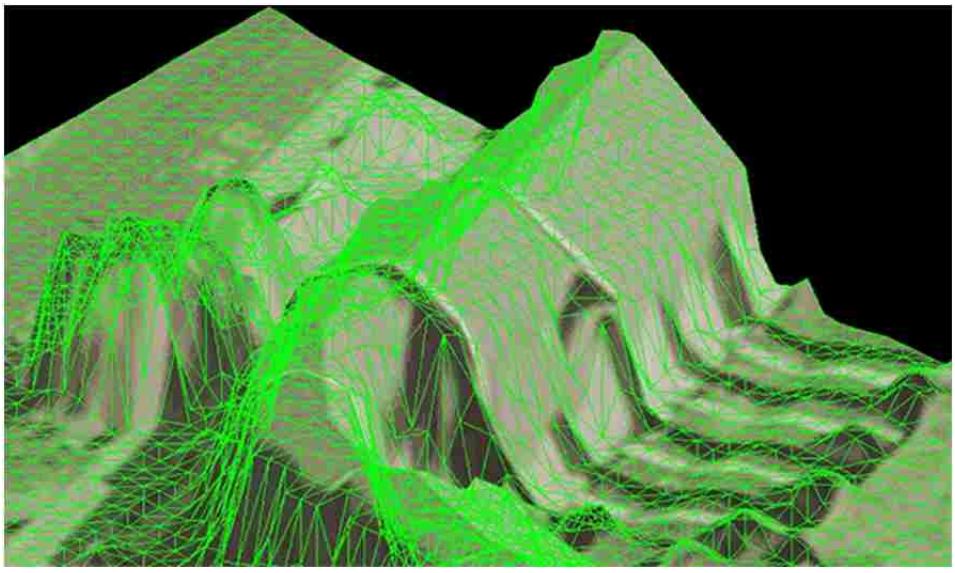

Figure 8: A region modeled through a TIN
Source: http://www.stanfords.co.uk/business-mapping/lidar/

- The last technique is based on a discrete representation of space; it is also called a grid, a grid-mesh, or *Regular Square Grid* (RSG), and seems to be the most popular as it is used by many

studies (Grignon et al., 2002; Kim & Hespanha, 2003; Fok, 2005; Royset et al., 2009) and implemented in coverage analysis commercial software such as AGI's STK[3] (see an application example in Carrico and Downer, n.d.). The grid digitization promotes simplicity and flexibility in modeling and developing algorithmic solvers, but can require high memory storage; this concern is however decreasingly considered as computational power continuously increases (Fok, 2005).

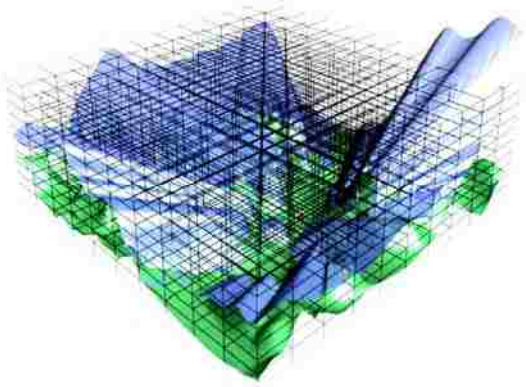

Figure 9: Dividing the 3D space into a voxel-grid
Source: Sikranth et al. (2008)

RSG also leads to the tradeoff between accuracy and computational complexity; as an example, for the average slope of a certain grid cell to be accurate, the mesh has to be granular enough, which might cause computational expensiveness. Last, a benefit of this technique is that once the grid is established, a network model can be generated (as performed by Royset (2009), by having every two adjacent cells represented by two nodes connected by an arc), which leads the end result to resemble the firstly-presented representation (a network of waypoints), but with a much greater difference in scale and structure.

[3] STK, or Systems Tool Kit, is a software package that allows the user to model, analyze, and visualize space, defense, and intelligence systems. STK is developed by AGI. Source: http://www.agi.com/

On the other hand, when talking about analyzing space and evaluating terrain, a number of assessment techniques can be mentioned. One reason for the existence of several methods, whether developed for military, government, or civil purposes, is that the aim and the type of the study as well as its associated factors need customized analyses. Waser et al. (2011) for instance uses raw data obtained through visual sensors to classify tree species. Caplan and Kennedy (2010) utilizes a multi-layered matrix framework on crime data to predict future crimes in city neighborhoods in order to assist the police force allocate its limited resources more efficiently. Sikranth et al. (2010) employs 3D rendering of simulated terrain, along with a haptic interface to capture environmental constraints, in order to find aircraft paths that provide optimal visibility. Howard and Seraji (2002) uses real-time terrain assessment combined with fuzzy navigation algorithms to help autonomous mobile robots choose the safest and most trafficable path on the ground. Last, Fok (2005) discusses the design of a performance assessment architecture that uses a multi-phased analytical approach to evaluate how well a planned path performs in a complex terrain. His model receives ground coordinates and flight tours as inputs and presents visibility attributes and coverage quality as outputs, along with some other relevant statistics. In summary, the methodology followed to evaluate planetary surfaces and their surroundings depend on the needs and attributes involved in the study.

2.2. Counter-Insurgency

Terrain assessment becomes of a great value when the planetary surface is complex, especially when intelligent insurgents take advantage of this complexity. Places with high slopes and elevations might present a more attractive option for cover. Staying in villages and merging with civilians may pose a good opportunity to temporarily stay safe and acquire supplies. Furthermore, as a counter-measure against aerial surveillance, insurgents might choose to settle in regions with extensive foliage or where wind prevents flight of tactical UAVs. The intelligent adversary may then be able to use the

environmental aspects of the geographical locations that are familiar to them to hide and launch attacks. It is therefore important to obtain enough intelligence about the surrounding geo-locations.

On the other hand, situational awareness is acquired through *Intelligence, Surveillance, and Reconnaissance* (ISR) means, which are critical to mission planning and execution (Downs, 2007). Downs also adds that the missions' probability of success can be enhanced by augmenting the availability of ISR assets and using them efficiently. Indeed, *improvised explosive devices* (IEDs) can potentially be located, and infiltration routes and ambush locations can be identified and monitored. However, the study asserts that the current net effect of the ISR missions is far less than optimal despite the high ISR hours flown, and that is due to the lack of appropriateness of positioning airborne elements for *counter-insurgency* (COIN) operations. This is further complicated by the fact that freedom missions (such as in Iraq and Afghanistan) are meant to protect rather than attack, while the adversary is small, hidden, dynamic, and unpredictable. In fact, Turbiville (2009) studies and compiles a list of cases and approaches adopted by guerillas, insurgents, and terrorists from several parts of the world. The author demonstrates the importance of intelligence in these groups' operations, a fact that emphasizes that the adversary is indeed *smart*, and that nonconventional military methods should be explored. Using this mindset, Kaplan et al. (2010) considers counter-insurgency against smart enemy from a game theorist and force allocation standpoints. The study develops an optimization decision tool that takes into consideration different levels of government intelligence as well as casualties caused by attacks on terrorist cells. Nevertheless, this work is different than ours as the authors focus on engagement rather than surveillance, while they do not consider terrain complexity.

When exploiting airborne surveillance for COIN, four main tasks are usually performed against the enemy, as specified by Chandler et al. (2002): (1) search (or surveillance), (2) classification (or identification), (3) attack (or engagement), and (4) verification. Surveillance targets monitoring areas of interests and finding suspicious behavior or a change of local activities. Once a detection of an unusual

object occurs, the need for more intelligence is necessary; as a result, a reconnaissance mission is launched in order to identify the detected entity. If it is classified as a threat, then necessary actions are taken. Typically, these actions have consisted of engaging in attacks to eliminate the menace; however, counter-actions can also include fortifying the defense and waiting for the adversary to arrive, or calling for a rapid aerial backup. Once the attacker is destroyed, it might be required to order a verification mission so as to insure that the threat is not going to be persistent in the future. That been explained, it is worth noting that this study is centered around the first two tasks as its goal is to support a defensive model rather than an offensive one, especially that OPs and COPs have a temporary presence and limited human resources.

2.3. UAV Employment and Mission Planning

With the recent robotic evolution in the last decades, the military has developed a special rising interest towards UAVs (Mettler, 2003; Bayraktar, 2005). These airborne devices have several advantages over their manned counterparts such as being quiet, less expensive, easier to fly in low altitudes and around corners, and probably and most importantly, capable of reducing risk and workload on soldiers (UAS-COA, 2010). As a result, recent research and development efforts have aimed to render UAVs more employable in both theaters and civilian settlements through addressing constraints and control issues, advancing mission planning and aviation techniques, and improving computational performance.

Research has strived to ameliorate the UAV motion manipulation and the generation of flight paths. As a matter of fact, aerial vehicles' maneuverability limits have proved to be a challenge in real-time avionic control (Ahmadzadeh, 2008), a genuine motive behind the several explorations in autonomy and flight-routing during this past decade. Frazzoli et al. (1999) for example presents a framework that yields efficient real-time trajectory generation for an autonomous helicopter; the developed hierarchical architecture yields considerable reduction in computational requirements due to

the simplified system design. Choset (2001) explores the coverage path planning approach for mobile robots by surveying research studies as well as comparing their contributed algorithms and categorizing them into four major types; the author concludes that there are advantages and disadvantage to each kind in terms of coverage optimality, obstacle detection, computational complexity, and others. Mettler et al. (2003) develops test scenarios to expose and evaluate the capabilities and limitations of the constructed instantaneous route planner and the maneuverability of the subject autonomous UAV. Bayraktar et al. (2004) conducts UAV flight experiments both at the University of Pennsylvania testbed and using simulation in order to validate a real-time algorithmic model that accounts for both physical dynamics and mode-switching logic; the study reports positive results and concludes that the founded framework can serve as modeling abstraction for future UAV cooperative control.

Other researchers target path generation less at the control and aviation level, and more at the pre-flight decision making level; their studies are mainly concerned with mission planning and optimized vehicle routing, either specifically for UAVs, or generally for any free-flying platform. For instance, Srikanth et al. (2008) introduces a built simulation-based optimization system that exploits a haptic interface to refine the flight path in a triple-dimensional space; the mechanism allows the assessment of both the vision and visibility of the airborne asset in order to promote informed decisions during organizing a mission. On the other hand, Seacrest (2001) aims to reduce the complexity of the mission route planning (MRP) problem and its solution runtime though a heuristic approach; the author's focus revolves around fine-tuning and experimenting with different algorithm configurations to achieve near-optimality in the shortest possible time. Kim and Hespanha (2003) introduces a procedure to efficiently generate shortest paths in effort to minimize the UAV's risk and time over defended enemy grounds; the approach reduces the problem at hand by shifting from continuous representation to a discrete one, which results in better computation times without compromising the quality of the solution. Similarly, Royset et al. (2009) explores a constraint-based optimization model for shortest flight trajectories that

accounts for land threats such as surface-to-air missiles (SAMs) stations; it also provides great flexibility of analysis with regards to important tactical aspects like terrain avoidance, terrain-masking of enemy radar, varying aircraft speeds, as well as several kinds of aircrafts. The authors apply the presented tool to different synthetic military missions, one of which is specific to the unmanned Northrop Grumman Hunter (MQ-5B).

To help provide flight solutions that are more practical, some researchers opt to address constraints and limitations that are faced in the physical world. For example, a common potential hindrance in UAV flights is the impact of wind on airborne operations, an issue taken into consideration by several studies such as Mettler et al. (2003), Bayraktar et al. (2005), and Ahmadzadeh et al. (2008). Further, Ahmadzadeh et al. (2008) incorporates in the developed model the vehicles' maneuverability limits as well as the fixed positions of the camera sensors. From another perspective, Kilby et al. (2007) and Royset et al. (2009) model two predominant restraining factors: mission deadlines and fuel capacities. The latter study even considers more flight constraints by integrating the aircrafts' operating altitudes as well as the diving and climbing rates in the constructed *Mixed Integer Program* (MIP) model. Another practical study in this context is Srikanth et al. (2008), which applies kinematic and environmental constraints through forces exerted on a haptic interface, and allows the user to specify restrictions on curvatures and total length of curves.

A common issue observed through the reviewed work is dealing with the tradeoff between flexibility and optimality during the flight mission planning process. To explain, there are usually two kinds of utilized methods: online and offline. Online (or real-time) programs are executed during a vehicle's flight and are usually used as part of the control operation of robots, autopilots, or autonomous UAVs (Frazzoli et al., 1999; Howard & Seraji, 2002; Mettler, 2003; Bayraktar, 2005; Kilby et al., 2007); but they can also be utilized during a last-minute planning of an ongoing mission (Fok, 2005). To illustrate the utility of online algorithms, consider Kilby et al. (2007), a study that considers real-time

path planning computation to monitor the Australian nautical areas; the involved airborne vehicle has a pre-defined set of waypoints to visit, but without previous knowledge about the location of future ship encounter. If a (moving) maritime vessel is detected, the aircraft has to find the best way to get close enough to it, classify it, and then continue the surveillance mission. The corresponding online program is executed dynamically once every one minute to update the flight tour. While such a method provides great flexibility that yields a way to adjust for last-minute occurrences and unpredictable events, it falls short in solving large-scale problems or carrying out computationally expensive processes since there is a very tight limit on the solution runtime or processing power. To mitigate this limitation, the computation associated with flight planning and mission targeting can alternatively be performed offline (i.e. prior to the trip), but the flexibility aspect would be lost. On the other hand, offline techniques are ideal for some pre-flight mission planning tasks such as resource allocation, and even route generation, given a certain degree of certitude. For instance, Chandler et al. (2002) helps make cooperative decisions about vehicle assignments to teams during the offline mission planning process, whereas Royset et al. (2009) targets pre-flight route generation provided that the locations of ground threats are known beforehand. Note that there are also doubly-phased models that use both kinds of algorithms—online and offline—in an interoperable fashion. A good example is the analytical tool presented by Fok (2005), which first performs offline computations to analyze the terrain and generate horizon surfaces; then the online evaluation is triggered upon request during the assessment phase. Such paradigm helps the planning operations during the second stage be executed much faster.

Note that unlike the studies that assume the presence of artificial intelligence and cooperative control, the models developed in this research are kept general, and the availability of autonomous UAVs is not particularly assumed. This is an important point as this effort serves the near-term decision making on resource allocation, a task that requires careful consideration of manning requirements, especially with respect to the scarce qualified personnel that are able to control the UAVs.

2.4. Modeling and Algorithms

To model research problems, the relevant studies use diverse abstract structures and objective functions depending on the addressed issues, which span cooperative control, threat avoidance, surveillance coverage, resource allocation, and others. For example, Carrico and Downer (n.d.) targets a flight plan that maximizes coverage and minimizes revisit-times for a commercial aircraft utilized for the surveillance of coastal waters; the modeling aspect is handled by the employed software package—STK. Ahmadzadeh et al. (2008) pursues the same coverage objective for a set of cooperative unmanned aerial vehicles, but the approach is based on a combination of calculus and *Integer Programming* (IP) , whose model is dependent on the discretization of time, space, and curvature. Secrest (2001) on the other hand solves the MRP problem as a *Traveling Salesman Problem* (TSP) with an objective of minimizing the distance traveled, since shorter tours lead to reduced travel time, fuel consumption, and potential risk. Conversely, Novy (2001) considers generating UAV paths that have minimal exposure to the enemy's radar by relying on calculus of variation and sensitivity analysis. In a similar effort, Jun and D'Andrea (2003) assumes a *Shortest Path Problem* (SPP) structure and uses the *Bellman-Ford algorithm* to plan safe UAV routes on a digitized 2D terrain. Further, Royset et al. (2009) uses a constrained-SPP structure modeled as a *Mathematical Programming* (MP) formulation to find the path that gives the highest probability of mission success given that there are ground threats on the theater; the mission consists of having the aerial vehicle reach the destination where the identification or engagement mission is to be accomplished, and then return to the base unharmed. As for resource allocation approaches in military context, consider Chandler et al. (2002), a study that uses a hybrid method that is based on graph theory and a variant of the *Linear Assignment Problem* (LAP) to allocate teams and UAVs to military tasks; the objective is to complete the mission in the least amount of time. Alternatively, Kaplan et al. (2010) applies a *Knapsack* approximation to allot forces to the *yet-to-be-decided* attack plans with a min-objective function that reflects the damage inflicted on civilians and allied soldiers. Dong et al. (2010)

adopts a *Maximal Covering Problem* (MCP) structure in an effort to allocate protective resources to a set of threatened intermediary facilities in a military distribution network, with the aim to achieve the least instability after the attack; the objective function is modeled as a sum of demand points. Gomez et al. (2008) constructs an ontology-based architecture that exploits a *Set Covering Problem* (SCP) model to assign sensors and platforms to the appropriate tasks for ISR missions. All in all, a significant portion of the reviewed mission planning efforts uses optimization in order to achieve effective solutions or efficient performance through different modeling schemes. Such research can also be categorized in a general manner under two groups: (1) routing or path planning problems, whether it is TSP-based or SPP-based, and (2) resource allocation problems. Yet, it is possible to find work in *Operations Research* (OR) that stems from both categories. Specifically, we highlight Balcik et al. (2008), a study that employs a multi-phased mathematical model that incorporates routing and resource allocation, along with scheduling, to come up with a complete plan during the *Last Mile Distribution* of disaster relief efforts. It is true that their methodology assumes the existence of a transportation infrastructure and does not account for terrain attributes; however, their model does inspire us in designing the aforementioned architectural framework as well as considering routing and resource allocation in the same solution.

A relevant remark about modeling that is worth mentioning is related to the way flight paths are represented. To elaborate, it seems that fewer research studies address the challenging 3D aspect of route planning. Amongst the reviewed work, the following are highlighted: Fok (2005) uses altitudes and horizon lines to generate a triple-dimensional grid-mesh of the terrain. Sikranth et al. (2008) employs 3D-rendering of the planetary surface in their software visualization before the generation of vision and visibility field of the flown vehicle. Carrico and Downer (n.d.) utilizes the STK commercial simulation software to build a 3D CAD model so as to evaluate area coverage and revisit-times by a Boeing 737 that is equipped with the radar AN/APY-10. Last but not least, Royset et al. (2009) divides the space into a voxel (cubical) grid before converting into a network of flight edges; this eventually leads the problem

and storage requirement to grow exponentially, but the use of a network reduction technique helps mitigate this effect.

Note that studies pertaining to route generation and mission planning can generally be segregated into two collections in terms of the underlying algorithmic models of their approach. There are ones that assume a well-established system infrastructure such as a well-known heuristic or mathematical program, and whose authors gravitate towards improving the performance and efficiency of the mechanism. The second group on the other hand tends to focus more on the modeling aspect by constructing a novel problem structure and assessing its effectiveness at supplying practical solutions. The research effort of this study may be categorized under the second collection.

2.5. Conclusion and Contribution

The literature review shows that surveillance planning and counter-insurgency have indeed received some attention; nonetheless, the reviewed work is not directly applicable to the target problem of this study. For terrain assessment, all the so-far found methods and analyses are specific to their subject problem and have therefore limited functionality. In addition, there is a lack of integrated optimization models that take the terrain analysis as an input and produce resource allocation and routing solutions. It is also noticed that research in the optimization theory literature in general, and in military surveillance in particular, enjoys a wide variety of methodologies for solving problems. However, there is still a need for modeling some of the realistic aspects that the surveillance architect looks for during the planning phase. Some of these aspects happen to be a great challenge to model or integrate in an already complex framework.

The contribution of this study is threefold. First, there is a lack in the literature of work that solves the presented problem from beginning to end; meaning that terrain evaluation, routing, resource allocation, and scheduling are important aspects that rarely occur and get resolved within the same

solution process. This effort on the other hand offers a comprehensive analytical tool that takes all the aforementioned points into consideration. Moreover, the constructed solution further associates smart insurgents operations with planetary surface complexity to assess terrain. Last but not least, this study introduces optimization models that incorporate newly defined constraints to achieve more realism and practicality. From a military standpoint, the surveillance architect is empowered with a powerful software solution that is also user-friendly, which brings about the efficient generation of surveillance plans and portfolios. Ultimately, the software user enters three sets of data that are relatively available: terrain data, sensors capabilities, and platform attributes. Then, the analytical engine generates the list of sensors and platforms to be purchased as an output, along with the routes and areas that each system is to service, the flight schedules of airborne vehicles, and relevant statistics that reflect the quality of the achieved plan.

CHAPTER 3: METHODOLOGY

1. Concept

In this context, a *surveillance portfolio* represents a set of technological systems assigned to different geographical points of interest to provide adequate protection to a specific region. Moreover, a *system* here refers to a combination of sensors (such as radars and cameras) carried or mounted on a certain platform (an aerial vehicle, an aerostat, or a tower). Further, we define the *surveillance plan* to be a layout (for static assets) and a set of scheduled tours (for mobile assets) that are covered by employing the resources of a specific portfolio. This concept of plans and portfolios is used to provide the surveillance architect with a way to allocate technological assets to multiple geographical regions and assess the quality of such allocation. A portfolio contains a number of three-way assignments, which include a list of sensors, one platform, and a set of regions that are to be serviced by these surveillance systems; this is illustrated in Figure 10.

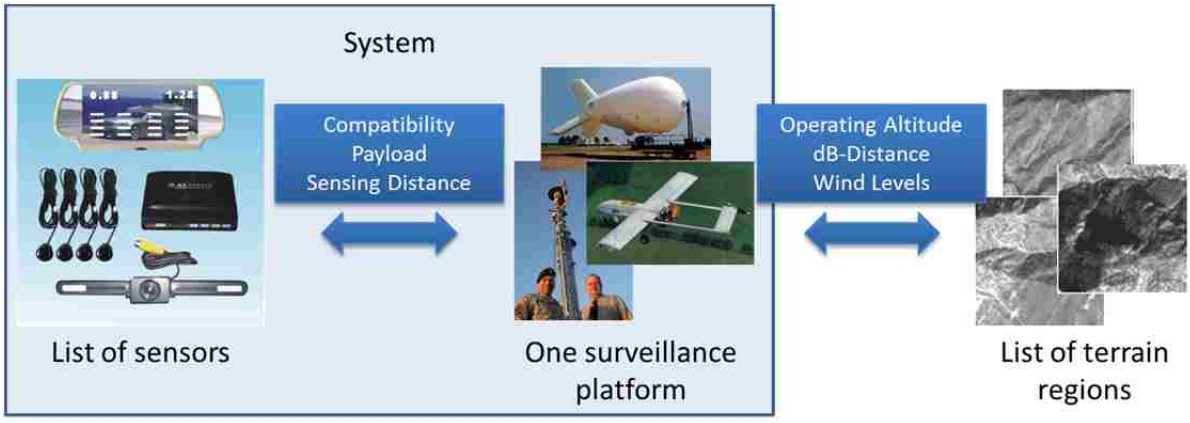

Figure 10: The three-way assignment model in a surveillance portfolio scheme

To model these regions in particular, and the spatial aspect in general, the terrain map is digitized into a two-dimensional grid, where the meaning of a grid cell is expanded from the notion of a small squared portion of the big picture to an abstract structural object. To elaborate, each cell shall act as an origin and destination node (for mobile assets), which provides a foundation for the construction

of edges and networks. A cell also serves as a container for data values pertaining to the characteristics of its geo-location, which span the following attributes, as well as the yet-to-be introduced *risk-factor* scores and their associated revisit-times:

- e_j is the average altitude (or elevation) of cell j's geo-surface
- e^m_j is the minimum altitude (or elevation) of cell j's geo-surface
- e^M_j is the maximum altitude (or elevation) of cell j's geo-surface
- β_j is the average slope angle of cell j's ridges and elevations
- p_j is the size of the population inhabiting the area of cell j
- t_j is the average trafficability level of cell j
- f_j is the average foliage level of cell j
- v_j is the average wind speed on the area of cell j
- o_j is 1 is cell j belongs to the target outpost, and 0 otherwise
- l^a_j is 1 if cell j constitutes a good location for an aerostat anchor-point, and 0 otherwise
- l^t_j is 1 if cell j constitutes a good location for a camera-tower installation, and 0 otherwise

Note that for the sake of simplicity, it is assumed that all these terrain attributes hold the same value across every point within the area of their comprising cell. Further, it is possible to adopt any cell size when adopting the presented methodology; however, the smaller this size is, the more accurate the procedure and the outcome are.

When constructing a surveillance portfolio, the challenge of establishing assignments of assets to geo-spots or terrain cells is the presence of a number of constraints that prevents systems from operating in specific regions. One advantage of using a constraint analysis framework in such a process is that the infeasible options are automatically omitted. This feature is even more valuable in cases

where most or all of the assignments are impossible as the task of building and checking all possible combinations is rendered unnecessary.

The constraints that are taken in consideration in this analysis are the following:

1. The **compatibility constraint**, which is verified when the selected sensor is installable on and compatible with the selected platform.
2. The **payload constraint**, which is violated if the sum of the weights of the selected sensors exceed the payload capacity of the selected platform.
3. The **sensing distance constraint**, which is verified when the system can be at an altitude from which it can sense at least a vehicle on the ground, with some tolerance on the altitude threshold (a 10% tolerance is used as a default value).
4. The **operating altitude constraint**, which is verified if the platform flies at a height that is between its minimum and maximum possible operating altitudes (specified by the manufacturer).
5. The **terrain elevation constraint**, which is verified if the platform can be at an altitude that is higher than the average altitude of the terrain, with some tolerance on the altitude threshold (a 10% tolerance is used as a default value).
6. The **stealth or noise (dB) distance constraint**, which is verified if the platform can be at an altitude from which targets on the ground cannot hear its noise.
7. The **wind level constraint**, which is violated if a system is assigned to a region where the wind level is higher than the platform's resistance.
8. The **control range constraint**, which is violated if the airborne vehicle is scheduled for a tour that leads it to leave its control range radius.

9. The **diving rate constraint**, which is verified if the difference in the operating altitude of a flying system from one cell to another adjacent cell does not exceed the maximum platform's diving rate. Note that diving does have a different consumption rate of battery.

10. The **climbing rate constraint**, which is verified if the difference in the operating altitude of a flying system from one cell to another adjacent cell does not exceed the maximum platform's climbing rate. Note that climbing does have a different consumption rate of battery.

11. The **endurance constraint**, which is violated if the route or area to be covered by the assigned system requires a time frame that is larger than the period allowed by the battery or fuel limit of the platform.

12. The **refueling need constraint**, which is violated if a platform is not provided with enough time to refuel or charge its battery before its next flight.

13. The **revisit-time constraint**, which is verified if the revisit-frequency required to regularly monitor a certain area or route is met.

14. The **asset protection constraint**, which is verified if the stationary platforms are within the range of the base's protection.

15. The **minimum resolution constraint**, which is violated if the minimum required image quality at a certain region or sub-route is not met.

16. The **scheduling constraint**, which is violated if a platform is flown during its non-designated time periods or during its unavailability due to maintenance.

2. Analytical Framework

This study aims at developing an analytical framework that can help the surveillance architect or the decision maker answer the investment question stated in Chapter 1. To achieve this goal, the

concept discussed in the previous section, along with the set of defined constraints, is leveraged to construct a structure for the developed tool in order to solve the problem of interest in a feasible and efficient fashion. In this section, a high-level architecture of the framework is introduced, and the analytical processes associated with each of its sub-levels are summarized.

Firstly, the high-level architecture of the proposed analytical framework is divided into four components that perform the computations of four sequential phases: (1) terrain risk assessment, (2) tour generation, (3) system allocation, and (4) surveillance plan evaluation (as depicted in Figure 11). The framework receives data about platforms, sensors, and terrain as input. In return, it presents as an output a solution to the research question, along with some performance measures about the proposed surveillance plan. This output also includes data that shall serve as a foundation for visualizing results through software tools such as Microsoft-Excel or custom programs.

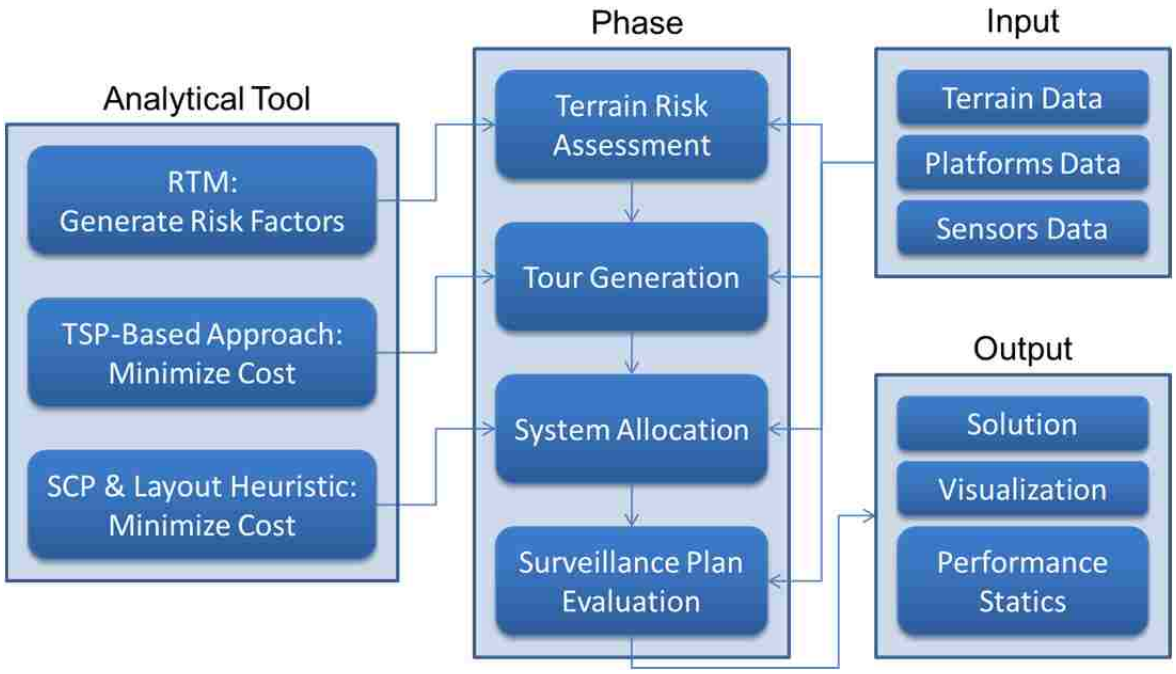

Figure 11: The high-level architecture of the analytical framework

The first phase of the analysis is concerned with assessing the risk of each grid cell in the aforementioned digitized terrain. This is achieved by employing a technique called *Risk Terrain Modeling*

(RTM) on the terrain data that is extracted from the framework's input. This evaluation method enables the approximation of the importance of each cell as well as the need and frequency to monitor it. Cells that are found to be of high potential importance to insurgents are referred to as *hotspots*. The criteria that this study is considering to identify these geo-spots span:

- Grounds that are of higher altitudes than the outpost
- High degrees of foliage
- Proximity to the villages and their population size
- Areas with low slopes
- Terrain with a high level of trafficability
- Places from which line-of-sight of the outpost is not obstructed
- Locations of interests from the eye of an expert in the field

The second phase receives the collection of identified hotspots as input and generates for each tactical system a tour that potentially visits every one of these cells in an efficient manner. This is accomplished by solving the Traveling Salesman Problem, whose output is a near-optimal sequence of hotspot visits. Next, the TSP tour is refined through a set of optimization models and algorithmic methods so as to acquire the complete cell-to-cell routes on the digitization grid. This process includes finding the appropriate operating altitude at which the UAV should fly so its sensor can capture good-resolution images. This is particularly important because the purpose of sending out surveillance assets is to obtain clear information and intelligence that allow the outpost personnel to acquire accurate situational awareness. Some of the flight specifications are determined using the framework's input information, spanning terrain, sensors, and platforms data. The flight restraints that are considered in this tour generation phase are related to the following constraints: the compatibility between the platform and its assigned sensor, the payload capacity of the platform, the sensing distance, the operating altitude, the dB-distance, the wind level, the diving and climbing rates, and the platform's

endurance. Planning routes without considering these not only jeopardizes the success of the mission, but might lead to the loss of the surveillance asset as well. Note that the refined tours are not guaranteed to be entirely traversable due to the endurance limit of the aerial vehicle; if so, the trajectory is segmented into feasibly crossable subtours, only the most efficient of which are selected in the next stage.

During the third phase, surveillance systems are to be assigned to service given routes and areas. Accordingly, optimization models and heuristics are developed with a built-in consideration of allocating the proper assets to the right regions and trails. In other words, the analytical framework takes into account the fact that aerial vehicles can appropriately monitor high-risk routes and regions far away from the outposts, whereas the stare-systems are suitable for the persistent surveillance of the nearby surroundings of the outpost. By exploiting the terrain and systems data along with the tours and subtours generated in the previous phase, the resource allocation process for mobile assets is performed in a time-based manner in order to realistically scale the flights schedule for the whole 24-hour period. This is performed using a variant of the *Set Covering Problem* that takes the apt constraints into consideration and minimizes the number of required resources to purchase. As for the stationary assets, a layout heuristic is executed to efficiently position sensors so as to decrease unnecessary redundancy in FOV coverage and provide a lean solution. The models and algorithms are also concerned with satisfying the constraints associated with the previously introduced factors: the need to refuel the free-flying vehicles, the appropriate revisit-frequency, and the necessity of asset protection. Moreover, scheduling restrictions are considered by accounting for aspects that cause predictable temporal inconvenience during flight operations—mainly downtime during refueling and maintenance. In summary, time-based and space-based optimization is applied in this stage of the analysis to provide a realistic resource allocation solution with the minimum required assets to be deployed, and that completes the construction process of the surveillance plan.

Finally, the last phase uses a set of identified performance measures to reflect the feasibility of the attained portfolio of systems and the robustness of its associated surveillance plan. This is discussed with more details in the next section of this chapter.

3. Performance Measures

The surveillance plan is as important as the lives it protects; therefore, it is compulsory to have a way to evaluate it and measure its quality. This becomes even more essential when the solution obtained through optimization does not tell the whole story. In other words, after producing the best possible plan, it is critical to know where the weaknesses are. These include the regions and spots that: (1) cannot be reached, (2) can be reached less frequently than necessary, or (3) can be reached in a timely manner but with low image resolution. It is also necessary to know how much slope coverage can be achieved as a UAV may pass right by an insurgent situated on a high slope and not capture him visually. Although the developed optimization models cannot address this issue, this can serve as a warning to the operators so that they tilt the UAV's sensor with an angle that is large enough, or even make several rounds around a hill or a mountain to increase the chances of detection success. In such cases, close attention has to be paid to fuel/battery limits—and one way to mitigate this problem is to plan flights with energy reserves. Another important point is the rate of utilization of flown assets because maintenance is as important as operation. These explained situations introduce the need to extract useful statistics for the surveillance architect after the acquisition of the optimized surveillance plan.

After a plan and its associated portfolio are constructed, the surveillance architect can analyze its performance by evaluating four identified measures:

1. The **required manning**, which is the amount of human resources necessary to operate the systems that are included in the portfolio. This measure is computed by summing up the number of needed personnel to control, manage, and maintain both platforms and sensors.

2. The **surface coverage**, which is the overall area around the outpost that is constantly monitored through the set of installed stationary systems. Such a measure is computed using a heuristic that minimizes the overlap between the intersecting sensors' FOV. The computation process is described in details in Chapter 6 Section 2.

3. The **overall image quality**, which is modeled by the average sensor resolution provided by all the surveillance systems that are included in the portfolio:

$$R = \text{Average}_{\{i \text{ in Sensors, } j \text{ in Regions}\}} (R_{ij}) = \text{Average}((1000\, s_i/r_i) \,.\, \cos(\theta_{ij}) \,.\, (1 - f_j/3))$$

Where: - R_{ij} is the image resolution of sensor i in region j

- s_i is the sensitivity (power) of sensor i

- r_i is the slant-range of sensor i

- θ_{ij} is the angle between the slant direction of sensor i and the normal line on the surface of cell j, as shown in Figure 12

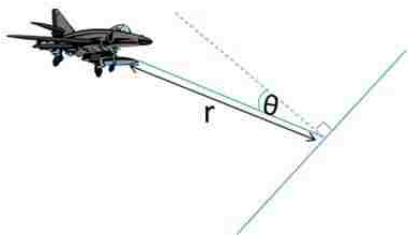

Figure 12: A geometric representation of the angle θ

- f_j is the foliage degree of cell j (note: f_j an integer score that ranges from 0 to 3, with 3 being *dense foliage*)

4. The **average slope coverage**, which is the average amount of slope (across all traversed cells) that each system can potentially cover over a mountainous region. The *slope coverage* (SC) of a cell is modeled by a percentage that results from dividing the *covered range* of a mountain (i.e. the sloping distance that can be feasibly scanned by the specified aerial vehicle) by the *total*

(sloping) range. Such a measure can be extremely hard to obtain due to the random shape of elevated grounds. Nonetheless, a good approximation can be acquired in a much simpler way, and that is based on supposing that the terrain of every cell is a flat surface with a simple isosceles-triangular mountain on top of it, as depicted in Figure 13.

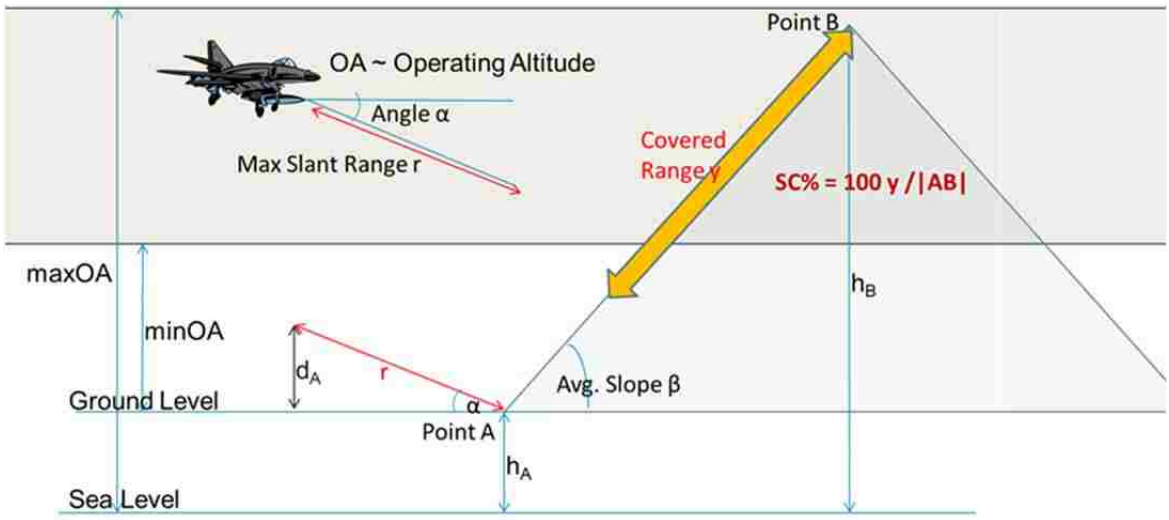

Figure 13: A geometric representation of the slope coverage model for a cell

To ease the construction process of the SC formula, consider the following parameters:

- minOA is the minimum operating altitude (above ground level) of the flying system
- maxOA is the maximum operating altitude (above sea level) of the flying system
- r is the slant-range of the most powerful sensor mounted on the surveillance system
- α is the tilt angle of the most powerful sensor mounted on the surveillance system
- β is the (average) slope angle
- A is the lowest point of the cell's mountain
- B is the highest point of the cell's mountain
- h_A is point A's elevation (above ground level)
- h_B is point B's elevation (above ground level)
- h_{AB} is the elevation between point A and point B, i.e. $h_{AB} = |h_A - h_B|$

- |AB| is the (land) segment between points A and B, which is equivalent to the aforementioned *total range*; this also means that: $|AB| = h_{AB} / \sin(\beta)$
- d_A is the distance from which the flying system can spot point A, hence: $d_A = r \cdot \sin(\alpha)$
- y is the aforementioned *covered range* (depicted by the orange arrow in Figure 13)
- z is the vertical distance between point B and the vehicle at minOA, i.e. $z = |\text{minOA} - h_{AB}|$

From its definition, the slope coverage is formulated as follows:

$$SC = y / |AB| = y \cdot \sin(\beta) / h_{AB}$$

However, the variable y is computed in different ways depending on the circumstances that the surveillance system confronts given the technological limitations r, minOA, and maxOA. Subsequently, four practical cases are identified, all of which assume that the aerial vehicle can feasibly fly over the target cell (i.e. $\text{maxOA} > h_B$):

- Case 1: The aerial vehicle may have to fly so high that point B is missed by the sensors; this happens when $d_A + h_{AB} < \text{minOA}$. In such a case, there is no coverage and y = 0, as demonstrated in Figure 14.
- Case 2: Ideally, the flying system has a sufficiently large slant-range to capture the whole mountain; this occurs when $h_{AB} \leq d_A$ (for any minOA that is smaller than $\text{maxOA} - h_A$). In such a scenario, full coverage is attained and y = |AB|, as depicted in Figure 15.
- Case 3: The airborne platform may cover the top of the mountain when it flies high enough, but it is possible that the slant-range is not long enough to spot point A. This happens when $h_{AB} \leq \text{minOA}$ and $d_A \leq \text{minOA}$, which leads to $y = (d_A + h_{AB} - \text{minOA})/\sin(\beta)$, as shown in Figure 16. Note that this case encompasses Case 1 as $d_A + h_{AB} < \text{minOA}$ implies $h_{AB} \leq \text{minOA}$ and $d_A \leq \text{minOA}$; this means that the formula $y = \max(d_A + h_{AB} - \text{minOA}, 0)/\sin(\beta)$ can address both Case 1 and Case 3.

- Case 4: In the situation where the aerial asset can indeed fly low enough to scan the bottom of the mountain (i.e. minOA < h_{AB}), spotting point A would depend on the length of the slant-range r, and hence the magnitude of d_A. In such a scenario, Figure 17 demonstrates that $y = \max[(d_A + h_{AB} - \text{minOA})/\sin(\beta), |AB|]$. Notice there is full coverage if minOA $\leq$ d_A.

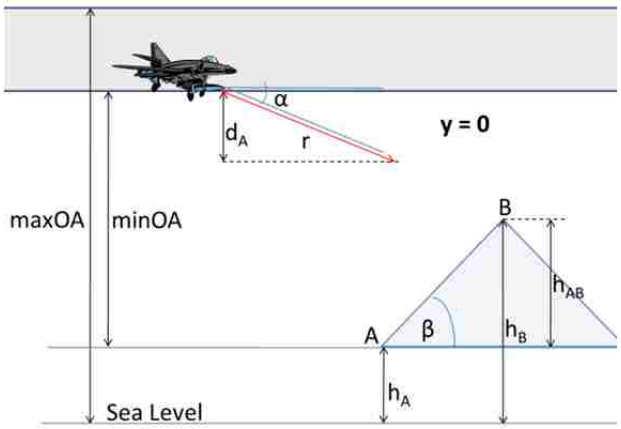

Figure 14: The geometric situation for Case 1 (i.e. $d_A + h_{AB} < \text{minOA}$)

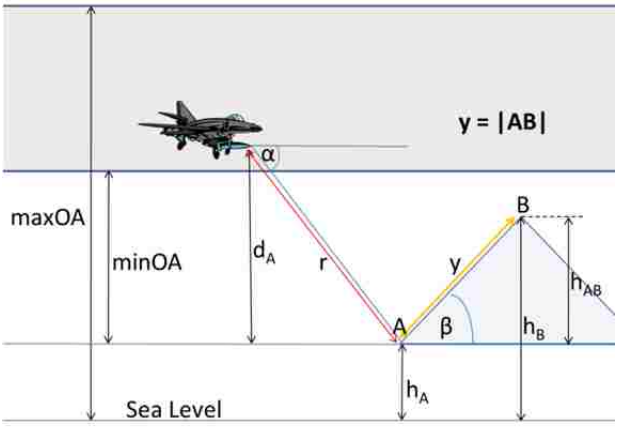

Figure 15: The geometric situation for Case 2 (i.e. $h_{AB} \leq d_A$)

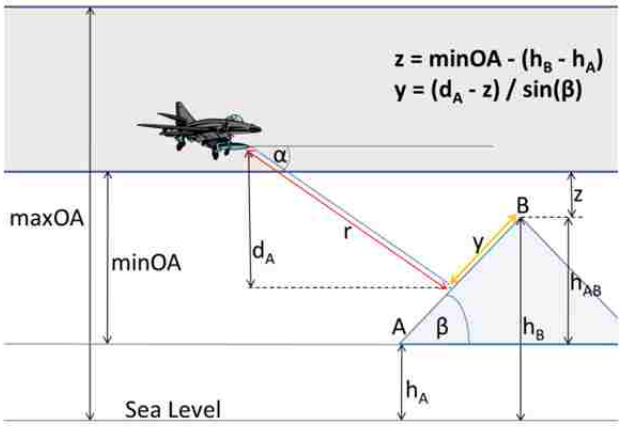

Figure 16: The geometric situation for Case 3 (i.e. h_{AB} < minOA and d_A < minOA)

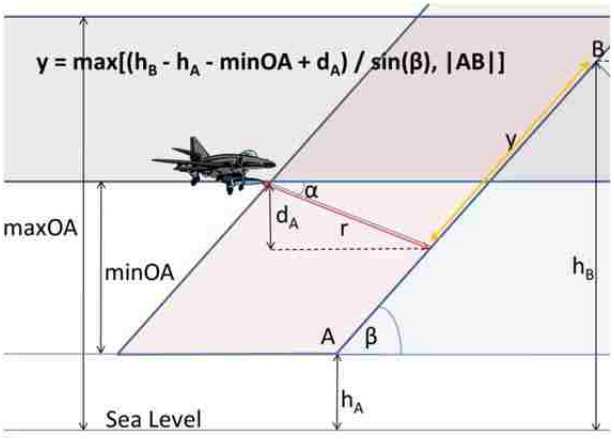

Figure 17: The geometric situation for Case 4 (i.e. minOA < h_{AB})

All these cases can be integrated in one formulation, and the slope coverage of a surveillance system for a specific cell can finally be computed as follows (provided that maxOA > h_B):

$$SC = \max[\max(d_A + h_{AB} - \text{minOA}, 0) / h_{AB}, 1]$$

5. Statistics and relevant information about the generated tours, which span:
    - The tour length, i.e. the distance traveled to perform the surveillance tour
    - The travel time, which is the period spends in transportation from one cell to another
    - The inspection time, which the period an aerial vehicle spends scanning a set of cells
    - The flight time, i.e. is the sum of the travel and inspection times
    - The cycle time, i.e. the sum of the flight time and downtime of the associated asset
    - The set of unreachable cells
    - The set of cells where the revisit-time is much higher than required
    - The highest and average revisit-times

Note: This set of outputs can also inspire future work such as the incorporation of asset reliability and maintenance scheduling in the model. We consider these to be out of the scope of this study.

CHAPTER 4: TERRAIN RISK ASSESSMENT

1. Background

1.1. Fuzzy Set Theory

Fuzzy Set Theory (FST) is a concept that extends the classical mathematical notion of sets (Zadeh, 1978). Instead of having an element be in exactly one of two or more disjoint sets, FST lets this element to partially belong to more than one set. This is accomplished by defining a *membership function* F, whose range is the interval [0,1], for each set S in order to map each element x in S to [0,1]. The value F(x) conveys the extent to which an element x belongs to the set S. A fuzzy set can therefore be formally defined as a pair (S,F) where S is a (classical) set and F is its membership function.

To illustrate this concept, consider the procedure of boiling cold water. The heating process is gradual, and the liquid goes through several states—cold, warm, hot—before it reaches the boiling point. During the transition from one state to another, there is obviously an overlap between some of the mutually exclusive states (i.e. between cold and warm, and between warm and hot) due to the gradual nature of change. This overlap can be captured by FST through the definition of three fuzzy sets for each state with fuzzy functions F_{cold}, F_{warm}, and F_{hot}. A temperature point x can then simultaneously belong to all three sets with specific membership degrees such as ($F_{cold}(x)$, $F_{warm}(x)$, $F_{hot}(x)$) = (0.5, 0.5, 0).

A more relevant example to this study is the scenario where an analyst is trying to assess the risk of a set of regions (assume that each region has a risk-score). This person is interested in assigning the inspected geo-locations to one of three sets: *low-risk* regions, *medium-risk* regions, and *high-risk* regions. The issue of interest is how to draw the border between the *low-risk* and *medium-risk* sets, as well as the *medium-risk* and *high-risk* sets, especially that such categorization can be subjective.

In this research effort, we focus on the set of regions (or terrain cells) that are *high-risk*, or in other words, assess the degree of their membership to the *high-riskiness* set. This is carried out by constructing a fuzzy membership function based on certain terrain attributes that are believed to give advantage to the adversary. Subsequently, the fuzzy membership value of a certain geo-location is adopted as a risk-score since FST provides a naturally standardized way to measure how *risky* an element is.

1.2. Risk Terrain Modeling

This study identifies an adequate and flexible tool that can be applied in the context of military surveillance in complex terrain to assist with evaluating how risky a cell can be. Here, risk refers to the degree of possibility of insurgent presence. This analytical tool is presented by Caplan and Kenedy (2010) and is called *Risk Terrain Modeling* (RTM). The authors have used it for criminology analysis to help the police forecast and locate future violent incidents in different neighborhoods so that the force can allocate resources more efficiently. Their strategy is based on selecting specific *risk-factors* that are believed or demonstrated to be correlated to the *outcome of interest*.

For illustration, Caplan and Kennedy provide the example of the *gun shooting incident* as the outcome of interest during the period of January 1 to June 31 in Irvington, NJ. The risk-factors they consider are: (1) gang members' residences, (2) retail outlets (e.g. liquor stores, fast-food places, bus stops, etc), and (3) areas with high concentration of drug arrests. Each one of these three aspects leads to the generation of a map, which is converted into a grid of values. The matrix cells are then selected or unselected through the comparison against a (given) threshold, and are assigned a value of 1 or 0, respectively. Finally, the grids are added together to create a composite map that shows the *hotspots*, which are scored from 1 to 3 (due to the existence of 3 risk-factors). For illustration, see Figure 18 and Figure 19.

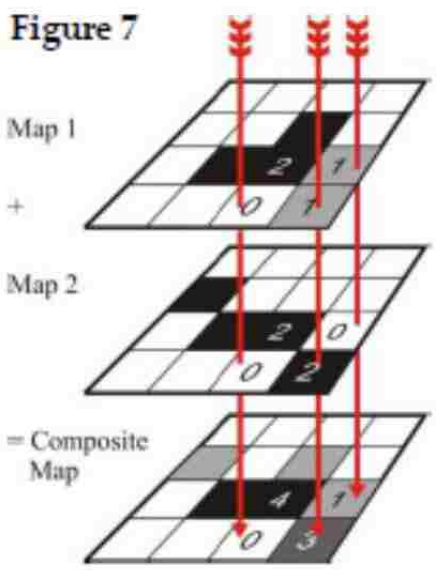

Figure 18: Process of adding up risk-factors to obtain the composite map

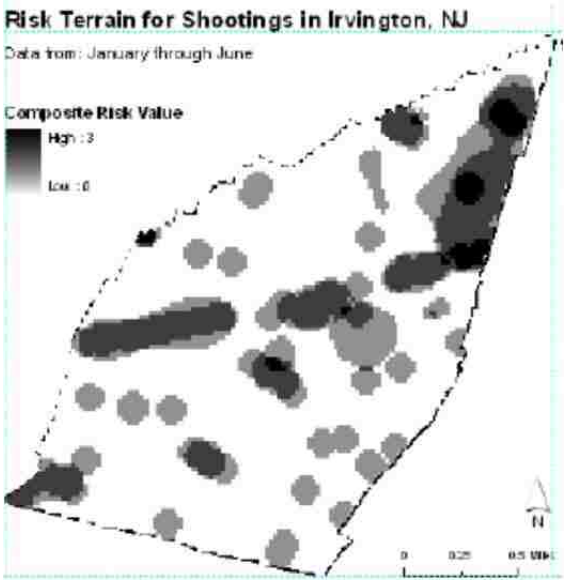

Figure 19: A composite map that incorporates the three risk-factors for the Irvington, NJ study (Courtesy of Caplan and Kennedy)

A similar strategy is followed in this study in order to identify hotspots in the terrain that is surrounding the outpost (also referred to as *named areas of interest* or NAIs in military contexts). However, two modifications to RTM are explored as a way to adapt the developed analytical tool to the studied problem. The first modification is concerned with the customization and incorporation of meaningful risk-factors as sometimes a lone terrain attribute does not tell the whole story about the risk level or the reason why a certain cell should be considered as a hotspot. When grouped with other geo-characteristics however, they convey a message about a certain scenario that can be valuable at predicting the attacker's actions. Consider for instance line-of-sight; this attribute does not provide any value if the region being analyzed is so far away from the outpost that even the most powerful weapon cannot be used. Another example is the advantage created by being on the higher grounds; nonetheless, extremely high altitudes that prevent targeting accuracy are quite probably not of any value. Now, take these two attributes and gather them in one formula along with a reasonable weapon's range. When the right distance meets the right altitude with line-of-sight, this gives birth to a situation that is likely to benefit a stealthy opponent; thus, these three measures are treated as a group.

Similarly, other types of attributes are aggregated into a comprehensive measure, and an associated score for RTM is generated using a special formula or algorithm, as discussed in the next section.

The second modification on the other hand is related to the prevention of underestimating risk when assigning scores to grid-cells. To explain this, consider a fictitious scenario where insurgents favor strong winds because that situation prevents UAVs from operating, they prefer villages and high-altitude caves to temporally settle down, and they use foliage to hide. Then a cell that has a relatively high altitude, some foliage, high winds, and is close to a village, may be of a tactical advantage to the adversary, and hence, the higher probability of their presence in this cell. The issue here is that it is possible for RTM to underestimate the importance of such a cell and dismiss it from the analysis because the cumulative effect of partial risks is not considered. To be more specific, RTM may assign a value of 0 to the target cell when generating the risk map associated with foliage—as only the values 1 and 0 are allowed as scores. Consider the case where the degree of foliage is just moderate; as a result, it is not large enough to qualify for a risk-score of 1. If however partial risk is permitted—say in the form of a number between 0 and 1 (e.g. 0.8 for altitude, 0.6 for foliage, 0.9 for winds, and 0.5 for proximity to the village)—then, the cell would have a different final score from a pure summation of 0's and 1's (e.g. 1+0+1+0 < 0.8+0.6+0.9+0.5), a fact that can make a difference in qualifying the target cell to be a hotspot.

To prevent the case of erroneously eliminating a cell from the risk analysis, FST is employed in order to address both types of needed modifications in a fluid and approximate fashion rather than a rigid discrete one. To be more precise, a fuzzy membership function F with an X-vector input can be employed to obtain an output between 0 and 1 (i.e. F(X) in [0,1]) to reflect how true is the *hotness* or *riskiness* of such a cell in terms of a certain risk-factor. Not only does this formulation enable the incorporation of more than one terrain attribute to obtain a risk-score, but it also ensures that the

output measure is in the function's continuous range [0,1], which adequately eliminates the use of inaccurate discrete binary values.

2. Risk-Factors

As previously stated, this study deals with the case where the adversary is smart and has good situational awareness. Therefore, an effort to predict their behavior is invested through analyzing different scenarios and taking into consideration multiple aspects that provide tactical advantage to insurgents. Consequently, five risk-factors are introduced along with their corresponding fuzzy function and weight coefficient, a scalar that represents their importance in the process of identifying hotspots:

- The *foliage camouflage* (FC) factor, with a function F_{FC} and a weight w_{FC}
- The factor pertaining to *proximity to population* (PTP), with F_{PTP} as its function and w_{PTP} as its weight
- The factor associated with the potential for *long-range attacks* (LRA), with a function F_{LRA} and a weight w_{LRA}
- The factor corresponding to *accessibility* to regions and spots, with a function F_A and a weight w_A
- The *expert input* (EI) factor, with a weight w_{EI} (note that no function is provided for this risk-factor—whose cell values might be extracted from historical incidents or insight gained from experience—mainly because it is meant to serve as a way for the analyst or surveillance architect to manually add other spots that are believed to be high-risk and missed by the developed decision tool)

These risk-factors are derived from a set of terrain attributes such as altitude, foliage, distance, line-of-sight, slope, trafficability, and demographic data. The relationship between each factor and its relevant attributes is established, modeled within the associated fuzzy membership function, and thoroughly explained in the forthcoming subsections. Given that the terrain is digitized into a grid of

cells, each cell j is assigned its corresponding membership values $F_{FC}(j)$, $F_{PTP}(j)$, $F_{LRA}(j)$, $F_A(j)$, and $F_{EI}(j)$, which are later integrated into one weighted average calculation—using the weight coefficients (note that this necessitates that the five weights sum up to 1). The result is a composite map that is represented by a grid matrix, whose elements are the final risk-scores. The cells with high scores are identified as *hotpots*.

2.1. Foliage Camouflage

High-level foliage regions can help intruders hide well and achieve successful steps towards the COP/OP. Therefore, this aspect has to be considered in a serious manner and should be countered with appropriate utilization of special technological assets such as thermal sensors. This importance further intensifies as the adversary's forces reach nearby spots since they may be able to position themselves in close and highly strategic positions right before a surprise deadly attack. Consequently, the score associated with the camouflage factor ought to be a function of both the foliage level and the proximity to the base. Note that for each cell j in the grid, such a function is to be positively correlated with the foliage level f_j and negatively related to the distance d_j from the center of cell j to the centroid of the outpost. When targeting the *high-riskiness* fuzzy set in this context, the camouflage factor can then be modeled through the following membership function:

$$F_{FC}(j) = (f_j/f_{max}) \,.\, (d_{max} - d_j)/d_{max} \qquad \text{for every cell j}$$

Note that the division by f_{max} and d_{max} ensures that the output of F_{FC} is in the interval [0,1], and hence the validity of its membership. f_{max} refers to the highest level of foliage possible, which holds a value of 3—a standard adopted RAND Corporation; this suggests that the foliage level is in the domain [0,3]. On the other hand, d_{max} is the maximal existent distance (from the base) in the grid, which can be computed through the maximum of the distances from the COP/OP to the four grid's corners. Figure 20 shows an example where d_{max} is the distance between the outpost's cell to the northeast corner cell.

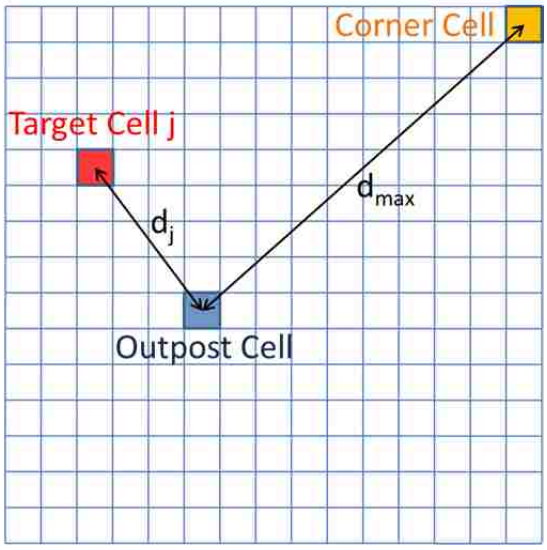

Figure 20: A grid setup where the outpost and the target cell j are in the western half of the grid

2.2. Proximity to Population

For insurgents, proximity to villages can prove very valuable; being amongst civilians does not only allow them to restock certain supplies and maybe access medical treatment, but it also helps them blend with the settlers and hide from the enemy. Furthermore, even if their presence is revealed, attacks might not happen as that would risk losing civilian lives. Another aspect about settlements is that sometimes population is scattered in smaller groups over a large area rather than grouped in one large city. In this section, a process is developed to take into consideration these aspects in order to generate cell scores that are associated with the advantage created by being close to one or more population settlements. Such a procedure starts by identifying cells in which there are inhabitants; and the larger the size of the populace, the better for the insurgents. This leads the associated fuzzy function to be in terms of both the size of and the distance from population settlements. When targeting the *high-riskiness* fuzzy set in this context, such a function is constructed in two steps, the first of which corresponds to the case where only one cell in the grid has inhabitants; the second step expands the computation to any number of inhabited cells.

- Step 1: Consider cell i, the only cell in the grid with human settlements, with an aggregate population size of s_i. The following formula is adopted to compute the risk-score of every cell j in the grid; such a score increases when s_i increases, and decreases when the distance d_{ij} between the centers of cell i and cell j decreases:

$$F_i(s_i,d_{ij}) = (s_i/s_{total}) \,.\, (d_{max} - d_{ij})/d_{max} \qquad \text{for every cell j}$$

where s_{total} refers to the total population of the grid, whereas d_{max} is the maximal existent distance in the grid—the diagonal of the grid—as illustrated in the example of Figure 21.Note that the division by s_{total} and d_{max} ensures that the output of F_i is in the interval [0,1], and hence the validity of its membership.

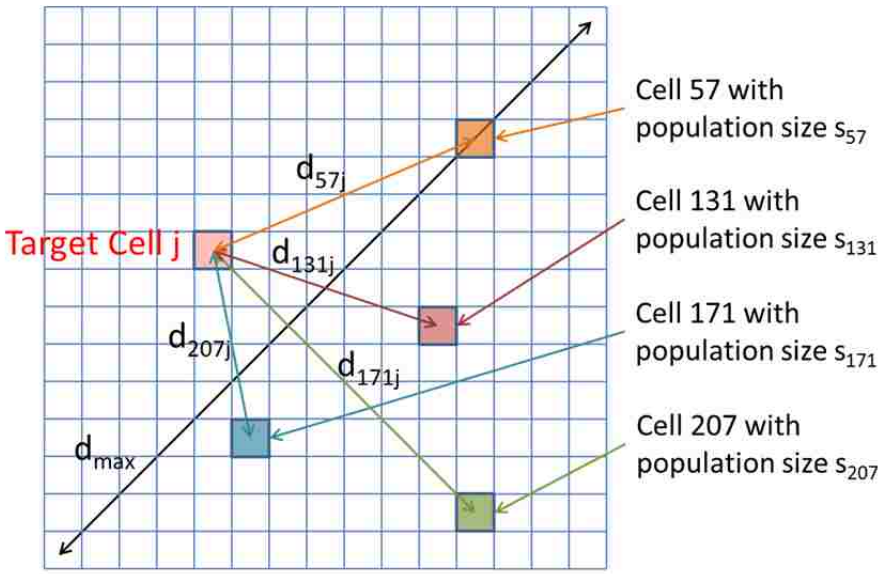

Figure 21: An example of a terrain grid with four populated cells

- Step2: To extend the scenario in the previous step to any number of populated cells—given that it ought to be encouraged to give more points to a cell that is close to most of the inhabited cells than a cell that is close to only few of them—the risk-scores $F_i(s_i,d_{ij})$ associated with a target cell

j are aggregated in an average computation as demonstrated in the following formula (also illustrated in Figure 21):

$$F_{PTP}(j) = \text{Average}_{\{i \text{ with } Fi(si,dij)\neq 0\}}(F_i(s_i,d_{ij})) \qquad \text{for every cell } j$$

2.3. Long-Range Attacks

Higher altitudes usually give a tactical advantage to the opponent, but not at all times. To explain, consider the case where the adversary has long-range arms. Certain regions around the outpost would generally be ideal for the attacker if three conditions are verified (also depicted in Figure 22):

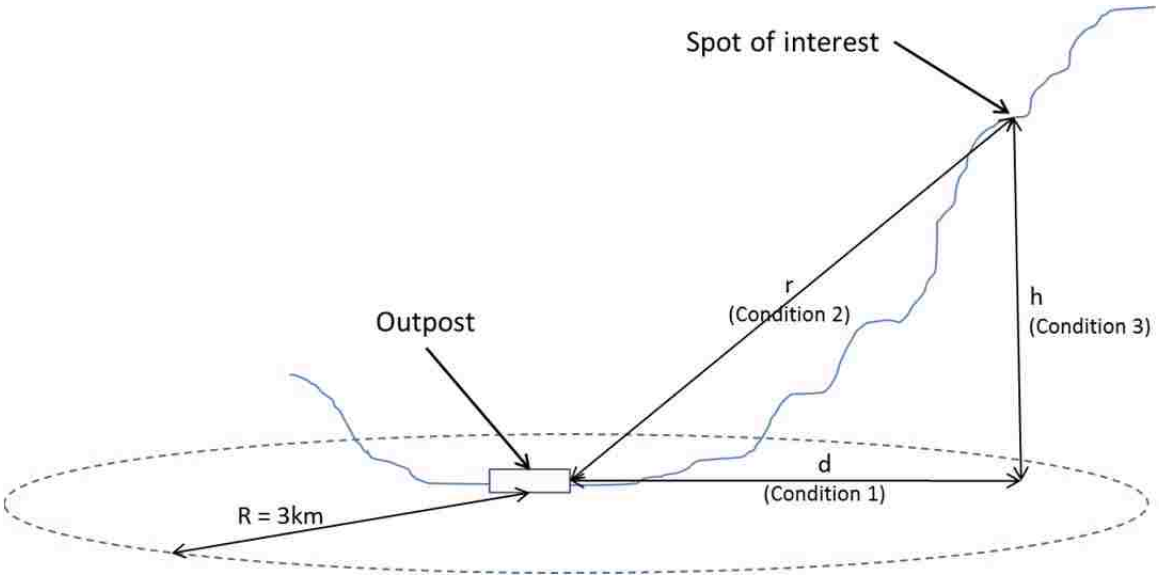

Figure 22: Illustration of the three ideal conditions and their associated parameters

- Condition 1: The geological spot of interest is close enough to the outpost where long-range weapons become a threat. Expert feedback reveals that a 3km radius around the base is a good approximation as it includes ranges for the powerful portable mortars. For notation simplicity, let C1 represent the logical proposition *Condition 1 is verified*; i.e., C1 holds a value of *true* when the distance between the outpost and the cell of interest is less than 3km.
- Condition 2: For weapons to be fired successfully and accurately, the adversary must have line-of-sight from their temporary setup spot. Obstructions not only act as potential barriers for the

projectile, but they also cause hindrance to the shooter's precision. A challenge with this analysis is finding out whether or not a cell has line-of-sight of the outpost in a computational manner. This is addressed by drawing a line from the cell of interest to the closest outpost cell (center to center) and identifying all the intersections with this line. This reveals the cells that have to be crossed by the projectile in order to reach the base as illustrated in Figure 23; such cells are referred to as *intermediary cells* in this context.

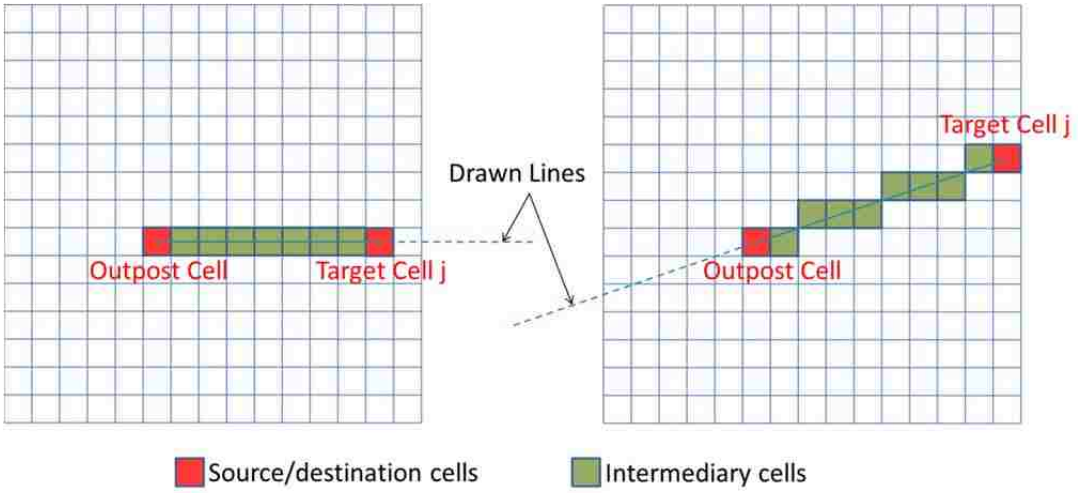

Figure 23: Two examples of finding intermediary cells

These identified cells have to have an elevation that is less than a certain threshold; otherwise, they constitute an obstacle. For instance, Figure 24 illustrates a case where one cell causes obstruction between the outpost cell and the cell of interest; these two cells are separated by six intermediary cells that the line-of-sight crosses. To determine this threshold value, which depends on the proximity of the associated cell to the outpost, consider the following notations:

- d_{ij} is the distance between the centers of cell i and cell j
- h_{ij} is the ASL elevation difference between cell i and cell j
- H_i is the threshold to exceed in order for a cell i to be an obstruction
- ϕ is the angle between the target line-of-sign and sea level

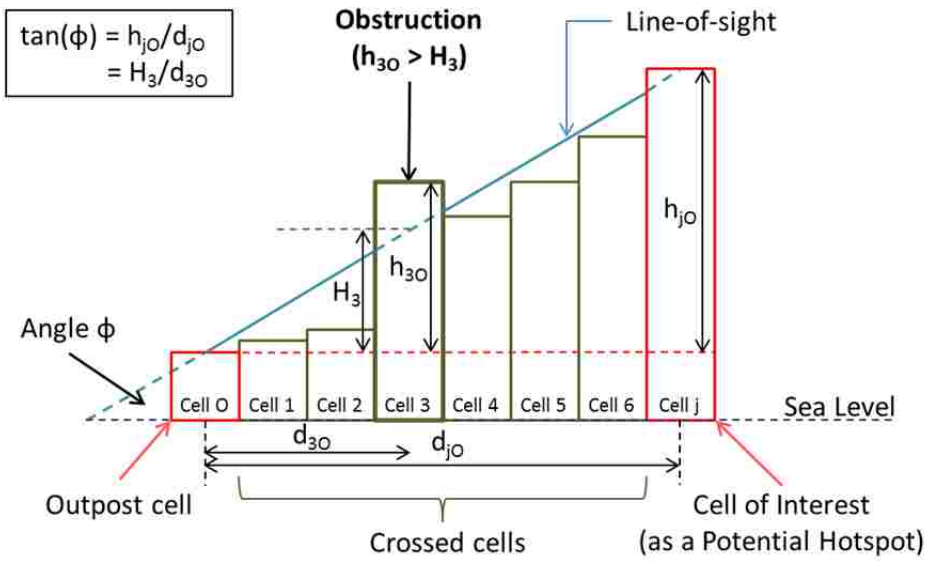

Figure 24: An Illustration of a cell causing obstruction

For an intermediate cell i (between the outpost cell O and the target cell j), the following equation can be derived from the geometry presented in Figure 24:

$$\tan(\phi) = h_{jO}/d_{jO} = H_i/d_{iO}$$

Hence:

$$H_i = d_{iO} \,.\, h_{jO} \,/\, d_{jO}$$

In conclusion, there is an unblocked line-of-sight between a target cell j and an outpost cell O if and only if for every intermediary cell i, the threshold value H_i is larger than ($d_{iO} \,.\, h_{jO} \,/\, d_{jO}$). For notation simplicity, let C2 represent the logical proposition *Condition 2 is verified*; i.e., C2 holds a value of *true* when line-of-sight between the outpost and the cell of interest is not obstructed.

- Condition C3: High altitudes in this scenario help in two ways. First, they render it difficult to spot and reach the opponent (if spotted). Second, they make it easier on the shooter to strike the outpost with a firearm attack. However, this is only correct if the elevation is not great enough to cause targeting problems. According to experts, a difference of h_o=150m in elevation is ideal, and beyond h_{max}=300m is ineffective. When targeting the *high-riskiness* fuzzy set in this

context, we adopt the fuzzy function depicted in Figure 25—whose input shall be h_{jO}, the elevation difference between a given cell j and the outpost cell O.

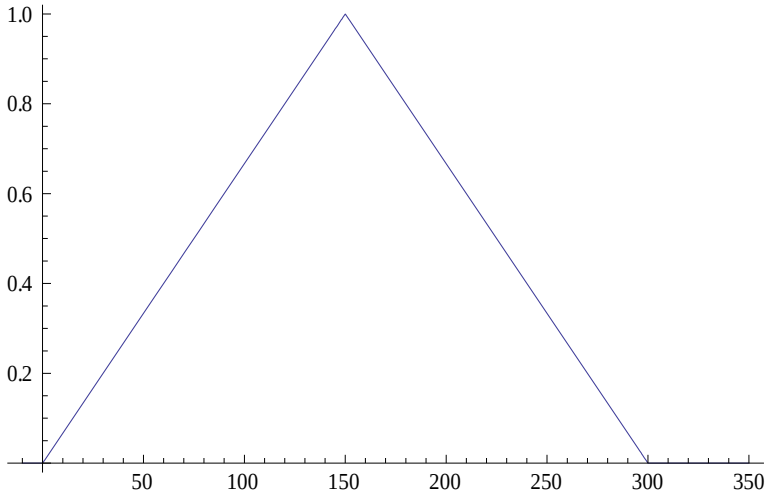

Figure 25: The fuzzy membership function in terms of elevation difference h_{jO} (in meters)

Lastly, the three conditions are to be integrated in one function. Given the Boolean nature of the first and second conditions as well as the fuzzy membership function associated with Condition 3, a score for the LRA risk-factor can be computed for every cell in the grid as follows:

$$F_{LRA}(j) = \begin{cases} h_{jO} / h_o & \text{if } 0 \leq h_{jO} < h_o \ \&\ C1 \ \&\ C2 \\ (h_{max} - h_{jO}) / (h_{max} - h_o) & \text{if } h_o \leq h_{jO} \leq h_{max} \ \&\ C1 \ \&\ C2 \\ 0 & \text{otherwise} \end{cases} \quad \text{for every cell j}$$

Note: the symbol & in the above formula refers to the *logic-and* operator.

2.4. Accessibility to Regions

A geo-location might be of a high tactical advantage, but it becomes useless if it is not accessible. In this section, the concept of *accessibility* is developed through several steps to attain a formula that considers the relevant attributes.

First, consider an adversary who is very familiar and comfortable with complex terrain; as a result, he is less likely to be bothered by high slopes, and is more likely to take high-slope paths as they are less predictable by his enemies. However, there is a limit on how steep terrain can be to yield

advancement as too much slope starts becoming a hindrance (usually of an angle between 45 and 60 degrees), especially if a heavy weapon is carried. For modeling purposes, let β_o be the angle at which slope starts becoming a burden, and let β_{max} be the angle at which slope prevents advancement. When targeting the *high-riskiness* fuzzy set in this context, a helper fuzzy function that is based purely on slope is modeled through the following formula (an instance of which is plotted in Figure 26):

$$F_\beta(\beta_j) = \begin{cases} 1 & \text{if } 0 \leq \beta_j < \beta_o \\ (\beta_{max} - \beta_j) / (\beta_{max} - \beta_o) & \text{if } \beta_o \leq \beta_j \leq \beta_{max} \\ 0 & \text{otherwise} \end{cases}$$

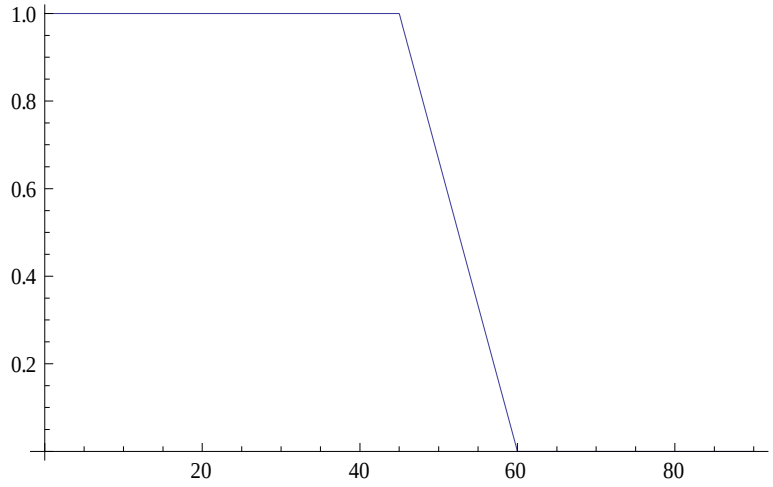

Figure 26: The fuzzy membership function F_β given that $\beta_o = 45^o$ and $\beta_{max} = 60^o$

This leads to the next point, which is: given a spot that has high but manageable slope, it is valuable only if it is actually accessible from other regions. In other words, if it is surrounded by steep-terrain cells, it is most likely avoided by insurgents, even if it turns out to have apt characteristics such as a fair amount of foliage, proximity to the outpost, line-of-sight, etc. This perspective adds a bit more complexity to the aspect of *accessibility* as cells with attractive attributes become even more valuable if they are easily accessible and allow access to other attractive cells. Therefore, when evaluating the slope β_j for a specific cell j, two questions are raised: (1) Is cell j accessible through its adjacent cells? (2) Does cell j give accessibility to its surrounding cells? Such aspects can be captured by incorporating the slope-

based scores of the adjacent cells through a calculation of average when seeking the *accessibility* risk-score of the target cell, as follows:

$$[F_\beta(\beta_j) + \sum_{\{i \text{ adjacent to } j\}} F_\beta(\beta_i)] / k_j$$

where k_j is the number of surrounding cells of cell j (note: k_j is equal to 3 for corner cells, 5 for edge cells, and 8 for inside cells).

Last but not least, there is a missing exception to the fact that a stealthy adversary is more likely to avoid flat land and low-slope areas because that is too obvious or predictable. Yet, if a fairly flat path is highly trafficable, it might be used by vehicles to reach the base in a short timeframe. Swift attacks have shown in past events to be effective in inflicting serious damage, even when predictable paths are taken. This notion of *trafficability* is hence used to expand the concept of *accessibility* to account for swift attacks. To understand this newly introduced measure, which is a function of the terrain's soil texture and water saturation, consider the following figures:

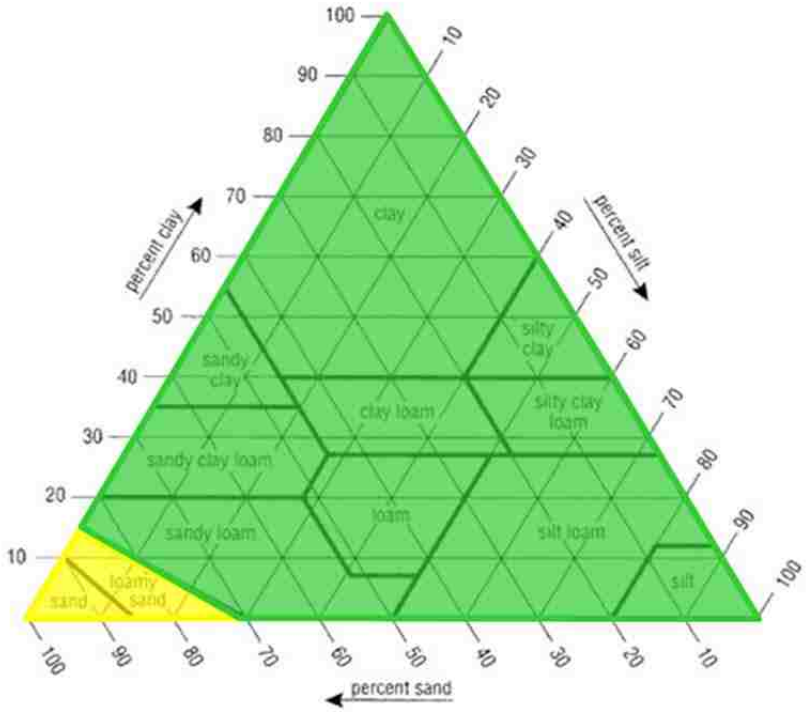

Figure 27: Trafficability assessment and categorization for dry terrain
(Courtesy of RAND Corporation)

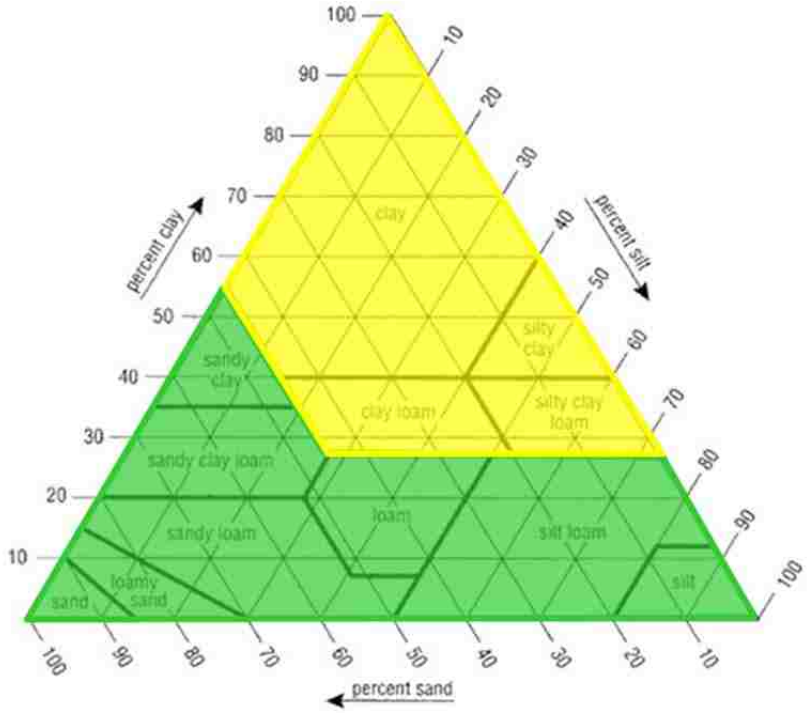

Figure 28: Trafficability assessment and categorization for moist terrain
(Courtesy of RAND Corporation)

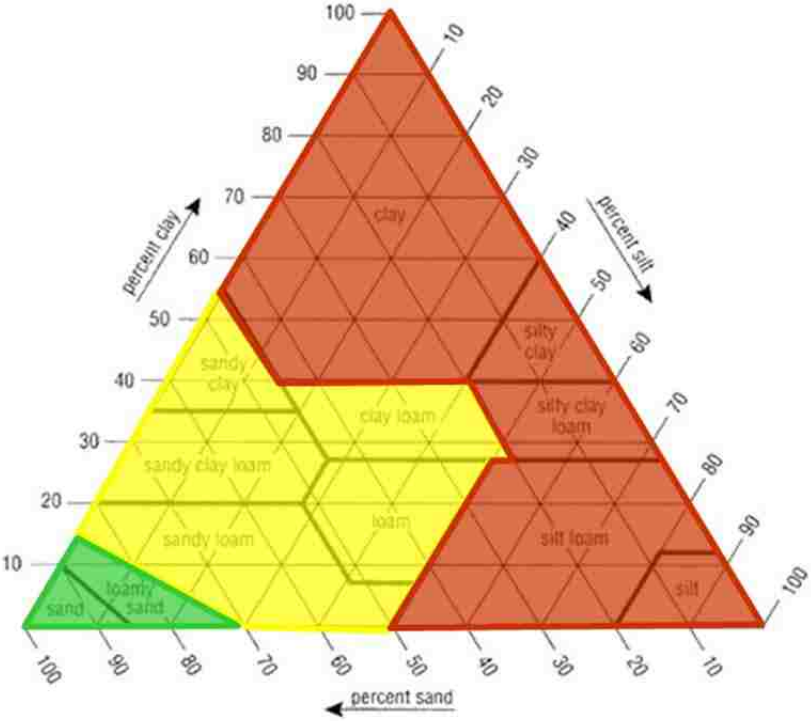

Figure 29: Trafficability assessment and categorization for a full-saturation terrain
(Courtesy of RAND Corporation)

Table 4: Legend for Figure 27, Figure 28, and Figure 29

Trafficability Assessment	Code
Fair	2
Poor	1
No-go	0

As can be seen from these charts, there is a decent amount of situations when land is fairly trafficable. Therefore, this attribute has to be captured in the analysis when terrain is believed to be flat enough. This can be achieved through the fuzzy function plotted in Figure 30, given that the trafficability of cell j is represented by a scalar t_j in the domain [0,2], and that the slope angle is less than a threshold value β_{min} (usually about 30^o).

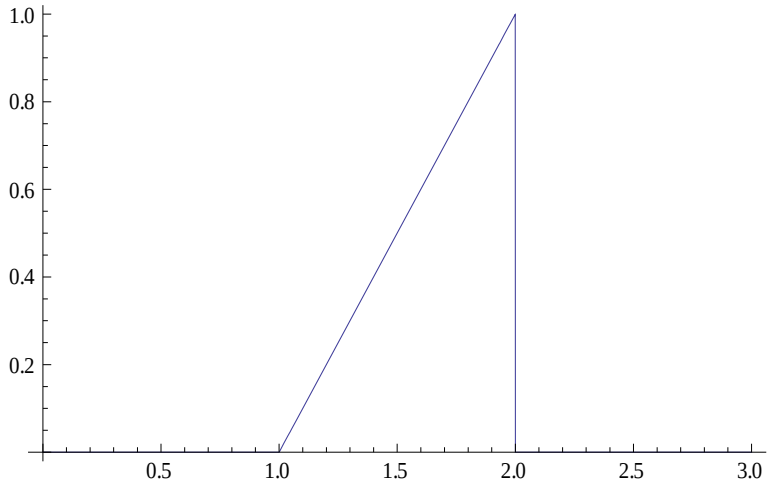

Figure 30: The fuzzy membership function in terms of trafficability t_j

Finally, by combining the two discussed attributes—slope and trafficability—into one function, the accessibility risk factor can be computed for every cell in the grid as follows:

$$F_A(j) = \begin{cases} (t_j - 1) \cdot [F_\beta(\beta_j) + \sum_{\{i \text{ adjacent to } j\}} F_\beta(\beta_i)] / k_j & \text{if } 0 \le \beta_j \le \beta_{min} \ \& \ 1 \le t_j \le 2 \\ [F_\beta(\beta_j) + \sum_{\{i \text{ adjacent to } j\}} F_\beta(\beta_i)] / k_j & \text{if } \beta_{min} < \beta_j \\ 0 & \text{otherwise} \end{cases} \quad \text{for every cell } j$$

3. Composite Map

To recapitulate, five risk-factors are introduced along with their relationship with terrain attributes. The presented methodology is based on treating terrain attributes as inputs, extracting risk-scores for each factor in the form of a grid matrix, and combining these matrices into a *composite map* (as shown in Figure 31).

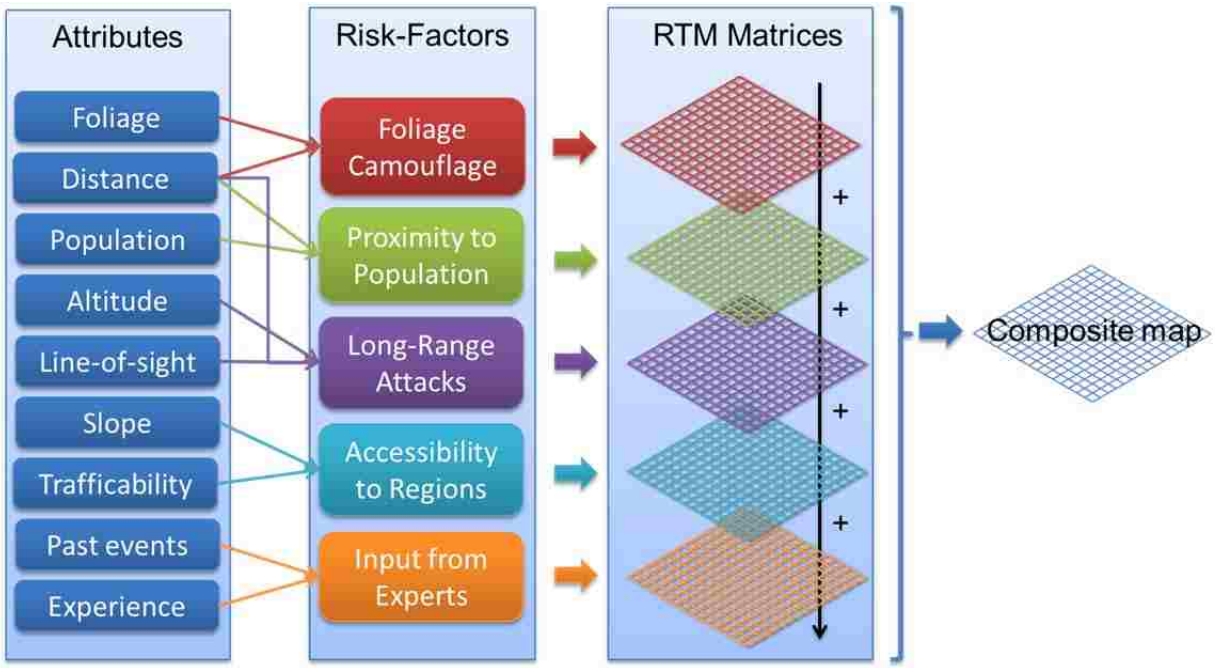

Figure 31: The process used in terrain evaluation in order to generate the composite map

To be more specific, geological characteristics such as slope, altitude, foliage, along with demographical information, are intelligently combined to produce formulas that yield the generation of scores for each risk-factor. Such scores convey the degree of how desirable a geo-location can be to insurgents, and hence, their risk level. Each one of the risk-scores stems from a fuzzification process, which helps ensure that the output number is in the range [0,1]. After all the risk-factor's scores are computed, they are aggregated through a weighted average to produce an RTM-based composite map, whose matrix elements are denoted by Risk(j) and are guaranteed to carry values between 0 and 1:

$$Risk(j) = w_{FC} \cdot F_{FC}(j) + w_{PTP} \cdot F_{PTP}(j) + w_{LRA} \cdot F_{LRA}(j) + w_{A} \cdot F_{A}(j) + w_{EI} \cdot F_{EI}(j)$$

where: $w_{FC} + w_{PTP} + w_{LRA} + w_A + w_{EI} = 1$

Weights can be adjusted by the user of the analytical framework to reflect different degrees of importance amongst the factors. Notice that this method is flexible as it allows an easy integration of other attributes and factors, assuming the score-generating function is appropriately constructed. Once the risk-scores are computed and the final (composite map) matrix is generated, the user can then choose the threshold at which a cell can be considered as a hotspot. Alternatively, s/he can opt to use a certain percentile to separate the high-risk cells from the rest.

The cell scores happen to have additional usefulness, and that is associated with the frequency at which they have to be revisited during the surveillance process. In other words, the higher the risk of a hotspot, the shorter the revisit-time for this cell ought to be. This fact can be reflected through a function that converts the cell score Risk(j) to a revisit-time τ_j, which is later used in a time-based resource allocation of assets as part of building the surveillance plan. Such function can be sophistically constructed or be as simple as an interval-based or linear one, depending on how the risk-scores are distributed and how the analyst wants to handle the conversion procedure.

CHAPTER 5: TOUR GENERATION

1. Concept

As seen in the previous chapter, the terrain risk assessment phase ends with generating a list of high-risk geo-locations (called hotspots) that need to be visited regularly for surveillance purposes. These hotspots basically constitute a subset of the cells in the digitization grid that has been already established. To model the motion of a mobile platform in such a structure, every grid cell is represented as a node, and an edge is created to link such a node to its adjacent nodes, as shown in Figure 32. The motion of a vehicle from a cell to another cell is assumed to be a center-to-center vertical, horizontal, or diagonal move on the 2D grid.

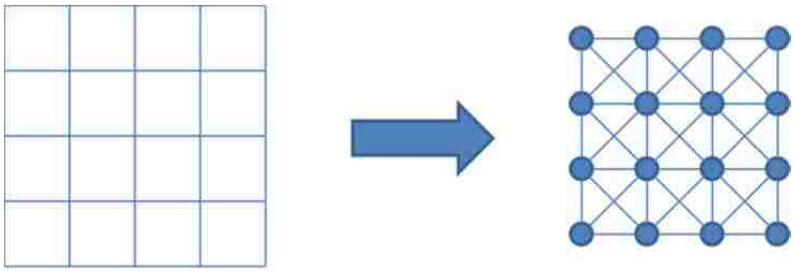

Figure 32: The network infrastructure obtained from transforming the 2D digitization grid

This representation provides a network infrastructure that supports the construction of tours comprising (hopefully all) high-risk nodes along their paths. These tours however are only generated for tactical assets (such as the Shadow and the Raven) as this process is obviously inappropriate for stationary assets (e.g. towers and aerostats), and inadequate for strategic systems since they fly at high altitudes and are able to scan large areas, generally at detection level. To clarify some more, the purpose of planning routes for tactical vehicles is to find low-altitude paths—which are not always guaranteed to be feasible—in order to approach high-risk regions as closely as possible and perform a surveillance routine at identification level.

Nonetheless, generating such tours is a challenging task because several complex issues have to be considered at this stage. First, the tour has to consist of a sequence of visits and paths that minimize fuel consumption. Second, only cells and edges that are feasibly traversable by the UAV are to be selected. Third, an apt operating altitude has to be found in a way that achieve acceptable image resolution with adequate stealth, while avoiding unnecessary altitude changes that lead to more fuel/battery (or endurance) consumption. Fourth, the computational expense of solving such a complex problem grows polynomially with the number of nodes, and edges eventually. Lastly, this analysis has to be performed for every tactical UAV.

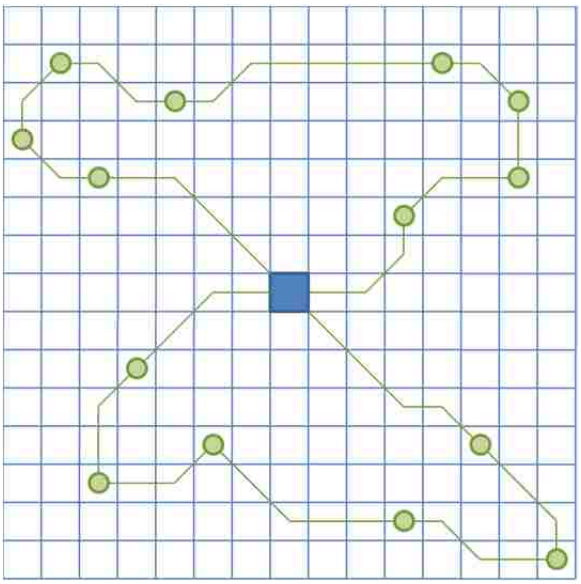

Figure 33: Synthetic flight tours planned for the RQ-7 Shadow (no infeasible spots)

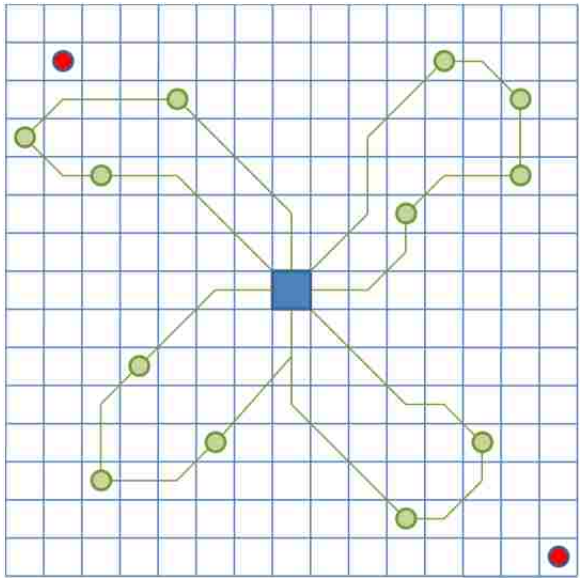

Figure 34: Synthetic flight tours planned for the RQ-11 Raven (two infeasible spots)

Note that this route generation phase is expected to produce different results for each tactical vehicle as they have different capabilities. To illustrate this point, consider planning two surveillance missions, one with the RQ-7 Shadow and the other one with the RQ-11 Raven, for the same set of regions. For this, a fictitious scenario is presented in Figure 33 and Figure 34 as an example. Notice that the shadow is able to visit all the required spots in two tours, with the need to refuel only once. The Raven however cannot visit the northwestern point because it is situated on high grounds. The UAV also

misses the southeastern spot because it is too far from the outpost and requires more endurance to be reached. Moreover, this aircraft's battery needs to be recharged three times, which increases idle-time during the mission. As can be seen, the Shadow might be favored over the Raven with respect to most characteristics; still, the latter has to be considered as it requires fewer personnel than the former, and might actually be chosen to carry out the mission for that specific reason.

In order to handle the complexity of the path planning process for each one of the considered UAVs, an approach with diverse techniques is exploited. Firstly, the tour generation problem is partitioned into a series of sequential sub-problems (or stages), which is a *divide-and-conquer* tactic that helps lessen the involved difficulty. Such sub-problems include the Traveling Salesman Problem (to determine the sequence in which the set of appropriate hotspots are to be visited), the Shortest Path Problem (to find the shortest path from one hotspot to another), and tour segmentation (to generate subtours that can be feasibility traveled given the endurance of the specified UAV). Secondly, a select subset of the constraints of interests is addressed outside the introduced *Mathematical Programming* formulations. Let such a group be referred to as *exogenous constraints*; subsequently, let *endogenous constraints* be defined in this context as the restraints that are explicitly formulated within the optimization models. The benefits that stem from handling constraints *exogenously* are twofold: (1) the number of endogenous constrains is reduced, which leads to smaller search spaces for the MP solvers (i.e. a smaller set of extreme points); and (2) the established network infrastructure can be reduced by filtering out the nodes that fail the verification of exogenous constraints, which leads to smaller input data sets for the MP solvers (i.e. less node and edges to choose from), and hence faster runtimes. Table 5 reveals the identified constraints, their classification, and the stage in which they are integrated.

Table 5: Types of involved constraints and their stage of integration

Constraint	**Type**	**Solution Stage**
Compatibility*	Exogenous	Input data processing
Payload*	Exogenous	Input data processing

Communication range	Exogenous	Pre-TSP
Terrain elevation	Exogenous	Pre-TSP
Wind level	Exogenous	Pre-TSP
Min/max operating altitude	Endogenous	Altitude Optimization
Climb-/dive-rate	Endogenous	Altitude Optimization
Noise (dB-) distance	Endogenous	Altitude Optimization
Sensing distance	Endogenous	Altitude Optimization
Endurance	Exogenous	Hotspot elimination, tour segmentation
Wireless interference	Negligible	-
Turn radius	Out of scope	-

*Note: it is assumed that the constraints associated with the compatibility between the platform and its sensors as well as the non-exceeded payload capacity of the platform are satisfied at this point, as they are easily verified at the beginning of the computation run through simple checks (especially that the compatibility matrix is very sparse).

Thirdly, an efficient way that yields the consideration of the third spatial dimension is sought. With such a mindset, modeling space using a 3D grid is avoided as it renders the problem unnecessarily much larger. Instead, a different approach is adopted to achieve the 3D aspect of the problem: after the 2D flight paths are found, each node in these paths is coupled with a new variable that shall specify a feasible operating altitude of the target UAV at that cell.

As a result of this multi-technique approach, a sequence of computational stages is established (as depicted in Figure 35) to solve the tour generation problem:

- Stage 1: a network reduction is performed by employing the appropriate exogenous constraints to evaluate the grid cells and eliminate the nodes that are infeasible or unreachable, along with the appropriate edges. The outcome of such a process is a leaner network infrastructure that shall ease the computations in the forthcoming steps.
- Stage 2: the high-level strategy of conducting surveillance over the reachable hotspots is devised by attaining an optimized sequence of visits through a TSP solution that is based on approximate distances. The objective of such optimization is to minimize the traveled distance by the UAV in order to expedite the cycle of visits and reduce energy depletion.

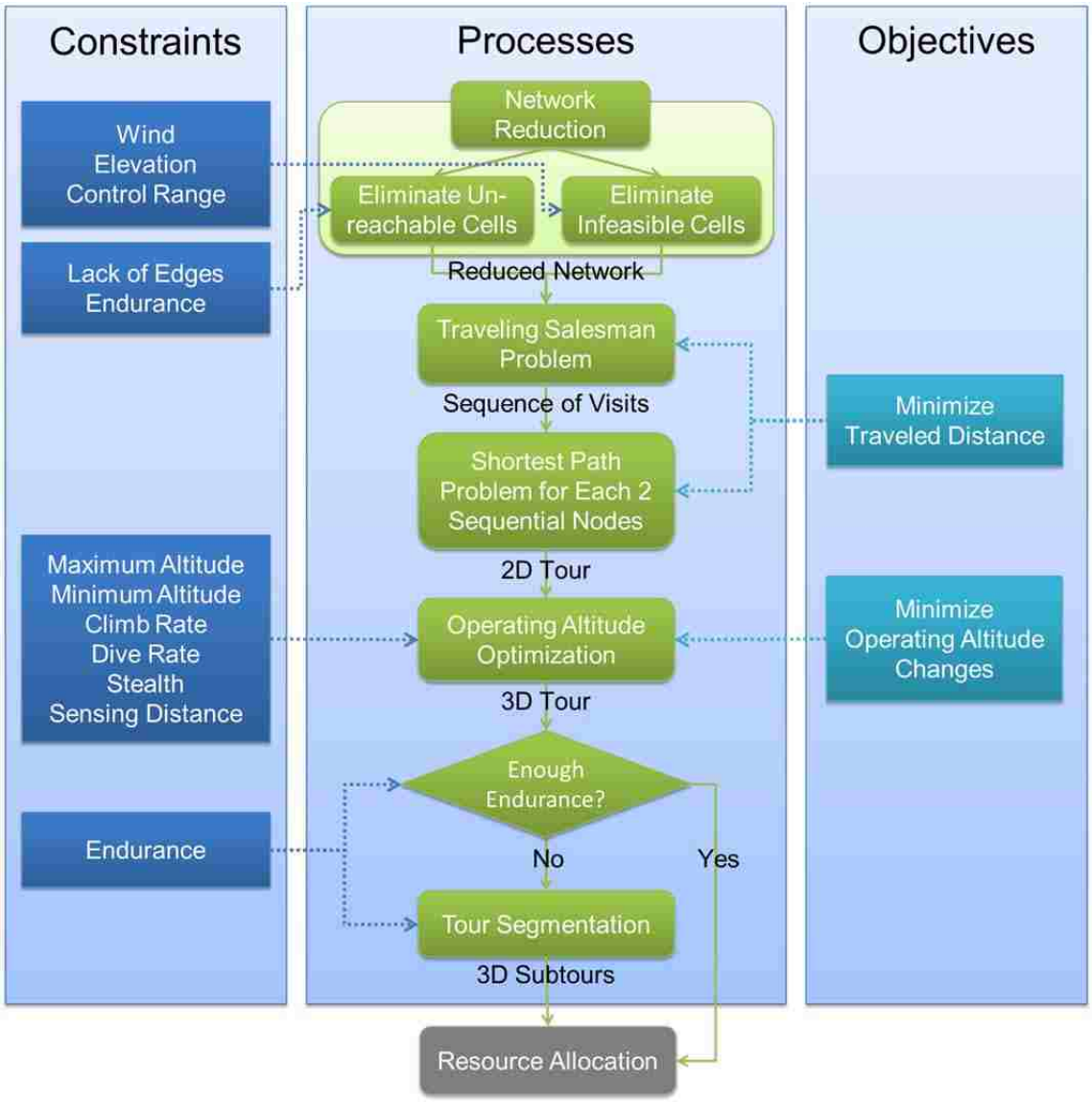

Figure 35: The sequence of steps to generate (sub)tours for a UAV

- Stage 3: each two consecutive nodes in the generated TSP sequence need to be connected by a feasible and efficient path; this is achieved by solving the Shortest Path Problem for each TSP segment. Note that an alternative (non-adopted) strategy is to solve the SPP for any pair of hotspots and find a more accurate travel cost before solving the TSP; however, the computation time would be much greater, while the tour is expected to change just a bit. Conversely, the current framework executes the SPP model only when necessary, which leads the computation to run in a much faster fashion.

- Stage 4: this step of the process is concerned with determining the most adequate operating altitude at each traversed cell in the UAV's trip. The vector comprising the set of OA variables is resolved using a custom linear program, through which flight climbs and dives are planned such that the sensing distance and stealth constraints are verified while the change in operating altitude is feasibly minimized.
- Stage 5: Tour segmentation is a conditional routine as it is executed only when the generated surveillance trip is too long to be handled by the aerial vehicle. In that case, the endurance constraint is used to split this 3D tour into subtours so as to obtain a feasible flight plan.

Each one of these stages is explained in more details in the forthcoming subsections.

2. Network Reduction

When a grid is represented as a set of nodes and edges as mentioned previously, the network seems to have a very robust infrastructure with a large amount of redundancies. However, depending on the capability of the utilized UAV, several of these nodes and edges may not be possible to traverse, especially if the terrain has a complex geo-surface and is frequently subject to harsh weather conditions. Therefore, the concept of exogenous constraints is employed to reduce the preliminary network infrastructure for every UAV considered in the analysis before the commencement of the tour generation. This is achieved through the comparison of the platform's characteristics against the terrain attributes. To be more specific, for a node to be *feasible* (for a certain UAV), it has to satisfy the following constraints: (1) its elevation has to be low enough so the UAV can fly over it, (2) its location has to be within the communication range of the UAV, and (3) its wind level has to be low enough not to disturb the flight of the UAV.

In case of infeasibility, the node and its associated edges are omitted from the network. This may give birth to other phenomena: *islands*, *isolated nodes*, and *infrastructure holes*, as depicted in

Figure 36. When nodes and edges are omitted from the network, and islands and isolated nodes appear, some cells become *unreachable*. That being said, reachable nodes can be computationally identified by solving the SPP from an outpost cell to the target cell. If this target cell is a hotspot, the found shortest path is referred to as a *link*, because it links the outpost to the surveillance tour. A link also helps identify and address another type of unreachability, which is related to the case where the traveled path is too long for the UAV to endure during a round-trip. The endurance constraint can therefore be used to further omit nodes that are too far from the outpost. (Note that for computational efficiency purposes, the check for unreachability can be performed on hotspots only instead of all cells, and this would not alter the outcome of the analysis.)

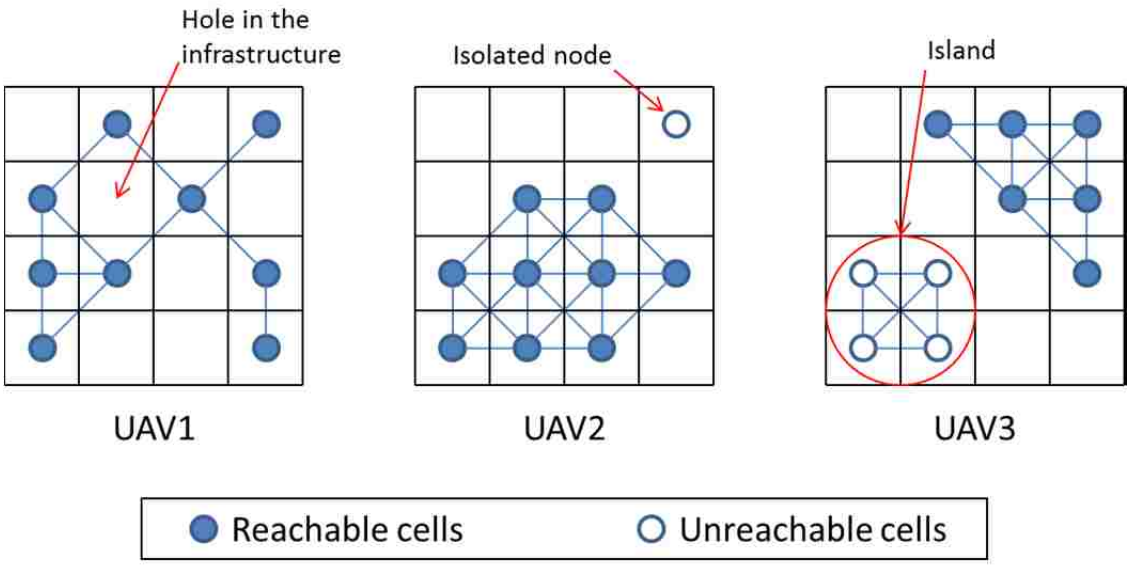

Figure 36: Three network infrastructures associated with three different UAVs

As a result of identifying infeasible and unreachable cells, the original list of high-risk regions has to be adjusted. To explain, any hotspot that coincides with an unfeasible or unreachable node has to be omitted from the analysis for the currently evaluated UAV. To visualize this, consider the real-world example depicted in Figure 37, which shows the proposed surveillance tour based on 30 randomly generated hotspots for the Shadow in the 5x5 km^2 Wanat region that surrounds COP Kahler. The colors significance is as follows: hotspots in red, outpost cells in green, network infrastructure in black, links in

cyan, and the surveillance tour in orange. Notice that because of the network reduction, 14 out of the 30 hotspots are excluded from the UAV's trajectory (mostly because of high elevation or strong winds). Moreover, one of these cells—called *Cell A* in the figure—is situated in an *infrastructure hole* that is close to the outpost. As for islands, there seems to be none in this case.

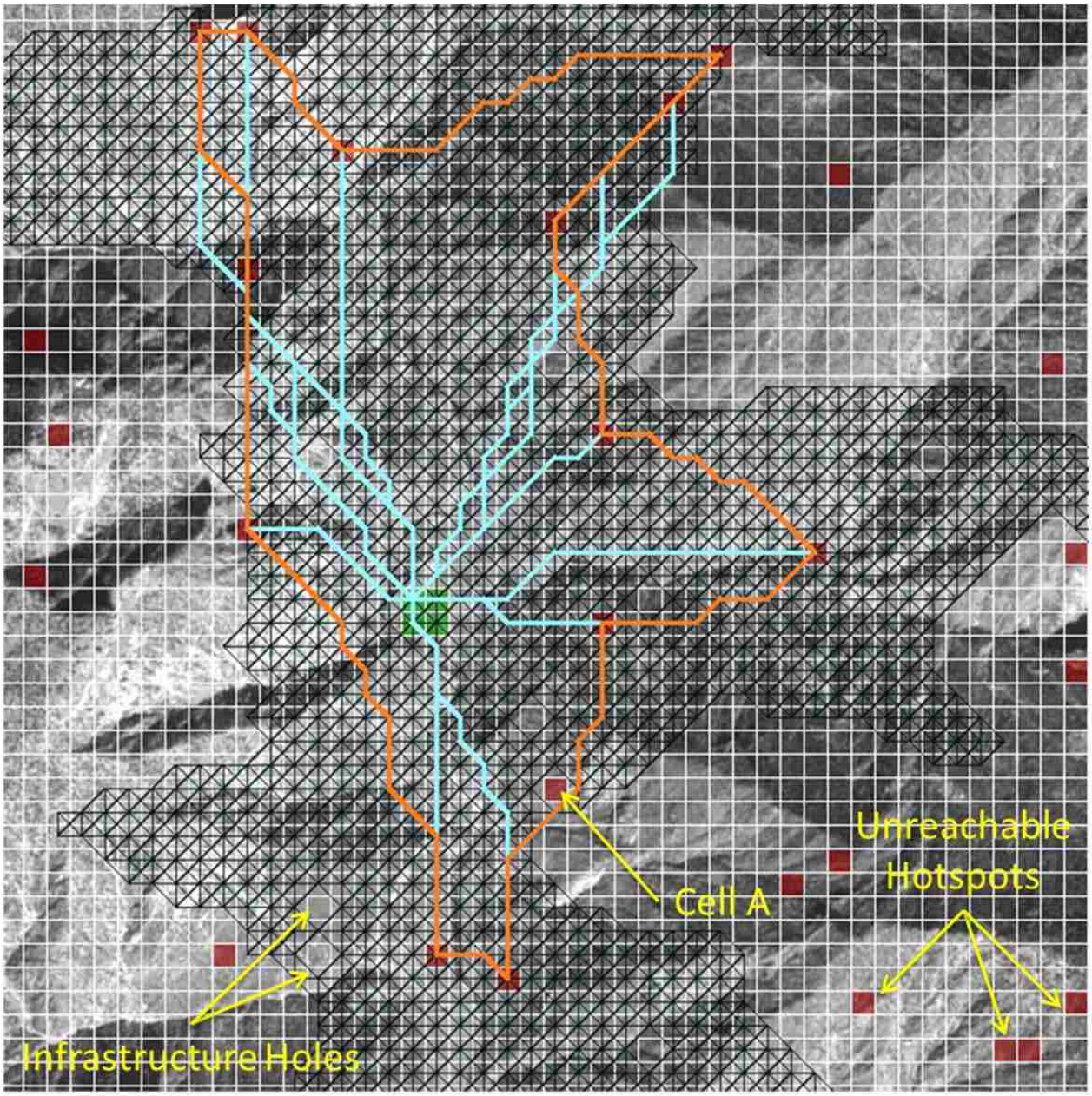

Figure 37: The reduced network for the RQ-7 Shadow in a 5x5 km^2 Wanat region

3. Tour Construction

3.1. TSP Heuristic

Now that a virtual transportation infrastructure is constructed, and only the feasible and reachable hotspots are included in the analysis, the next step is to generate an efficient sequence of high-risk cells to be visited through the surveillance tour. This is achieved by solving the (unconstrained) Traveling Salesman Problem using the formulation below, by Miller et al. (1960). In this context, the cost of traveling from one node to another is the Euclidean distance between them (which is assumed to be linearly proportional to the UAV's endurance). The AMPL code for such a formulation, provided by Lee and Raffensperger (2006), is included in Section 1 of the Appendix.

n	number of nodes
c_{ij}	cost of travelling from node i to node j
x_{ij}	1 if trip from node i to node j is chosen, 0 otherwise
y_i	number of so-far visited nodes when at node i

$$\text{Min} \sum_{\{i \text{ in } 1..n,\ j \text{ in } 1..n,\ i \neq j\}} c_{ij} \cdot x_{ij}$$

S.t.	$\sum_{\{i \text{ in } 1..n,\ i \neq j\}} x_{ij} = 1$	$\forall j \text{ in } 1..n$	(1)
	$\sum_{\{j \text{ in } 1..n,\ i \neq j\}} x_{ij} = 1$	$\forall i \text{ in } 1..n$	(2)
	$y_i - y_j + n \cdot x_{ij} \leq n - 1$	$\forall i \text{ in } 2..n, \forall j \text{ in } 2..n, i \neq j$	(3)
	$x_{ij} \text{ in } \{0,1\}$	$\forall i \text{ in } 1..n, \forall j \text{ in } 1..n$	(4)
	$y_i \geq 0$ and integer	$\forall i \text{ in } 1..n$	(5)

Constraints (1) and (2) ensure that every node is entered and left exactly once, whereas constraint (3) prevents incomplete tours (i.e. tours having less than n nodes).

The problem with solving the TSP with an exact method is the polynomial growth of the problem size, whereas in this study, a good approximation is sufficient. Consequently, an efficient heuristic is developed, and its performance is compared against that of the CPLEX solver, as shown in Table 6. This heuristic is inspired from the fact that solving a TSP with n nodes is much computationally expensive than solving two TSPs with n total nodes and then linking the two resultant subtours together.

In addition, the presence of a grid structure renders it easy to split geographical locations and acquire appropriate subsets of nodes.

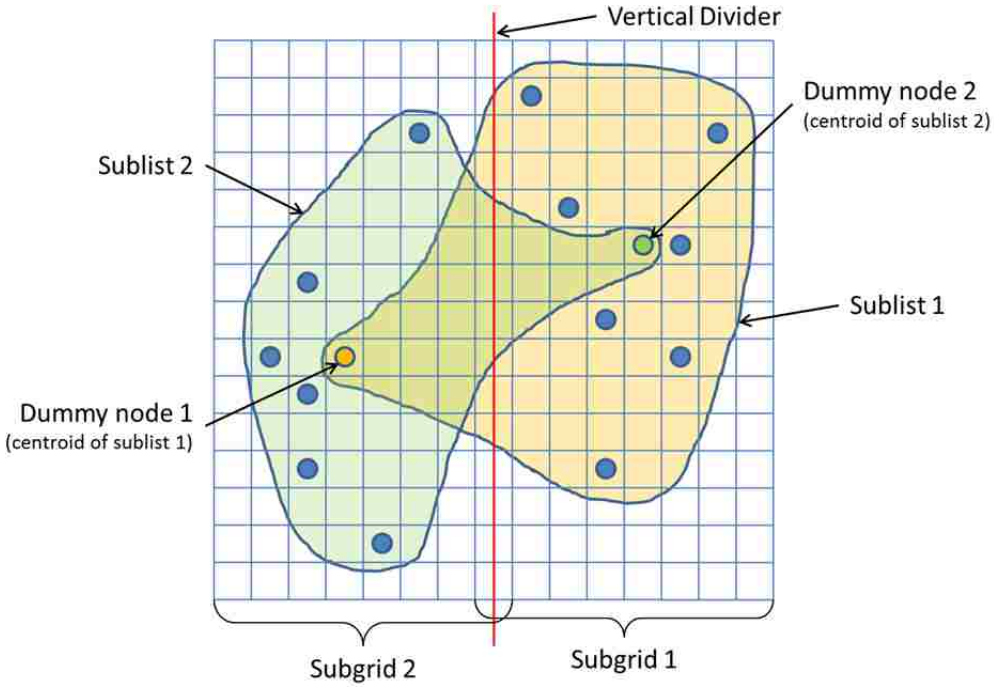

Figure 38: Dividing the grid into two subgrids in order to solve two smaller TSPs

The algorithm first checks if the given node list is small enough and solves the TSP exactly if it is the case. Otherwise, it uses a recursive scheme to construct and solve two subproblems with two node sublists that are attained by dividing the input subgrid in half, either horizontally or vertically in an alternative manner. To join the two subsolutions, a breakpoint is needed to create an opening in each of the subtours and linking them through those gaps. To intelligently determine such a breakpoint, one dummy node representing the first node list is acquired by computing the grid coordinates of the centroid of this cluster ($x_{c1} = \sum_i x_i/n_1$, $y_{c1} = \sum_i y_i/n_1$) and added to the second sublist; the first sublist is also adjusted in a similar fashion. To visualize this, consider Figure 38. These dummy nodes constitute good breakpoints as they provide a good approximation of where a subtour can be geographically extended to encompass another subtour. Eventually, these dummy nodes are removed right before connecting the two TSP subsolutions, as shown in Figure 39. The way the two broken tours are linked together is by

connecting each of their two ends with their counterparts in a least cost fashion. Depending on how many TSP divisions occur during the heuristic's execution, the process of joining subtours is repeated accordingly until a complete TSP solution is acquired.

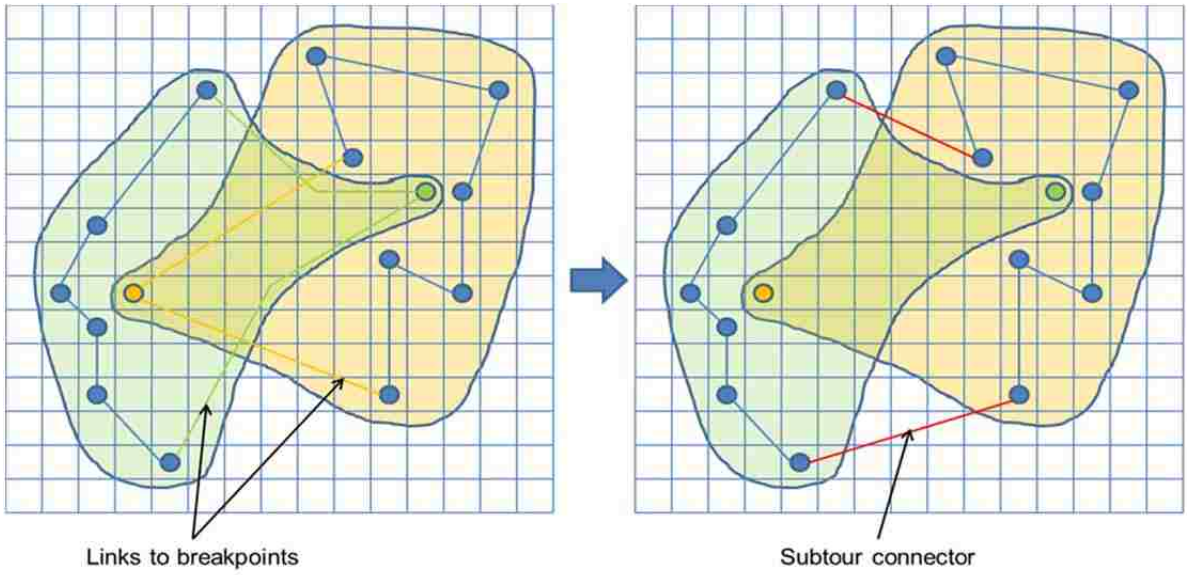

Figure 39: The process of removing the breakpoints and connecting the two subtours

The associated recursive algorithm can therefore be described by the following steps (whose corresponding pseudo-code can be found in Section 2 of the Appendix):

(0) Obtain the input pair, a (sub)grid and its associated hotspots (i.e. node list), either from the passed parameters at the start of the main (tour construction) procedure or from the previously called recursive function.

(1) If the number of nodes is less than a pre-specified threshold (a value of 25 is used here), solve the TSP exactly using the above integer program, and then go to step (7). Else, go to step (2).

(2) Divide the (sub)grid into two equal halves, Subgrid 1 and Subgrid 2, as shown in Figure 38. Subsequently, the list of hotspots is divided into Sublist 1 whose elements reside in Subgrid 1, and Sublist 2 whose elements reside in Subgrid 2. Note that the division ought to be carried out in a vertical fashion if the number of columns in the input grid is larger than the number of its

rows. Otherwise, the divider is to split the structure horizontally. (This scheme eventually results in an alternation between vertical and horizontal cuts in the consecutive recursive calls.)

(3) For each of the two subgrids, find the center of gravity (or centroid) for the corresponding hotspot sublists. Let c1 be the centroid of Sublist 1, and c2 be the centroid of Sublist2; similarly, let n1 be the size of Sublist 1, and n2 be the size of Sublist 2. The coordinates are computed as follows: $(x_{c1}, y_{c1}) = (\sum_i x_i/n_1, \sum_i y_i/n_1)$ and $(x_{c2}, y_{c2}) = (\sum_i x_i/n_2, \sum_i y_i/n_2)$; note that rounding is allowed since a good approximation is sufficient.

(4) Add c1 as a dummy node in Sublist 2, and c2 as a dummy node in Sublist 1, as depicted in Figure 38.

(5) Solve the two resulting TSP subproblems by executing these very steps (0)-(7) on the input pairs (Subgrid 1, Sublist 1) and (Subgrid 2, Sublist 2).

(6) Join the two solutions acquired in step (5); this is achieved by treating the dummy nodes c1 and c2 as breakpoints to create an opening in every subtour before linking them together as demonstrated in Figure 39. Note that these breakpoints are removed completely from the analysis at this point, and that the two broken subtours are combined by connecting their closest ends.

(7) Return the obtained solution to the previously called recursive function, or to the calling main procedure if the recursion is finished.

To evaluate and validate this heuristic, seven experiments are performed on a Dell Latitude D630 with 2GB of RAM and a 1.86GHz Intel® Core™ 2 Duo processor: the TSP is solved for n randomly generated hotspots on a 115x115 grid. Table 6 reveals three aspects that are associated with each experiment: (1) the size of the problem in terms of the number of nodes, variables, and non-integrality constraints (in non-closed form); (2) the performance of the integer program solver CPLEX; and (3) the performance of the developed heuristic along with its optimality gap from the optimal solution. The

results show that the heuristic gives good results as the optimality gap is less than 10%. Moreover, this algorithm proves to be computationally efficient since it runs much faster as the problem size increases and does not greedily deplete memory resources.

Table 6: A comparison of the heuristic against CPLEX in terms of solution quality and runtime

N	Variables	Constraints	Optimal	Runtime	Heuristic	Runtime	Optimality Gap
10	110	92	1356.74	0.02s	1356.74	0.02s	0%
30	930	872	2136.46	5.50s	2283.46	0.32s	6.7%
50	2550	2452	2697.18	5.37min	2724.47	2.61s	1%
75	5700	5552	3078.09	39.14min	3172.72	9.94s	3%
100	10100	9902	3428.14	6.11h	3753.38	15.20s	9.4%
150	22650	22352	OOM	-	4929.58	39.98s	-
300	90300	89702	OOM	-	7189.65	22.671s	-

* OOM ~ Out of memory (by CPLEX)

To conclude, this heuristic is ideal for this study because it is practically fast at yielding a good solution. Furthermore, it enjoys other traits such as being: (1) computationally inexpensive, as it does not require the use of a significant amount of memory to maintain a solution population and perform overhead comparisons); (2) deterministic, so it is more predictable and less prone to the victimization of randomness; (3) not dependent on a specified number of iterations (unlike search algorithms whose solution quality depends on the number of iterations performed); (4) easy to tune as it only has one input parameter (i.e. the threshold value); and (5) a mechanism that easily enforces endogenous constraints—should the need to incorporate constraints arise—as it uses mathematical modeling.

3.2. SPP Grid Reduction

Upon attaining a near-optimal sequence of visits to high-risk cells, the next step is to determine the best path from each node to its subsequent one by solving the (unconstrained) Shortest Path Problem. While this can be achieved by employing well-established efficient algorithms such as Bellman-Ford or Dijkstra, this study prefers solving this problem using an IP formulation because Mathematical Programing allows the enforcement of endogenous constraints, should the need for that arises during

future development or expansion. One example of such a restraint is related to the integration of turn-radii in the UAV's path. In this case, solving the SPP is accomplished via the formulation introduced in Carlyle et al. (2008), which is structured in a way that supports additional linear constraints. In this context, the cost of selecting an edge is equivalent to its geographical approximated length (distance); this helps acquire paths that consume the least amount of energy as well as save the largest amount of time. The unconstrained version of Carlyle et al.'s formulation is as follows, and its associated AMPL code is provided in Section 3 of the Appendix:

V	set of vertices/nodes
E	set of directed edges
c_e	cost of travelling through edge e
b_v	1 if node v is the origin, -1 if it is the destination, and 0 otherwise
a_{ve}	1 if node v is the start of edge e, -1 if v is the end of e, and 0 otherwise
x_e	1 if edge i is selected to be in the path, and 0 otherwise

$$\text{Min} \sum_{\{e \text{ in } E\}} c_e \cdot x_e$$

$$\text{S.t.} \quad \sum_{\{e \text{ in } E\}} a_{ve} \cdot x_e = b_v \qquad \forall v \text{ in } V \qquad (1)$$

$$x_e \text{ in } \{0,1\} \qquad \forall e \text{ in } E \qquad (2)$$

Note that the decision of choosing the IP formulation over efficient SPP algorithms assumes that the practical speed of the developed framework is not compromised. If this assumption does not hold anymore because of a certain problem size or a newly added set of (endogenous) constraints in the SPP model, the Lagrangian-Relaxation-based algorithm developed by Carlyle et al.'s (2008) can be employed to improve computation efficiency.

Regardless of the solution method, an opportunity for improvement in terms of runtime is seized before solving the apt SPPs (one for each pair in the TSP sequence), and that is based on the observation that in a dense grid structure, a significantly large number of edges are not selected to be a part of the shortest path. This fact inspires the idea of engaging only a subset of edges—ones that are more likely to be in the shortest path between two cells—to decrease the amount of required

computation. To elaborate, consider the example in Figure 40, which shows that such trajectory is situated within a *virtual subgrid* where one of the two diagonally opposite corner pairs comprises the origin and destination cells. Notice that given there are no network disconnections that prevent a SPP feasible solution within this subgrid, the shortest trajectory is always contained within this *virtual* structure. If however the origin and destination are somehow disconnected within the subgrid (as depicted in Figure 41), the shortest path then comprises edges from the outside of the virtual structure (assuming there is a feasible solution).

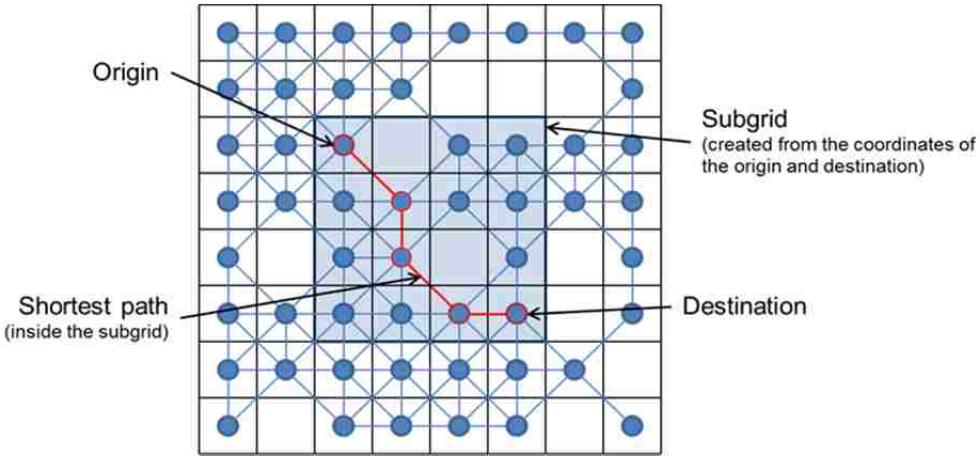

Figure 40: A subgrid comprising the origin, destination, and shortest path

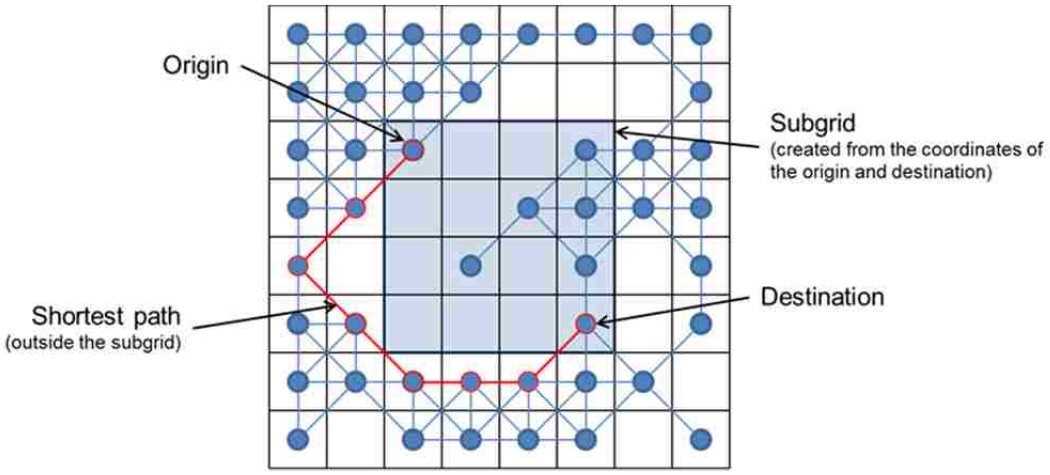

Figure 41: A subgrid comprising the origin and destination, but not the shortest path

To exploit this notion of solving SPP within a smaller scope while mitigating the risk of falling in the situation where no feasible solution exists within the virtual structure, the original NxN grid is

reduced to a subgrid that has some leeway around the contour defined by the origin and destination. To be more specific, if the thickness of this leeway is measured with the number of cells, then such a subgrid is defined by the following bounding rows and columns:

- North boundary: $\max(\min(Row_{origin}, Row_{destination}) - leeway, 0)$
- South boundary: $\min(\max(Row_{origin}, Row_{destination}) + leeway, N-1)$
- West boundary: $\max(\min(Column_{origin}, Column_{destination}) - leeway, 0)$
- East boundary: $\min(\max(Column_{origin}, Column_{destination}) + leeway, N-1)$

Note: in this study, the leeway thickness is set to 5 cells, though it can be reset to any adequate number. Furthermore, if such allowance is not enough, it can be intelligently increased or even maximized to contain the whole grid; this case however is less likely to happen, and it should only be invoked as a last resort.

Using this network reduction method along with an exact method solver to find all the shortest paths associated with the TSP tour proves to be a practically fast mechanism; and in the case of the 50x50 grid example given in Figure 37, test data for multiple hotspots scenarios shows that the use of this reduction technique reduces runtime by half.

4. Altitude Optimization

After effectively and efficiently acquiring a set of 2D surveillance paths for the tactical UAVs, it is important to switch focus to the third dimension—the operating altitude. This does not only reveal how complex the flight might be, but it also enables the computation of the extra energy required for all the altitude changes during the trip, and hence, leading to a more accurate verification of the endurance constraint (as discussed in the next section). Consequently, the UAV's operating altitude associated with each cell in a specific target path is resolved subject to the following set of endogenous constraints:

1. The **minimum/maximum operating altitude constraints**, which ensure that the UAV does not fly below or above its lower or upper altitude limits, respectively. Note however that it is important to distinguish between *above ground level* (AGL) altitudes and *above (mean) seal level* (ASL/MSL) altitudes. Usually, the given data for minOA's is given in AGL whereas its maxOA counterpart is given in ASL (as shown in Figure 13), and this is the standard followed in this study.
2. The **climb-/dive-rate constraint**, which helps verify that the UAV does not perform a change of altitude that exceed its climb and dive capabilities when moving from one cell to another.
3. The **stealth constraint**, which ensures that the UAV flies above its dB-distance, the distance that reveals its presence through its noise. Note that the dB-distance parameter is an AGL altitude. An important aspect about this constraint is that unlike its aforementioned counterparts, its violation does not physically handicap or forbid the UAV's flight. Furthermore, it would not be appropriate to refuse a good solution just because this constraint is violated by a small degree. As a result, this study treats it as a *soft constraint* (whose violation does not lead to infeasibility), and this is achieved by introducing an allowance (a.k.a. deficiency or elasticity) variable that would be minimized with a certain importance or priority in the objective function.
4. The **sensing distance constraint**, which verifies that the UAV is flying at an altitude that enables its onboard sensors to obtain a high image quality, at least at the level of vehicle detection. However, this restriction is also treated in this study as a soft constraint as it is not recommended to disregard a solution just because the image quality at a certain cell is a little below the required threshold. This becomes even more important after realizing that the struggle to satisfy this constraint might be an objective that conflicts with satisfying the stealth constraint. To clarify this, maximizing image resolution drives the UAV to be closer to the ground while maximizing stealth pushes the aerial asset away from the earth surface

As for the objectives, the below three measures are identified:

- According to discussions with experts in the area, one of the factors involved in determining the appropriate operating altitude for a UAV is minimizing the change in ASL altitudes. This not only helps keep the endurance consumption to a minimum, but it also simplifies the task of the controller during the surveillance shift as s/he does not have to keep up with all the flight climbs and dives. As for the endurance consumption itself (which is measured in time), it can be easily computed outside of the developed optimization model without affecting the accuracy of the analysis because this measure is positively proportional to ASL altitude changes.

- The second objective is a result of treating the sensing distance restriction as a soft constraint. In other words, the image resolution becomes part of the optimization model's objective function. As introduced in Chapter 3, the formula for computing the image resolution acquired by a specific UAV is as follows:

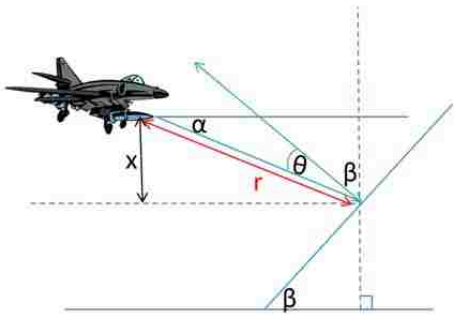

Figure 42: The geometric situation at a given cell

$$R_i = (1000 \cdot s/r) \cdot \cos(\theta_i) \cdot (1 - f_j/3)$$

Where: - R_i is the acquired image resolution at cell i

- s is the sensitivity (power) of the UAV's sensor

- r is the slant range of the UAV's sensor

- θ_i is the angle between the slant direction of the UAV's sensor and the normal line on the surface of cell i, as shown in Figure 42

- f_i is the foliage degree (an integer score that ranges from 0 to 3) of cell i

Let x_i be the AGL operating altitude of the UAV at cell i, let β_i be the slope angle at cell i, and let α be the tilt angle of the UAV's sensor, as demonstrated in Figure 42. Hence, two geometric expressions can be derived: $\sin(\alpha) = x_i/r$ and $\theta_i = \pi - \alpha - \beta_i$. For simplicity purposes during the optimization process, assume that the terrain at cell i is flat, i.e. $\beta_i = 0$; this way, x_i can be kept constant over cell i. Thus, the image resolution formula becomes as follows:

$$R_i = (1000 \,.\, s \,.\, \sin(\alpha)/x_i) \,.\, \cos(\pi - \alpha) \,.\, (1 - f_i/3)$$

This mathematical expression however renders it cumbersome to utilize in a MP formulation as it is nonlinear and involves a number of parameters. To ease this matter while still pursuing the same goal, this part of the objective function is set to x_{max} such that $x_{max} \geq x_i$ for all i in arg{path nodes} since this still optimizes the operating altitude to maximize the image quality at each node (which is inversely proportional to x_i). Furthermore, x_{max} yields the same measurement unit as the other objectives, eliminating by that the need for conversion and calibration.

- Similarly, stealth is treated as a soft constraint, which leads to the third objective function that tries to achieve the minimal allowable violation through the sum of all allowances across all traversed nodes.

The result of such a discussion is the following linear program (referred to as OAP, and whose AMPL code can be found in Section 4 of the Appendix):

V	set of cells (i.e. vertices/nodes) in UAV's path
E	set of edges in UAV's path (including links)
e_i	elevation (above sea level) of cell i
d_{ij}	distance between cell i and cell j (center to center)
r_c	UAV's maximum climb-rate
r_d	UAV's maximum dive-rate
a_L	UAV's operating altitude lower bound (above ground level)
a_U	UAV's operating altitude upper bound (above sea level)
a_{dB}	UAV's dB-distance (above ground level)
w_δ	objective weight associated with change in operating altitude

w_x objective weight associated with operating altitude (i.e. sensing distance)
w_z objective weight associated with stealth

x_i UAV's operating altitude (above ground level) at cell i
x_{max} UAV's maximum operating altitude across all nodes (above ground level)
y_i UAV's operating altitude (above sea level) at cell i
δ_{max} UAV's maximum operating altitude change across all edges
z_i allowance for violating stealth constraint at cell i

$$\text{Min } w_\delta \cdot \delta_{max} + w_x \cdot x_{max} + w_z \cdot \sum_{\{i \text{ in } V\}} z_i$$

S.t.	$y_i = x_i + e_i$	$\forall i$ in V	(1)
	$x_i \geq a_L$	$\forall i$ in V	(2)
	$y_i \leq a_U$	$\forall i$ in V	(3)
	$y_i - y_j \leq \min(r_c, r_d) \cdot d_{ij}$	$\forall (i,j)$ in E	(4)
	$y_j - y_i \leq \min(r_c, r_d) \cdot d_{ij}$	$\forall (i,j)$ in E	(5)
	$y_i - y_j \leq \delta_{max}$	$\forall (i,j)$ in E	(6)
	$y_j - y_i \leq \delta_{max}$	$\forall (i,j)$ in E	(7)
	$x_i + z_i \geq a_{dB}$	$\forall i$ in V	(8)
	$x_i \leq x_{max}$	$\forall i$ in V	(9)
	$x_i \geq 0, y_i \geq 0, z_i \geq 0$	$\forall i$ in V	(10)
	$x_{max} \geq 0, \delta_{max} \geq 0$		(11)

Constraint (1) establishes a relationship between the corresponding x_i and y_i variables. Constraints (2) and (3) force the sought operating altitude to be within the UAV's limits. Constraints (4) and (5) ensure that an edge (i,j) can be crossed in both directions without exceeding the vehicle's diving and climbing rate limits. Constraints (6) and (7) together force the variable δ_{max} to carry the largest value of all the changes in operating altitudes. Constraint (8) establishes a relationship between the corresponding x_i and z_i variables. Constraint (9) ensures that x_{max} holds the highest value across all x_i variables (i.e. the maximum AGL operating altitude found so far). Lastly, constraints (10) and (11) are to guarantee that all involved variables are nonnegative. As for the multi-objective function, it comprises a weighted summation of three main terms that are associated with the three aspects previously discussed: change in altitude, sensing distance, and stealth. Note: these weights sum up to 1 (i.e. $w_\delta + w_x + w_z = 1$).

Integrating weight multipliers (i.e. w_δ, w_x, and w_z) in the three-objective function and treating them as input parameters gives the surveillance architect flexibility in redefining his/her priorities with regards to determining the right operating altitude. For instance, the analyst may regard the captured image resolution as all that matters; in this case, s/he can assign the following values: (w_δ, w_x, w_z) = (0, 1, 0). It is noteworthy however that the user of this developed decision tool should pay great attention to the values assigned to these weight parameters as they can lead to substantially different results. To illustrate this, consider the small example depicted in Figure 43; this instance has four nodes (0, 1, 2, and 3) and three edges ((0,1), (0,2), and (0,3)), a scenario that specifically mimics the situation of a hotspot as such a cell is usually connected to two tour paths and one link. Cells 0, 1, 2, and 3 have an elevation of 450m, 500m, 400m, and 400m, respectively, whereas edges ((0,1), (0,2), and (0,3) have a length of 50m, 50m, and 70m, respectively. Moreover, it is assumed that the employed UAV has a climb-rate and dive-rate that are equal to 1, a dB-distance of 100m, and OA limits of 50m and 1500m (Note: these input data are coded in AMPL and can be found in Section 4 of the Appendix). Running the OAP model with different objective preference configurations gives the results in Table 7.

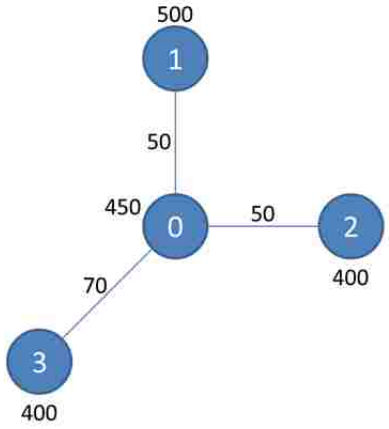

Figure 43: Example mimicking a hotspot

To sum up, the first scenario favors a constant ASL operating altitude with no allowance. The second preference yields variations in both the ASL and AGL altitudes with some stealth allowance. The final configuration leads to a constant AGL operating altitude, a high allowance across all nodes, and great variations in ASL altitudes. Because of this sensitivity to the objective weights, the surveillance architect is advised to define the objective priorities well and evaluate few scenarios to internalize the impact of his/her decisions.

Table 7: Results of running the OAP model on the input data of the above example

Preference	Node i	x_i	y_i	z_i	Total Objective
$w_\delta = 0.5$ $w_x = 0.2$ $w_z = 0.3$	0 1 2 3	150 100 200 200	600 600 600 600	0 0 0 0	40
$w_\delta = 0.5$ $w_x = 0.4$ $w_z = 0.1$	0 1 2 3	75 50 100 100	525 550 500 500	25 50 0 0	60
$w_\delta = 0.4$; $w_x = 0.5$; $w_z = 0.1$;	0 1 2 3	50 50 50 50	500 550 450 450	50 50 50 50	65

5. Tour Segmentation

In the previous section, it is shown that a surveillance tour is algorithmically designed to visit all reachable hotspots, and its associated trip cost is determined in terms of distance traveled. Furthermore, efficient links from the outpost to the high-risk cells (as shown in Figure 37) as well as their associated energy consumption are also established. The only missing piece now is the verification of the endurance constraint for the whole surveillance trip, as the traveling UAV might not be able to traverse all the tour's segments plus two complete links (for going back and forth from the outpost) with one charged battery or fuel tank.

To address such an issue, a *finishing* algorithm is executed in order to reach one of the following states:

1. Confirm that there exists at least one tour that starts from the outpost and feasibly visits all hotspots before returning to the origin with enough energy reserve. (Note that if there is more than one of such tours—depending on which two links are chosen—the least-cost trip is selected.)

2. Segment the tour into a set of subtours that yield a feasible round trip. Note that this set has the same cardinality as the set of hotspots to be visited. In other words, every high-risk cell is adopted as the start of a unique subtour (given that the UAV has just left the outpost), which is then stored as an option for the later-solved Set Covering Problem. This helps construct an efficient combination of subtours performed by possibly different UAVs in order to optimize certain objective measures, which are later discussed in the upcoming chapter.

This finishing algorithm can be described by the following steps (whose corresponding pseudo-code can be found in Section 5 of the Appendix). Furthermore, Figure 44 is presented to illustrate the generated subtour's state during a certain iteration.

(0) Obtain the list of hotspots to be visited in a sorted state according to the TSP sequence, and establish an empty list of subtours that is returned as an output.

(1) If the hotspots list is empty, then go to step (6). Otherwise, select the cell that is on top of the hotspots list, call it the *start-node*, and proceed to step (2).

(2) Establish a new subtour that has two segments only: a link from the outpost to the start-note, and another link from the start-node to the outpost. Note that this subtour is feasible endurance-wise because the start-node has passed the reachability test (as discussed in Section 2 of this chapter).

(3) Perform an iterative loop of the following substeps until a break is reached:

   a. Refer to the TSP sequence that starts with the start-node.

   b. Obtain the next hotspot on the list that has not been included in the established subtour. If all hotspots are included in the subtour already, break out of the loop.

   c. Compute the trip distance that would be traveled by the UAV if this next hotspot were to be added to the subtour. This is accomplished by acquiring the total length of the subtour (which includes the links), subtracting the length of the recently-added link (also referred to

as the *current return path*), and then adding the length of the new link (from the next hotspot to the outpost) and the length of the new path (that connects the next hotspot to the current subtour). To visualize this situation, see Figure 44 where the solid lines portray the subtour's trajectory while the dotted lines represent the yet-to-be included paths.

d. If this computed distance does not violate the endurance constraints, then discard the current return path and add to the subtour the new path (to the next hotspot) and its associated link. Otherwise, break out of the loop.

(4) Add the subtour to the subtours list.

(5) Remove the start-node from the original list of hotspots, then go to step (1).

(6) If the subtours list contains at least one trip that comprises all hotspots, then return the least-cost trip alone. Otherwise, return the entire subtours list. Lastly, stop this procedure.

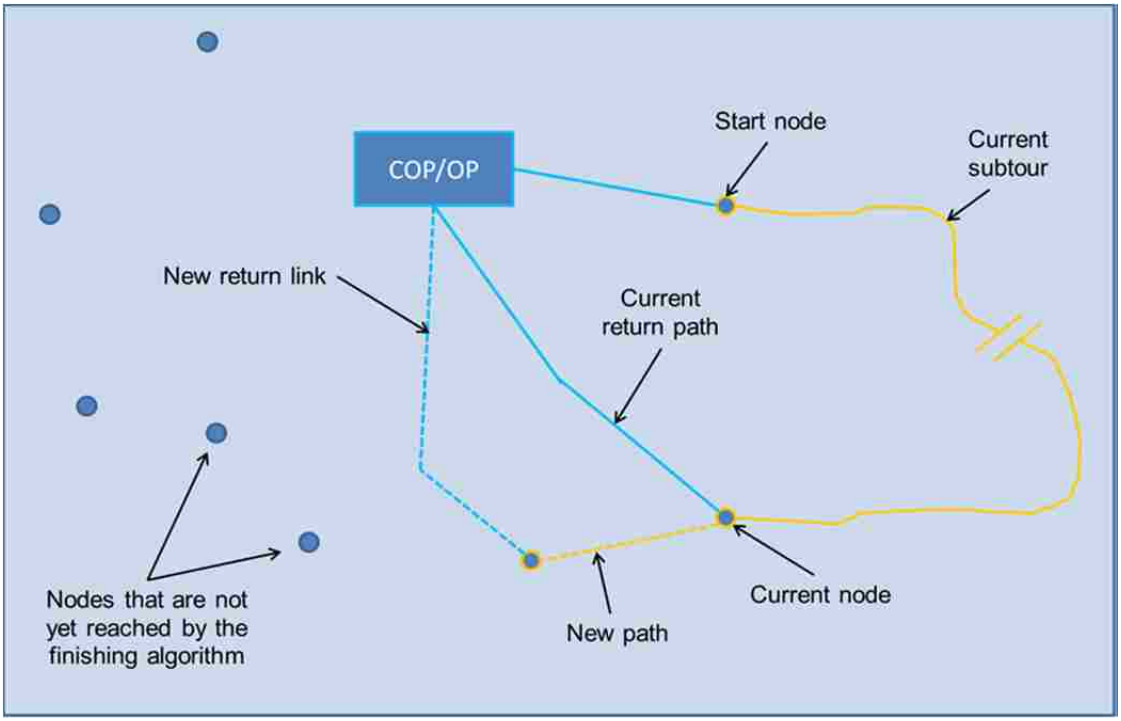

Figure 44: A snapshot during the process of expanding a subtour by the finishing algorithm

Note that each subtour ought to be associated with an approximated cycle-time in order to facilitate the resource allocation process (discussed in the next chapter). This parameter is obtained by

summing up three distinct periods: (1) the downtime, which is spent readying the device through performing maintenance and refueling the tank or replacing the battery; (2) the travel time, which is the temporal interval required to traverse all edges in the subtour; and (3) the inspection time needed to scan every hotspot (note: the scan of a specific high-risk cell i by a an aerial vehicle u is assumed to be performed in a serpentine flight pattern with a FOV of a diameter d(u,i); the computation process of the traveled distance is demonstrated in Figure 45).

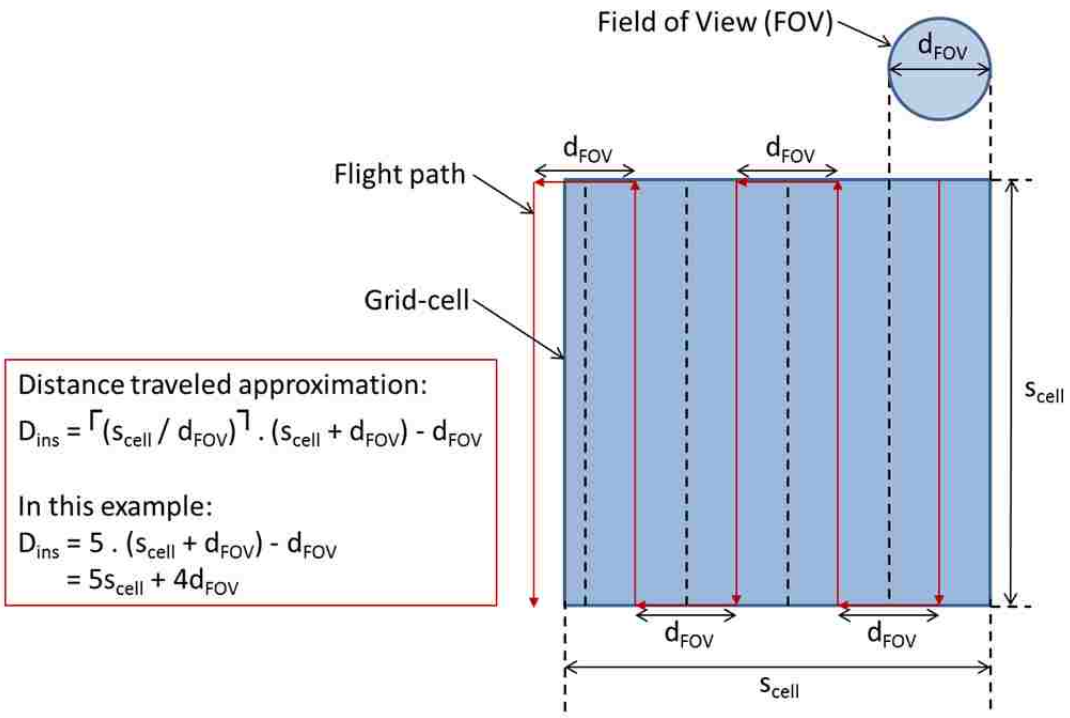

Figure 45: The computation process of the traveled distance for inspection (D_{ins})

Therefore, the following formula is utilized to compute the cycle time of a UAV u:

$$T_c(u) = t_D(u) + \sum_{\{e \text{ in edges}\}} [L(e) / v(u)]$$
$$+ \sum_{\{i \text{ in hotposts}\}} [p_{scan} . [\lceil (s_{cell} / d(u,i)) \rceil . (s_{cell} + d(u,i)) - d(u,i)] / v(u)]$$

where v(u) and $t_D(u)$ are the cruise velocity and downtime associated with UAV u, respectively. Furthermore, L(e) is the length of edge e, and s_{cell} is the cell-side's length. Finally, consider p_{scan}, the proportion of the scanning effort, which is a measure introduced to provide flexibility in controlling the

time spent inspecting. This parameter can also be utilized to adjust the time spent in inspection if a non-serpentine flight pattern is assumed. Note that the field of view diameter d(u,i) for a hotpot i that is scanned by a UAV u is obtained through this formula:

$$d(u,i) = 2 \cdot OA(u,i) \cdot r_{DV}(u) / r(u)$$

where OA(u,i) is the optimized operating altitude at which vehicle u flies over cell i, a measure acquired through the method introduced in the previous section. Moreover, r(u) and $r_{DV}(u)$ are the slant-range and the FOV radius for vehicle detection of the sensor carried onboard UAV u, respectively.

To conclude, this chapter explains how endurance-compliant 3D subtours are generated for each tactical system. These subtours represent all the possible options to carry out the desired surveillance in a feasible manner. The next chapter introduces a method to select the most efficient portfolio of these options in order to define the final set of assets to be purchased, and hence, provide a practical and cogent answer to the target investment question of this study.

CHAPTER 6: SYSTEM ALLOCATION

In this chapter, resource allocation is performed in order to build a complete surveillance plan and extract from it the manning and technological investments required to provide adequate protection to a specific outpost. In the context of this study, investment decisions are reliant on a comprehensive investigation that spans several attributes, factors, operations, and technological systems. So far, most of the analysis has been centered around the use of tactical UAVs as their way of operating is by far the most constrained, and is subsequently critical and necessary prior to making a decision. Nonetheless, these vehicles are not the only popular systems in the market; strategic aerial vehicles and stationary assets also play a role in providing advantages through their surveillance sensors. The procedure adopted to achieve an investment choice varies significantly between these three kinds of technologies due to several features.

First, tactical systems fly at low altitudes and have an accurate but small swath; this renders them adequate for monitoring routes and trails as well as checking geographical spots and small regions. In addition, they are vulnerable to obstructions and high-wind levels. As a result, their potential flight tours have to be evaluated. Strategic vehicles on the other hand operate from high altitudes. They are much less vulnerable to obstructions and weather conditions, and their swath can cover substantially large areas; as a result, their surveillance plan can be modeled as a surface area that is scanned with scheduled flight patterns, as discussed in Section 3 of this chapter. Conversely, immobile assets provide a stare-mechanism that is persistence over time, and neither motion patterns nor revisit-times are needed to be analyzed. However, the number of required stationary assets may be large, and even difficult to determine, as the sensors' field of view might be much smaller than the area to be covered. Furthermore, deploying several of them to achieve an adequate coverage depends on the ability to

protect them (and hence, their required proximity to the base) as well as the availability of anchored platforms and installation surfaces.

Another feature that causes a difference in technology evaluation is the refueling need. While stationary systems are not subject to this constraint, mobile aircrafts have different ways of recovering their endurance. To be more specific, tactical UAVs are organic (i.e. dedicated) to a specific outpost and can be easily recovered and refueled/recharged upon arriving to their origin. Strategic assets on the other hand must return to the main base (FOB), which ought to have a runway and is located tens, if not hundreds, of kilometers away from the surveilled outpost. This leads to two important aspects, first of which is associated with the fact that strategic systems do not compete for human resources with their tactical and stationary counterparts. Furthermore, strategic aerial vehicles belong to the main base, but they do surveil large areas surrounding several outposts. Therefore, such an asset can be considered a shared resource, and this requires that several outposts are to be considered in the corresponding resource allocation model in order to reach a valid investment conclusion.

To conclude, these main three types of technological assets should be treated separately and differently in order to reach a valid and practical investment decision. The various factors behind such an argument are summarized in Table 8. Nevertheless, regardless of the type of analysis involved, there is a serious limitation that has to be considered for outpost surveillance missions, and that is the lack of human resources during a certain operation period. This is due to the outpost's nature of being small and to the level of expertise the systems operator has to have. The manning requirement to carry out such missions is one of the main concerns during planning, and is therefore treated as the objective of the optimization model. The associated minimization is however performed in an implicit and natural fashion through optimizing the number of utilized assets per kind, as explained in the later sections.

Table 8: The factors involved in determining the appropriate analysis for a certain type of assets

	Tactical	**Stationary**	**Strategic**
Operating altitude	Low	Low	High
Modeling structure	Tour	Surface area	Surface area
Ownership type	Organic	Organic	Shared
Refueling	Needed	Not Needed	Needed
Service type	Intermittent	Persistent	Intermittent
Asset protection	Need for stealth	Need for proximity to outpost	Not needed
Control personnel	Needed	Not needed	Needed

1. Tactical Systems

1.1. Building the Solution

At this stage, information about the UAVs' subtours is available as explained in Chapter 5. For the sake of easy clarification, consider that the data is organized in a column format of 0's and 1's so as to facilitate its integration to an optimization model. In this matrix, the element a_{ij} equals to 1 when subtour j visits node i, and 0 otherwise. If this numerical structure is fed to the classic Set Covering Problem (see the model below, from Vaziran (2001, p. 108)), one can obtain a least-cost solution—in terms of traveled distance—that covers all the given hotspots.

S	set of subsets
E	set of elements
c_s	cost of selecting subset s
a_{es}	1 if subset s contains element e, and 0 otherwise
x_s	1 if subset s is selected, and 0 otherwise

$$\text{Min } \sum_{\{s \text{ in } S\}} c_s \cdot x_s$$

$$\text{S.t.} \quad \sum_{\{s \text{ in } S\}} a_{es} \cdot x_s = 1 \qquad \forall e \text{ in } E \qquad (1)$$

$$x_s \text{ in } \{0,1\} \qquad \forall s \text{ in } S \qquad (2)$$

The advantage of this model is that it can incorporate several system brands (e.g. Raven, Shadow, etc) in the solution at once, as opposed to solving the UAV-subtour allocation problem one brand at a time. To be more precise, each kind of vehicles is associated with a submatrix that comprises

every way in which the asset can be employed. An example of five hotspots and three UAVs is depicted in Figure 46.

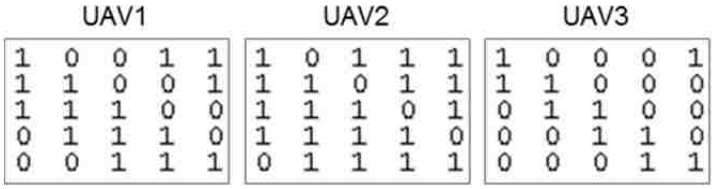

Figure 46: Subtour input data to the Set Covering Problem

This scenario assumes that none of the UAVs has enough endurance to complete the entire tour. As a result, there is at least one 0 in every column of the matrix. Also, notice that there are as many columns (subtours) as hotspots for each vehicle, and that this number coincides with the number of options of starting a subtour at a certain node.

The classic SCP model is however not directly applicable in the context of this study for two main reasons. First, there is a need to consider the time aspect when choosing subtours and assigning UAVs. In other words, the revisit-frequency has to be adequate enough for each hotspot to mitigate the risk associated with it. The revisit-time constraint can then be integrated in this phase to ensure that τ_i (the revisit-time of hotspot i, which is introduced in Chapter 4) is met for all high-risk cells. This issue becomes even more complex when the aspect of parallelism is employed. That is, there might be the case of being unable to satisfy this constraint for a certain subtour with one UAV. In this case, two or more vehicles have to be operated concurrently with an adequate timeframe between their departure times. As a result, the SCP is converted to a *Time-Based Set Covering Problem* (TBSCP), which takes advantage of the following formula that reflects concurrency:

$$1/T_{total} = \sum_i (1/T_i)$$

where T_{total} is the cycle-time that results from operating several systems concurrently, with each system i having a cycle-time of T_i (note that the cycle-time in this context spans the flight period, the refuel or

recharge time, and any periodic maintenance; as a result, the scheduling and refueling need constraints are integrated just by using this formula). To illustrate this, consider two UAVs with a cycle-time of 1 hour; if flown simultaneously and with an appropriate time gap between departure times, a hotspot can be visited every 30 minutes (i.e. 1/(1/2) = 1 + 1). This leads to the second point of modifying the SCP, which corresponds to the potential need to utilize two or more aerial vehicles of the same brand. This can be accounted for by converting the decision binary variables to integers. The customized TBSCP can then be formulated as follows, and its associated AMPL code is provided in Section 6 of the Appendix:

S	set of (sub)tours
V	set of nodes
c_j	cost of selecting (sub)tour j
f_{ij}	revisit-frequency (by a certain UAV brand) if (sub)tour j contains node i, and 0 otherwise
τ_i	required revisit-time for node i
x_j	needed number of trips for (sub)tour j

$$\text{Min} \sum_{\{j \text{ in } S\}} c_j \cdot x_j$$

$$\text{S.t.} \quad \sum_{\{j \text{ in } S\}} f_{ij} \cdot x_j = 1/\tau_i \qquad \forall i \text{ in } V \qquad (1)$$

$$x_j \text{ in } \{0,1\} \qquad \forall j \text{ in } S \qquad (2)$$

1.2. Refining the Solution

The output solution of the TBSCP optimization model is a conservative one and can sometimes be costly, as it assumes that the subtours have to be monitored by a set of tactical platforms and their operators concurrently, which might be unnecessary. In order to avoid superfluous investments and resource allocation, in both systems and personnel, an algorithm can be executed to verify if independent subtours can be travelled in a sequential fashion; this becomes even more beneficial when these subtours do not contain hotspots with short revisit-times. To illustrate this point, consider the example depicted in Figure 47; the optimal solution given by the TBSCP model requires four UAVs to operate in parallel: one vehicle for subtour1, one vehicle for subtour2, and two vehicles for subtour3. Also, notice that subtour3 contains a hotspot (in red) that requires a shorter revisit-time due to its

extreme high risk-score; this is why this subtour demands the concurrent operations of two UAVs, and as a result, it is left alone from further analysis. On the other hand, subtour1 and subtour2 may be able to be operated by the same UAV if the sum of the associated cycle-times (which include recharging/refueling and maintenance) is less than the minimum revisit-time across all the nodes in these two subtours; the corresponding formula models such a condition:

$$T_{subtour1} + T_{subtour2} < \min_{\{i \text{ in subtour1 U subtour2}\}}(\tau_i)$$

If this condition is met, then the number of vehicles in the solution can be decreased from 4 to 3. The issue with checking this condition is that it becomes cumbersome when there are several subtours that can be joined. In fact, this turns out to be an optimization problem that targets assigning assets to subtours in a way that minimizes investment costs while still satisfying the revisit-frequency constraint.

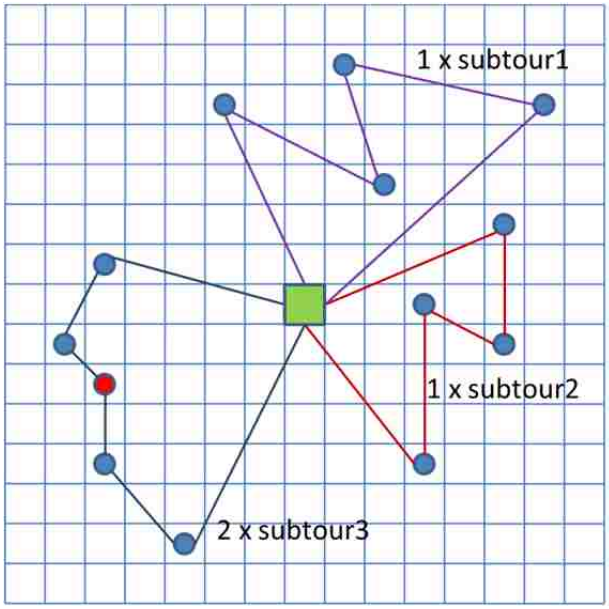

Figure 47: A TBSCP solution in which one hotspot (in red) requires a high revisit-frequency

This can be achieved through a variant of the *Assignment Problem* (AP), and sets of subtours can be formed to be serviced by one dedicated UAV. Similarly to TBSCP, this assignment model needs to satisfy the revisit-frequency constraint, and hence, requires the integration of the time aspect. This is accomplished via the following model, which solves the problem referred to in this context as the *Time-*

Based Vehicle Assignment Problem (TBVAP), and whose AMPL code can be found in in Section 7 of the Appendix:

S	set of subtours
V	set of vehicles
m	a very large number (a.k.a. big-m)
d_{ij}	duration that vehicle i takes to traverse subtour j
τ_i	required revisit-time for subtour j
x_{ij}	1 if vehicle i services subtour j, and 0 otherwise
y_i	1 if vehicle i is to be purchased, and 0 otherwise

$$\text{Min} \sum_{\{i \text{ in } V\}} y_i$$

S.t.	$\sum_{\{i \text{ in } V\}} x_{ij} = 1$	$\forall j \text{ in } S$	(1)
	$\sum_{\{j \text{ in } S\}} d_{ij} \cdot x_{ij} \leq \min_{\{j \text{ in } S\}} \tau_j$	$\forall i \text{ in } V$	(2)
	$\sum_{\{j \text{ in } S\}} x_{ij} \leq m \cdot y_i$	$\forall i \text{ in } V$	(3)
	$x_{ij} \text{ in } \{0,1\}$	$\forall i \text{ in } V, \forall j \text{ in } S$	(4)
	$y_i \text{ in } \{0,1\}$	$\forall i \text{ in } V$	(5)

Constraint (1) ensures that every subtour is serviced by one vehicle, whereas constraint (2) forces the required revisit-time to be met. Constraint (3) establishes the relationship between x_{ij} and y_i variables in such a way that if a certain UAV is not to be purchased, all the assignments associated with it are void. The objective function targets the reduction of purchased assets. In this model, it is important to note that:

- The revisit-time of a subtour is the minimum revisit-time across all the nodes in that subtour.
- Not all subtours qualify for such an analysis; they ought to have a revisit-time that is high enough to be monitored by one UAV at a time.
- When a vehicle cannot service a subtour, it is assigned a traversal duration of m (big-m, a very large number) in order to force the resource to be an unattractive alternative for the model to select.

- Swapping or reassigning UAVs of different brands to subtours (e.g. replacing UAV i of brand Shadow with UAV k of brand ScanEagle for subtour j) in this model is always a safe transaction as constraint (2) ensures that the newly assigned platform has enough endurance to service the subtour.

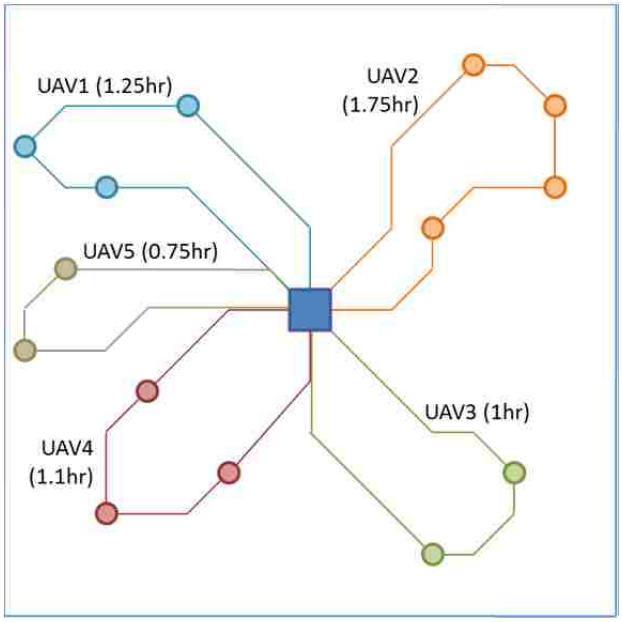

Figure 48: An example of five subtours that are assigned five UAVs

To explain how this works through a concrete example, consider Figure 48. It depicts a TBSCP solution where five subtours are assigned five vehicles whereby UAV1, UAV2, UAV3, UAV4, and UAV5 service their corresponding subtours in 1.25, 1.75, 1, 1.1, 0.75 hours, respectively. Again, the objective is to decrease the number of resources to monitor the five paths if possible. Note that for simplicity, all vehicles are assumed to have the same cruise velocity; and hence, the duration they spend to service a certain subtour is the same. Furthermore, assume that all paths have the same revisit-time—2.5 hours in this case. The AMPL code for this example's input data can be found in Section 7 of the Appendix.

The outcome of this instance is a solution that suggests employing three technological assets rather than the original five, yielding a 40% reduction in the number of assets needed. Moreover, a sensitivity analysis on the revisit-time demonstrates this input parameter can significantly impact the

output solution. In this example, a revisit-time ranging from 1.95 hours to 6 hours leads to an improvement from 20% to 80%, as shown in Table 9.

Table 9: Sensitivity analysis on the revisit-time with the reduction of needed assets as an outcome

Revisit-time (hrs)	Assets to be purchased	Reduction
1.95	4	20%
2.5	3	40%
3	2	60%
6	1	80%

In conclusion, depending on the risk degree of the subtours' nodes, the TBVAP optimization model can greatly refine the tactical system allocation solution (by TBSCP) to a less costly one. However, it is possible for this resource allocation mechanism to propose a solution that can be impractical with respect to the needed personnel (e.g. ten operators to control five Shadows in an outpost of twenty soldiers). For that, the option of reducing the manning requirement and its impact on the surveillance plan are to be investigated. To be more specific, because less personnel means the ability to control less assets, the process of flying aircrafts concurrently might have to become sequential at some degree; this leads the revisit-times to be higher. This tradeoff has therefore to be further studied and analyzed; this can be accomplished through repetitive sensitivity analysis, or alternatively, by altering the model to a surrogate that minimizes the revisit-time subject to an asset purchase constraint. All in all, the resource allocation solution that is obtained through this phase has to be evaluated to ensure practicality.

1.3. Scaling the Solution

The last remaining piece to attain a complete surveillance plan for tactical systems is to scale the solution constructed in the previous section to a 24-hour schedule. This is achieved by simply repeating the subtour cycle for each vehicle throughout the entire day. The cycle-time can also be increased with more idle time in case the required revisit-frequency is far from being reached. For instance, consider the case in which the Shadow can perform a subtour cycle in 1.5 hour while the minimum revisit-time is

2.5 hours. The cycle-time can be adjusted to 2 hours in order to minimize flight time that is deemed unnecessary. Another adjustment that can be introduced during this final phase is customizing the cycle-times and flight times to the personnel's availability as well as to the weather conditions. As an example, the outpost might find night surveillance in the farther whereabouts not as important if the rough terrain is believed to be highly unlikely to be crossed by insurgents during dark hours.

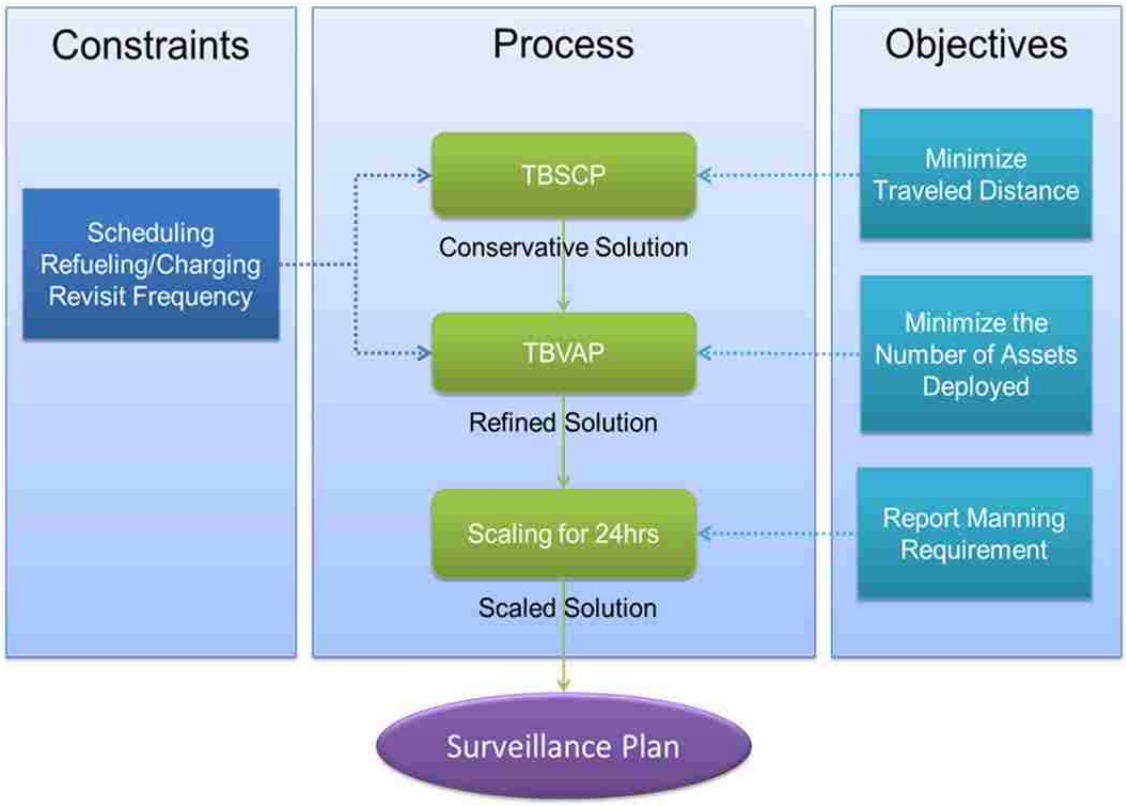

Figure 49: Process of building the surveillance plan for tactical systems

In this study however, it is assumed that the whole 24-hour range is covered since the aim is to find a good approximation for the required investment and resource allocation, as opposed to planning what the outpost ought to exactly do. Given that, the manning requirement stays constant during the day, and the measure is reported as an integer that reflects the required availability at each point of time (e.g. 3 people means that at a certain time during the day, there has to be 3 personnel on duty, as

opposed to 3 people per day). This way, flexibility is given to the decision maker to decide on the actual number of people to be in the outpost.

All in all, this section presents another piece of the developed framework that incorporates the constraints associated with scheduling, revisit-frequency, and refueling or charging the battery, for the tactical systems. This can be summarized in three main steps, which are depicted in Figure 49.

2. Stationary Systems

Stationary systems, mainly tower/wall cameras and aerostats, offer a convenient technological means to monitor whereabouts with very short or no revisit time. Nonetheless, because of their immobility, there are two aspects that have to be considered. Firstly, these assets have to be somehow protected because of their inability of fleeing an attack scene. Subsequently, they have to be close to the outpost if not inside it. Secondly, a digitization grid system is generally inappropriate for modeling static surveillance coverage. This is due to the undesirable dependency on the cell size. To be more specific, in such a paradigm, cells are assigned technological assets; while mobile platforms can flexibly fly in certain patterns to cover part or all of the cell area, immobile systems are fixed to a specific location, and their ability to cover this surface depends on the size of the grid cell. To further illustrate this inconvenience, consider the example in Figure 50 where the digitization of the map results in four cells representing the outpost. The first issue with this scenario is that the sensor range (assumed to be circular) only covers a small portion of the cell. Assigning one sensor per cell would create major vulnerabilities in surveilling the vicinity. Another problem is that the exact location of the sensor cannot be determined because the details of the shape and borders of the COP/OP are hidden behind the cell representation. Furthermore, this inconvenience and its resulting inaccuracy become more extreme as the cell size becomes larger.

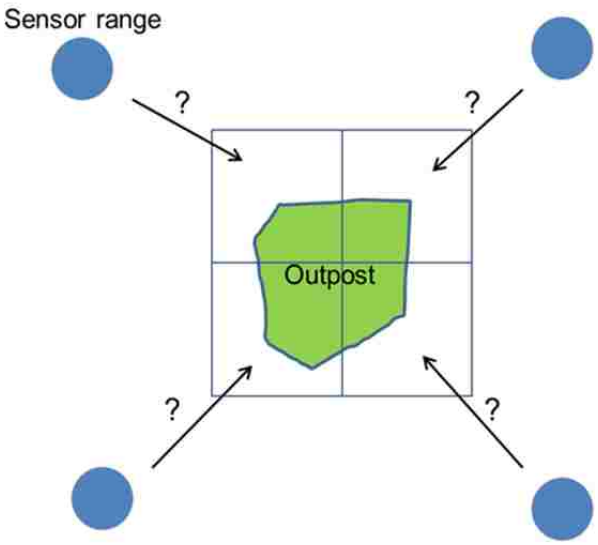

Figure 50: A digitized map of an outpost where each grid cell is assigned a stationary sensor

This study therefore suggests another strategy that is independent of the grid structure, yet still viewable in a map visualization interface. This idea is based on modeling the needed surveillance coverage as a thick *virtual wall* (also referred to as the *surveillance tape*) around the outpost that acts as an alarm zone between the inside and outside.

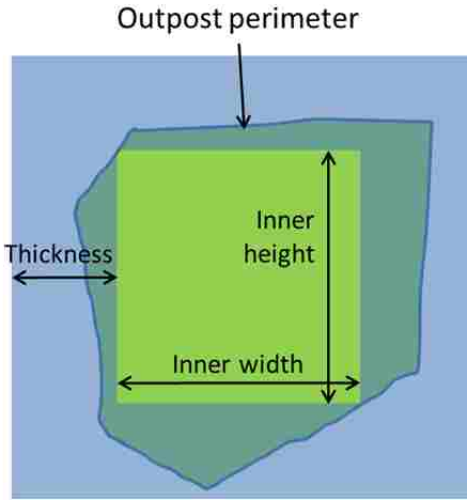

Figure 51: A way to envelop the irregular perimeter of the outpost with a surveillance tape

One of the benefits of using this structure is the elimination of the need to assume that COPs/OPs do not have irregular shapes, a typical case when space is occupied by a human settlement. Another advantage is the inherited satisfaction of *the asset protection constraint*—one of the main restraints

considered in this study (as mentioned in Section 1 of Chapter 3). This upside is due to the nature of the proposed strategy; in other words, the stationary assets are naturally located within the surveillance tape, hence, positioned nearby the outpost and receiving adequate protection. Subsequently, such a constraint does not have to be modeled or integrated in the solution process.

To achieve a surveillance tape structure, four pieces of information are required:

1. The height and width of the inner rectangle that either represents, or is inscribed by, the perimeter of the outpost, as demonstrated in Figure 51.
2. The thickness of the virtual wall (i.e. the needed range of coverage), which in turn helps determine the dimensions of the outer rectangle.
3. An allowance for lack of coverage, which helps avoid an investment that adds little value to an almost complete coverage, as shown in Figure 52.
4. The coordinates (x,y) of the center of the inner rectangle (which happen to be the same for the outer one as well).

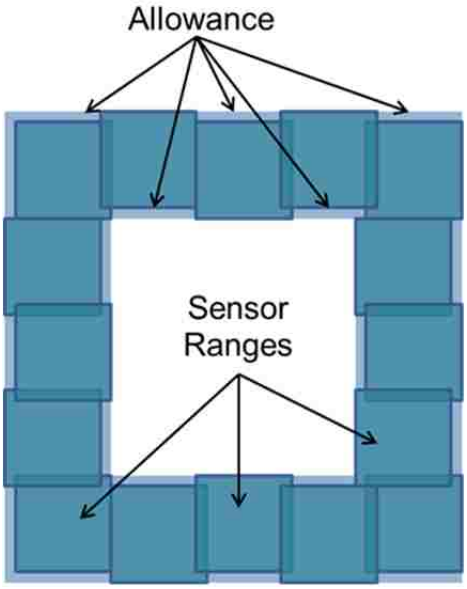

Figure 52: A solution example with multiple sensor ranges and allowance for lack of coverage

Once the virtual wall is established, the next step is to efficiently fill it with sensor ranges, as shown in Figure 52, in order to effectively monitor this zone. The task of positioning sensors to attain such a layout is adopted as an optimization problem, which from now on shall be referred to as the *Stationary System Positioning/Placement Problem* (SSPP). In the rest of Section 2, reasonable assumptions are established and justified in order to further simplify the structure of SSPP. Next follow the description of a solution process that considers investment alternatives with and without the aerostat system, the formulations of coverage contributions and overlaps, and a discussion about the algorithms invoked to efficiently position the surveillance assets.

2.1. Problem Complexity

SSPP is a complex optimization problem in the sense that it targets covering a certain percentage of the area of the thick virtual wall with the smallest possible number of sensors. To achieve this aim, a reliable method to measure the *effective* (overlap-free) *coverage* of a certain placement of stationary assets has to be established so as to check if the surveillance demand is satisfied. Furthermore, in addition to the nonlinear nature of the problem, there are about three more aspects that add complexity to the problem.

Firstly, to solve SSPP, there is this required flexibility to position a tower-camera in the east-, west-, north-, or south-side of the virtual wall. Hence, there is this inevitable need for the *logic-or* operator that allows the solution algorithm to achieve such positioning. On the other hand, a pure mathematical program assumes that all constraints are joined through a *logic-and* operator, a fact that renders this type of models lack the capability to solve SSPP. To overcome this intricacy, a fast construction algorithm is designed to take advantage of the virtual wall's four rectangular shapes (or segments) to simplify the solution process (explained in the Section 2.4).

Secondly, there is a delicacy in intelligently incorporating the aerostat and assessing its value added. This is due to the fact that such devices have a high cost of operation, maintenance, and support (OM&S) as well as several maintenance requirements, including manpower—an estimated cost of over 2.5 million dollars per year (Air Combat Command, 2007). This fact greatly limits the allocation of aerostats to COPs/OPs. In this study, it is assumed that a maximum of one tethered platform can be assigned to an outpost. This reasonable and practical restriction would also mean that the surveillance device is most likely to be positioned in the center of the military settlement (and hence, at the centroid of the virtual wall) in order to provide balanced coverage for all four segments. This established mindset raises new concerns: first, what is the degree of overlap between the virtual wall and the aerostat? This question is important as its answer affects the allocation of tower cameras. Another concern is related to whether or not the tethered vehicle represents a justified investment, as a significant portion of its coverage contribution lies within the outpost's area, but outside of the zone of interest—the surveillance tape. These aspects are addressed in Section 2.2.

Finally, there is the potential issue of having multiple kinds of mounted cameras. Such a situation creates the need to consider field of views and coverage ranges that have different dimensions; this renders the model more complex. On the other hand, there is considerable motivation to restrict the solution to only one kind of surveillance cameras for each solution run. This approach is associated with the fact that for central processing systems (of sensor output feeds), system compatibility, purchase, and maintenance may be an issue once two or more sensor types are utilized. Therefore, it is assumed that only one camera type is purchased in order to ease standardization, integration, and upkeep. (Note: prior to specifying the sensor-related parameters needed in this analysis, the user of this framework has two alternatives for evaluating the several brands of these mountable devices: (1) perform a pre-analysis sensor comparison, probably based on range and

resolution, and then select one candidate to be considered for procurement; or (2) run the actual analysis that is developed by this study for every considered camera type.)

2.2. Solution Process

To resolve SSPP, certain steps have to be followed in order to solve the right optimization problem at the right time. To be more specific, there is a need to know: (1) whether or not an aerostat is part of the solution, and (2) if this system is to be utilized by the outpost, are mounted cameras required to complement any uncovered surface of the virtual wall? As a result, a solution process is designed to identify all possible alternatives and their associated optimization solution method.

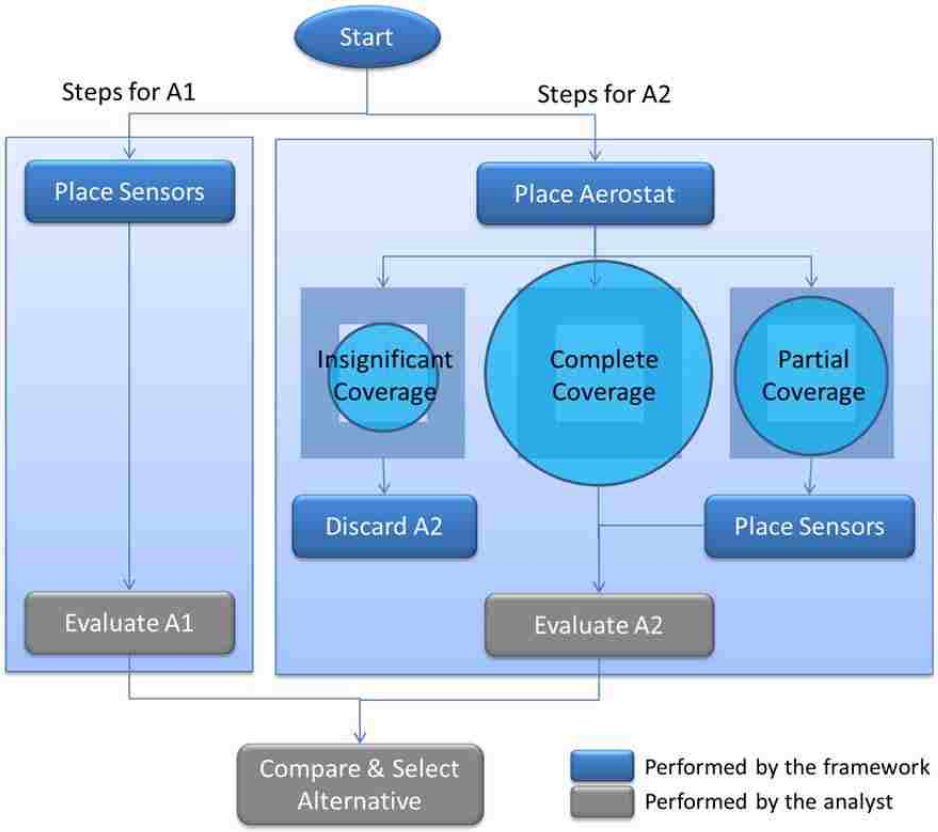

Figure 53: Steps of the solution process to select the best stationary surveillance alternative

The process starts by evaluating two investment alternatives: one that is based strictly on mounted cameras, and is referred to as A1; and one that includes one aerostat, and is referred to as A2.

First, the surveillance plan for A1 is constructed by efficiently placing camera ranges all over the virtual wall. This is achieved by using an *ad hoc* algorithm that is later discussed in details in Section 2.4. After the sensors are in place and the layout is complete, A1 is ready to be evaluated by the decision maker, a task that is most likely to involve cost analysis, and is therefore out of the scope of this study. Similarly, A2's plan is developed prior to its evaluation, but in a different way. To be more precise, the process of constructing a layout for A2 leads the analytical procedure to branch out to three mutually exclusive conditions. The first one is when the aerostat does not cover a significant portion of the virtual tape (set in this study as <50%); in that case, the investment option of acquiring the aerial platform is not justified, and thusly discarded. The second case is based on the situation where the aerostat covers all of the virtual wall's surface (and maybe much more); in such a state, the solution is complete and ready for evaluation. Finally, there is the scenario where the LTA platform covers a fairly high percentage of the tape (>50%), but still needs the help of tower/wall-mounted sensors to cover the whole surface (with some allowance). In this case, there is the complication of how to efficiently construct a hybrid plan for A2 (i.e. spread the cameras in the surveillance tape in a way that avoids the kind of coverage overlap that increases the cost of the solution without adding a considerable value). A high-level perspective of the solution process is depicted in Figure 53.

Note that as mentioned earlier, the evaluation and comparison of A1 and A2 are out of the framework's scope. The reason is that such analysis is expected to be intricate due to the existence of several decision criteria, and hence, needs its own rigorous study. To illustrate such complexity, consider expenditure, a measure that is affected by human resources, overhead cost, time value of money, and maybe even uncertainty (for example, despite its high cost, an aerostat might become an attractive option if the tape's thickness is predicted to increase over time). Such analysis is highly recommended as it results in plans and policies that are more financially sustainable over longer periods of time.

2.3. Coverage and Overlap Computations

The key to solving SSPP is understanding, and well formulating, the coverage of the surveillance systems, the resulting overlap amongst each other, the individual coverage contribution of each sensor, and the final coverage contribution of the sensors combination with respect to the virtual wall. In the next subsections, each of these items is discussed and an associated mathematical formulation is provided.

2.3.1. Coverage

In this study, the *range* or *coverage* of a stationary system simply means the projection of the sensor's field of view on the grounds' surface. As a result, the coverage area is a function of both the distance between the sensor and the ground (a.k.a. slant-range), and the alpha angle, as described in Figure 13 and Section 3 of Chapter 3. Also, it is worth noting that for stationary assets, this piece of information plays an important role in determining the sensor's coverage contribution, as opposed to mobile assets, which take advantage of motion to cover larger surfaces regardless of the size of their fields of view.

On the other hand, an increase in the slant-range distance and alpha angle, which leads to larger coverage, also reduces the resolution of the captured image. To handle this tradeoff, the alpha angle is set fixed to a specific value, whereas the projection area is evaluated at three different resolution levels (and hence three different slant-ranges): (1) the vehicle detection level, (2) the dismount detection level (with a slant-range four times smaller than that of level (1)), and (3) the dismount identification level (with a slant range four times smaller than that of level (2)). That being said, the coverage of tower-cameras and aerostats are determined in the following manner:

- The aerostat is assumed to carry sensors with an alpha angle of 90 degrees. This fact makes the FOV of the dominant sensor (which is most likely to be the AWAPSS-IR) to have a projection whose perimeter is a perfect circle. Assuming a FOV that is 120 degrees wide, the appropriate coverage radius can be found by multiplying the slant-range by Arctan(60°). The radius can be set to r_{ID} to identify dismounts, to r_{DD} to detect dismounts, or to r_{DV} to detect vehicles, by increasing or reducing the slant-range accordingly (see Figure 54). Note: the AWAPSS-IR typically has the following numbers: r_{ID} = 250m, r_{DD} = 1000m, r_{DV} = 4000m (look up the vehicle detection level slant-range in Table 3).

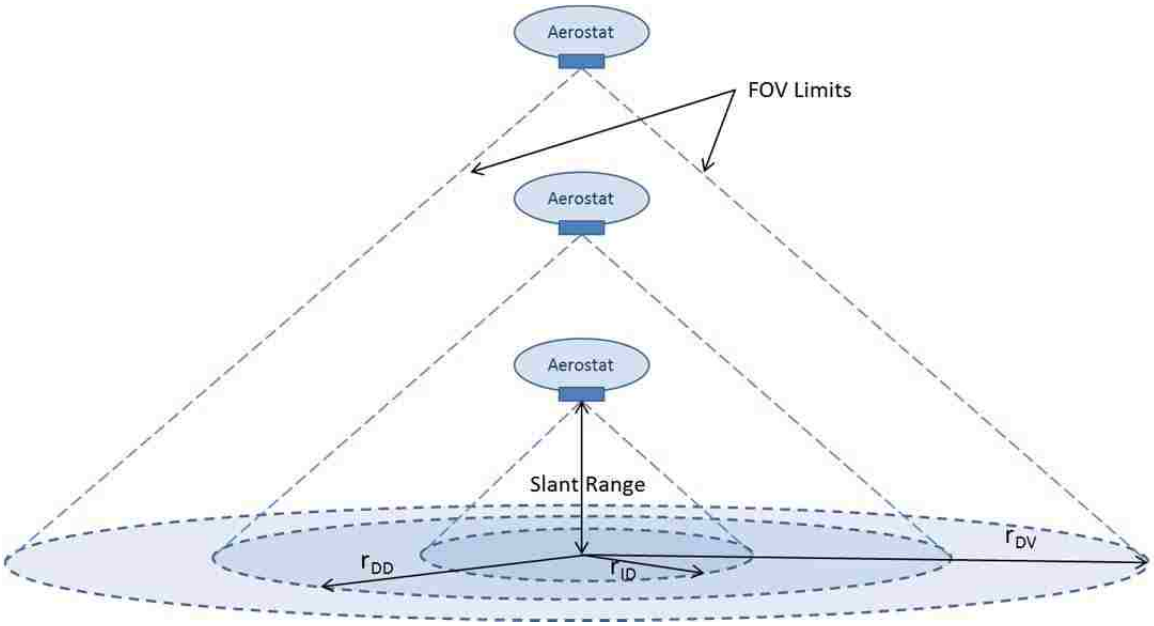

Figure 54: The representation of the LTA's sensor FOV for three different levels of surveillance

- For tower-cameras, the FOV concept is applied differently since the slant of the sensor—most likely to be the Raven-IR or a device with similar parameters—is set almost horizontally (i.e. the alpha angle is very small). The FOV projection on the ground surface can be represented as a long ellipse. Also, the cameras are assumed to have a panning capability of 360 degrees. As a result, the ellipse is rotationally translated around the sensor's location, creating a disk with a small hole in the center due to the blind-spot effect (see Figure 55). However, this blind-spot is

assumed to be very small, and is subsequently neglected. As a result, the coverage of mounted-cameras is also modeled as a disk.

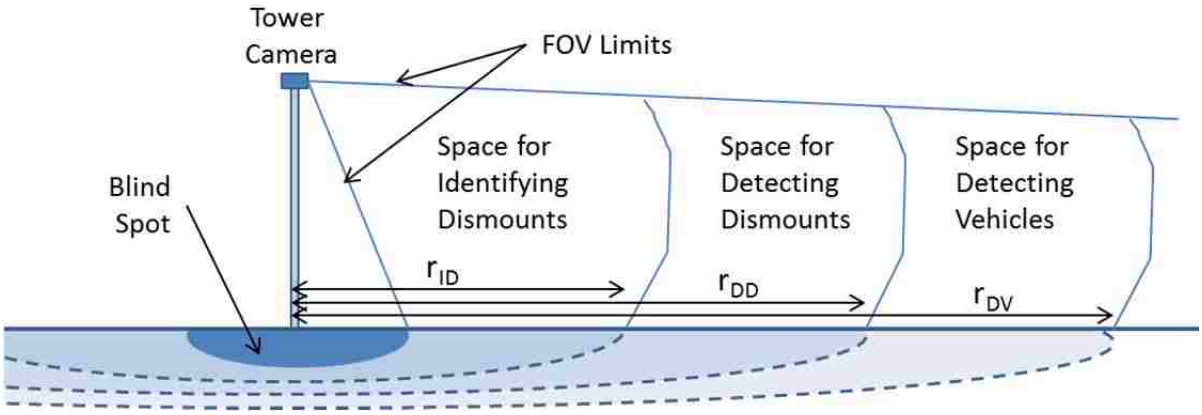

Figure 55: The FOV representation for a tower camera at three different levels of surveillance

From another side, the three resolution levels should still be captured. The way this is achieved is by determining the associated spaces where the detection and identification events can happen reliably, and then project them on the ground surface as disks. The corresponding radii (r_{ID}, r_{DD}, and r_{DV}) are estimated to equal their slant-range counterpart as the slant is almost parallel to the ground. Note: the Raven-IR typically has the following numbers: r_{ID} = 40m, r_{DD} = 160m, r_{DV} = 640m.

Finally, for the sake of simplicity when positioning towers in the virtual wall, and in order to well manage overlaps and coverage contributions, the FOV projection for mounted-cameras is modeled as a filled square rather than a disk. These squares are acquired by maximizing their dimensions while still staying inside the circular shape, as shown in Figure 56. Notice that this is a conservative approach because the sensor coverage is underestimated, and hence, the resulting surveillance plan is not prone to vulnerabilities due to this simplification. Moreover, after efficiently placing the tower cameras, the original shape of the coverage area is restored for more accurate reports and visualization.

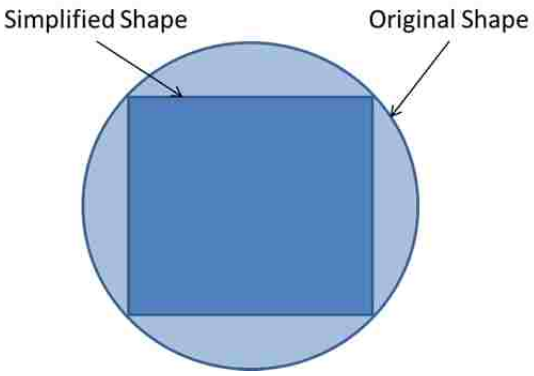

Figure 56: Triming the circular shape of the tower-camera's coverage to obtain a square

2.3.2. Overlap

In general, surface overlap refers to the occupation of the same area by two or more geometrical shapes. In this context, there is an interest in two kinds of overlaps, one of them is desired, and the other one is a bit fuzzy:

- The first type (referred to as O1) is the intersection between the sensor's FOV projection and one of the four segments of the surveillance tape. This measure is also referred as the *coverage contribution* (CC), as it determines (depending on the range size and location) the portion of a sensor's range that is actually meeting the demand for coverage, and hence effective to the surveillance plan.

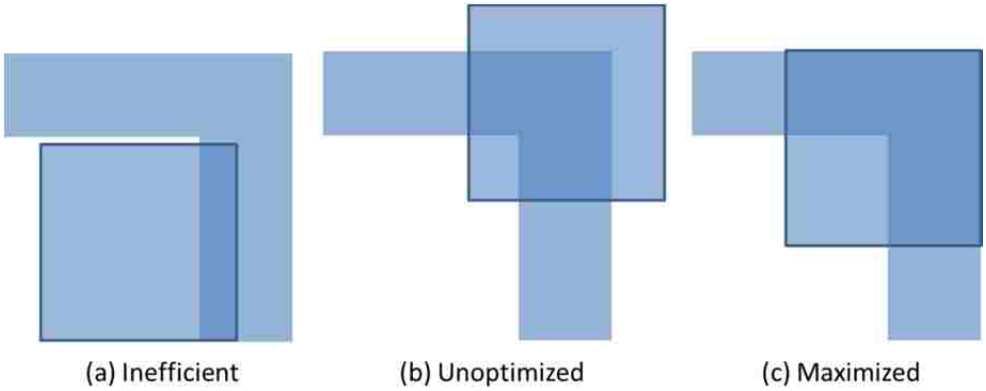

Figure 57: Three cases for positioning the squared shape and L-shape and their levels for overlap

This kind of overlap is desired, and therefore should be maximized (by efficiently utilizing the target asset, as opposed to adding another sensor to attain maximal coverage; Figure 57 depicts different levels of utilization efficiency).

- The second type (O2) refers to the intersection between any two or more sensor coverage contributions. This sort of overlap can be either desired or undesired, depending on the situation: if more overlap means increased coverage redundancy (and hence, more security for the outpost) without increasing the cost of the solution, then that is ideal; however, if O2 overlap is the result of positioning cameras inefficiently close to each other, requiring more assets to cover the target area, then that is undesired. Yet, regardless of the degree of desirability or undesirability, this measure has to be computed in order to calculate the surveillance plan's total CC without counting the overlaps twice.

It is worth emphasizing that to accurately compute the O2 overlaps, it is necessary to understand that O2 formulations depend heavily on O1 formulas. To elaborate, the overlap of interest between two sensor ranges is strictly the portion of their intersection area that is situated inside the surveillance tape, and not simply their intersection area. As a result, the coverage contributions of the two sensors have to first be obtained by acquiring their O1 overlap, and then the two CCs can be used to determine the O2 overlap through their intersection. Figure 58 shows an example of two sensors i and j covering a segment of the virtual wall; the illustration distinguishes the total overlap between the FOV projections (with the darker blue fill), the coverage contributions (with shade lines; also surrounded by dashed lines), and the undesired O2 overlap (with intersecting shade lines). Note that four pieces of data are needed to compute the area of O2 overlaps, and they are the centroid's (x,y) coordinates, width, and height of O1 overlaps.

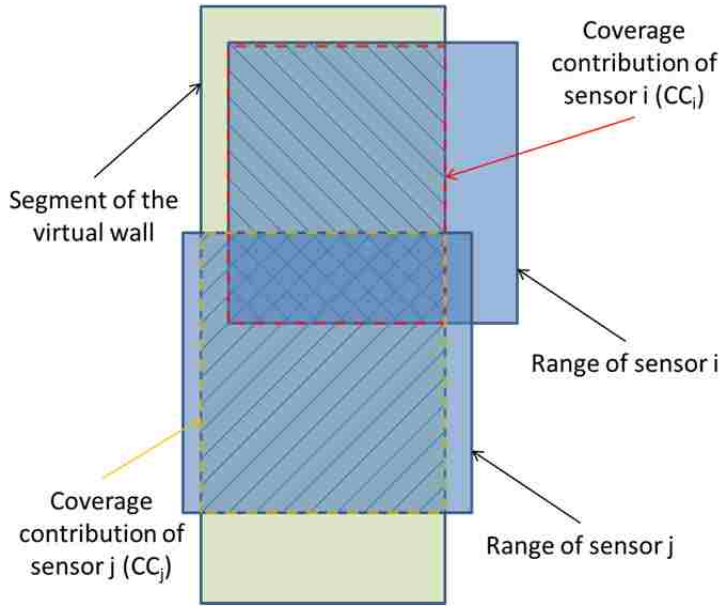

Figure 58: Two overlapped sensor ranges positioned over a segment of the virtual wall

Another useful aspect is that since both the virtual wall's segments and sensor ranges are assumed to have a rectangular shape at this point, both kinds of overlaps (O1 and O2) yield simple area computations (i.e. A = w . h, where w is the width and h is the height) as the intersection of two rectangles is a rectangle. Nevertheless, there is some complexity in formulating the input parameters themselves (w and h) since the dimensions of the virtual wall segments and the FOV projections can hold any value in theory. In practice, the interval of input values is much more limited, yet it still contains an unlimited set of numbers. Thus, the overlap formulation is constructed in a generic fashion in order to account for the many input values, and specifically for the following scenarios:

(a) two rectangles, i and j, are far away from each other (not having any overlap)

(b) rectangle j is completely inside rectangle i

(c) rectangle j is partially inside rectangle i

(d) rectangles i and j cross each other with maximum overlap without one fitting inside the other

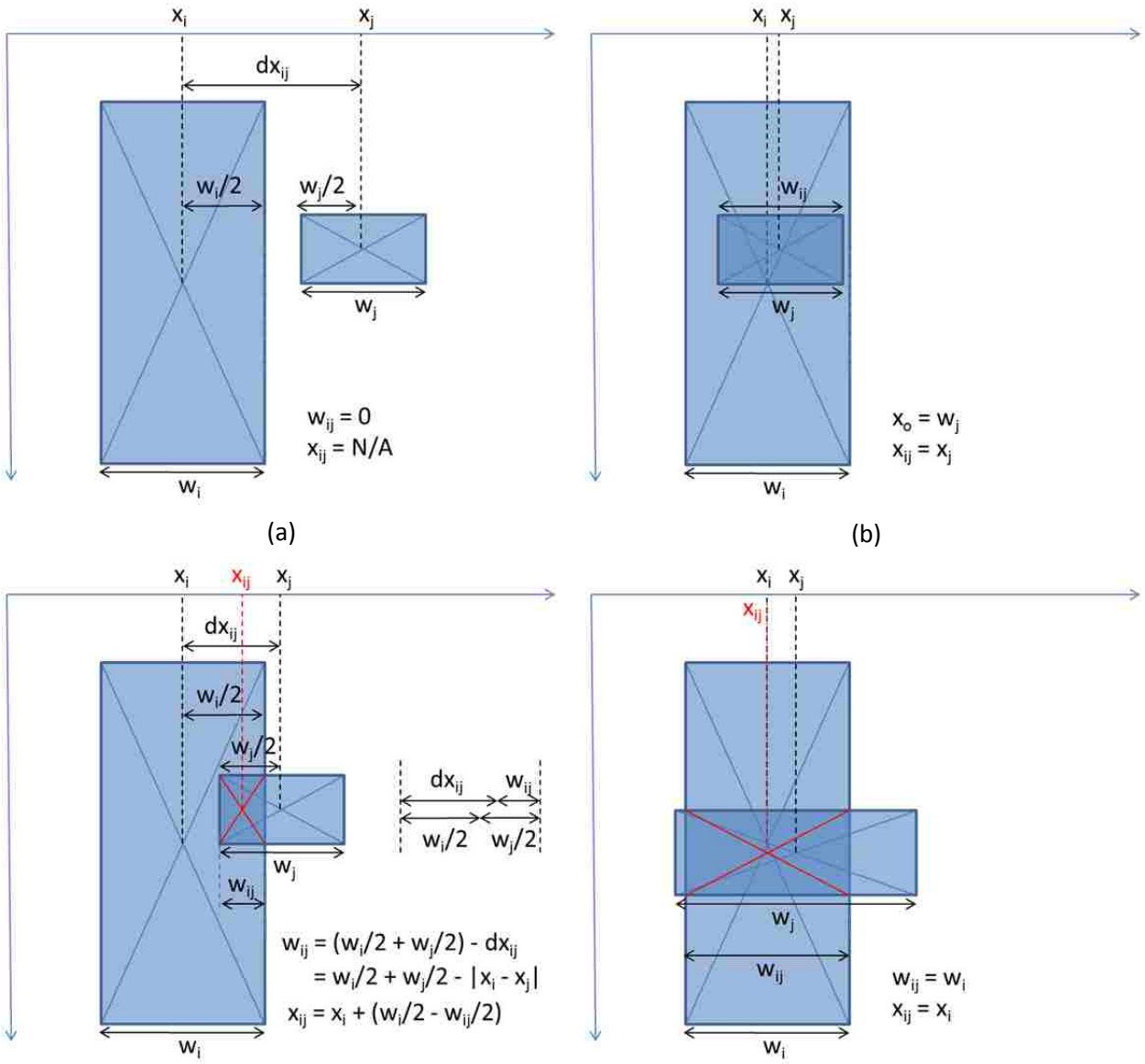

Figure 59: The four scenarios of overlap with regards to the width w_{ij} and x-coordinate x_{ij}

To formulate a two-shape overlap regardless of its type (i.e. both O1's and O2's), let (x_{ij},y_{ij}) be the coordinates of the rectangular intersection of rectangles i and j, while w_{ij} and h_{ij} be its width and height, respectively (and therefore $A_{ij} = w_{ij} \cdot h_{ij}$ is the area of the overlap). To render the formulation process easier, each of the aforementioned scenarios is applied on w_{ij} solely (and without losing any generality), as h_{ij} can be attained with the same footsteps as w_{ij}. To proceed with a complete

formulation, let i and j have widths w_i and w_j, heights h_i and h_j, x-positions x_i and x_j, and y-positions y_i and y_j, respectively.

The dimensions:

First, consider w_{ij}, the width of the overlap; the four aforementioned scenarios are applied to w_{ij} in the following manner (also depicted in Figure 59):

(a) When rectangles i and j are far apart, w_{ij} is zero:

$$w_{ij} = 0$$

(b) When rectangle j is inside rectangle i, w_{ij} is the width of j:

$$w_{ij} = w_j$$

(c) When rectangle j is partially inside rectangle (with respect to the x-axis), w_{ij} is the sum of the half widths $w_i/2$ and $w_j/2$ minus dx_{ij}, the distance between the rectangles' centroids (as long as the subtraction is positive):

$$w_{ij} = \max(0, w_i/2 + w_j/2 - dx_{ij}) = \max(0, w_i/2 + w_j/2 - |x_i - x_j|)$$

(d) When rectangles i and j cross each other with maximum overlap without one fitting inside the other, w_{ij} is the minimum of the two widths:

$$w_{ij} = \min(w_i, w_j)$$

Note that scenarios (a) and (b) are particular cases of scenarios (c) and (d), respectively. Therefore, the final formulation, along with an adequate function name $F_w(i,j)$, is:

$$w_{ij} = F_w(i,j) = \min\{w_i, w_j, \max(0, w_i/2 + w_j/2 - |x_i - x_j|)\}$$

Similarly, h_{ij} can be computed through the function $F_h(i,j)$ as follows:

$$h_{ij} = F_h(i,j) = \min\{h_i, h_j, \max(0, h_i/2 + h_j/2 - |y_i - y_j|)\}$$

which leads to the following overlap area formulation:

$$A_{ij} = F_w(i,j) \,.\, F_h(i,j)$$

$$A_{ij} = \min\{w_i, w_j, \max(0, w_i/2 + w_j/2 - |x_i - x_j|)\} \,.\, \min\{h_i, h_j, \max(0, h_i/2 + h_j/2 - |y_i - y_j|)\}$$

The Coordinates:

As for the formulation of the coordinates (x_{ij},y_{ij}), the above four scenarios are evaluated, and the process is simplified by first considering x_{ij} only (without loss of generality):

(a) When rectangles i and j are far apart, x_{ij} does not exist.

(b) When rectangle j is inside rectangle i, x_{ij} is the same as x_j:

$$x_{ij} = x_j$$

(c) When rectangle j is partially inside rectangle (with respect to the x-axis), x_{ij} is the:

$$x_{ij} = x_i + (w_i/2 - w_{ij}/2), \quad \text{if } x_i < x_j$$

$$x_{ij} = x_i - (w_i/2 - w_{ij}/2), \quad \text{if } x_i > x_j$$

(d) When rectangles i and j cross each other with maximum overlap without one fitting inside the other, x_{ij} is the same as the x-position of the rectangle with the smallest width:

$$x_{ij} = x_i, \quad \text{if } w_{ij} = w_j \quad (\text{i.e. } w_i < w_j)$$

$$x_{ij} = x_j, \quad \text{if } w_{ij} = w_j \quad (\text{i.e. } w_i > w_j)$$

Assuming there is an overlap between rectangles i and j (i.e. the condition $x_i - x_j \leq w_i/2 + w_j/2$ is verified) and that w_{ij} is computed (through the function $F_w(i,j)$), the x-coordinate of the overlap, x_{ij}, can be computed through the function $F_x(i,j)$:

$$x_{ij} = F_x(i,j) = \begin{cases} x_i & \text{if } w_{ij} = w_i \\ x_j & \text{if } w_{ij} = w_j \\ x_i + (w_i/2 - w_{ij}/2) & \text{if } x_i < x_j,\ w_{ij} \neq w_i,\ w_{ij} \neq w_j \\ x_i - (w_i/2 - w_{ij}/2) & \text{if } x_i > x_j,\ w_{ij} \neq w_i,\ w_{ij} \neq w_j \end{cases}$$

Similarly, the y-coordinate y_{ij} can be obtained using the variables y_i, h_i, y_j, h_j, and h_{ij} via a function $F_y(i,j)$.

2.3.3. Coverage Contribution

In order to be able to construct or search for a good solution for SSPP, a well-established coverage measure is necessary. In this context, the objective is to attain an efficient surveillance coverage setup of stationary assets for the outpost. Therefore, the previously developed overlap formulations are employed as a foundation to construct a formula that computes how much coverage a certain layout of sensor ranges yields. In order to achieve this, four steps are followed: (1) establishing assumptions and defining the variables, (2) formulating the coverage contribution of a sensor, (3) stating the overlap formula between two sensors, and (4) presenting the total surveillance coverage equation.

First, assume that the 2D space to be covered, A, has a rectangular shape (as opposed to the virtual wall's shape, which is empty from the inside) where the top left corner is the origin of the frame (0,0). This assumption is later scaled to meet the coverage demand of the surveillance tape. Let W be the width of this rectangle, and H its height. Also, for any sensor i (with a square shape), let $s_i = s$ be its side's length, x_i its x-coordinate, and y_i its y-coordinate. Furthermore, let C_i be the sensor's coverage contribution (i.e. its FOV's intersection with the space A), and Cw_i, Ch_i, (Cx_i,Cy_i) be its width, height, and coordinates, respectively. Finally, let O_{ij} be the O2 overlap between sensor i and j (i.e. the intersection of their coverage contributions) with Ow_{ij} and Oh_{ij} as its width and height, respectively.

Second, consider the following process to calculate the CC parameters by employing the appropriate functions and formulas developed in the previous section:

$$Cx_i = F_x(i,A)$$

$$Cy_i = F_y(i,A)$$

$$Cw_i = F_w(i,A) = \min\{W, s, \max(0, W/2 + s/2 - |W/2 - x_i|)\}$$

$$Ch_i = F_h(i,A) = \min\{H, s, \max(0, H/2 + s/2 - |H/2 - y_i|)\}$$

$$C_i = Cw_i \,.\, Ch_i$$

Third, obtain the O2 overlap between any two sensors i and j:

$$Ow_{ij} = \min\{Cw_i, Cw_j, \max(0, Cw_i/2 + Cw_j/2 - |Cx_i - Cx_j|)\}$$

$$Oh_{ij} = \min\{Ch_i, Ch_j, \max(0, Ch_i/2 + Ch_j/2 - |Cy_i - Cy_j|)\}$$

$$O_{ij} = Ow_{ij} \,.\, Oh_{ij}$$

Fourth, assuming there are N sensors during a certain evaluation run, calculate the overall CC of the surveillance plan (CC_T) by subtracting O_T, the summation of overlaps between every two sensors (if they are not far apart), from C_T, the sum of all the sensors' coverage contributions:

$$O_T = ½ \,.\, \sum_{\{i \text{ in } 1..N,\, j \text{ in } 1..N,\, i \neq j,\, |xi - xj| \leq (wi + wj)/2\}} O_{ij}$$

$$C_T = \sum_{\{i \text{ in } 1..N\}} C_i$$

$$CC_T = C_T - O_T$$

Notes:

- Because the overlap between two sensors is counted twice in the O_T summation—once for O_{ij} and another time for O_{ji}, this summation is divided by 2.
- This equation assumes that no aerostat is utilized.

2.4. Sensor Placement

The task of efficiently placing the squared (reduced) FOV projections (with a side length s) on the virtual wall can be a challenging task. It can however be simplified by separating the tape into four segments and filling each one separately. Since each of these segments has a rectangular shape, it adheres to the assumption established in the previous section, and hence qualifies for the developed overlap and coverage contribution computations. On the other hand, these four parts overlap at the corners, and so the asset positioning algorithm has to avoid allocating duplicates for the same spot.

To understand how the SSPP solver works, first assume that the side length of a sensor's range is larger or equal to the thickness of the virtual wall. Because a corner of the surveillance tape is where the

two segments meet, placing a sensor there yields the best use of the asset, as the sensor range leads to the highest O1 overlap possible (as depicted in Figure 57). Therefore, the algorithm starts by placing four assets in the four corners of the tape. The second step is to compute the CC_T measure for each segment and check whether or not the following condition holds, whereby $A_{segment}$ is the area of the segment and a is the allowance for lack of coverage:

$$CC_T \geq A_{segment} \cdot (1 - a)$$

If this condition is verified, the segment is finalized—by stopping the associated iteration as well as recovering the original circular shape of the FOV projections. Figure 60 for instance shows a virtual wall with four segments that meet the coverage demand, and are therefore finalized.

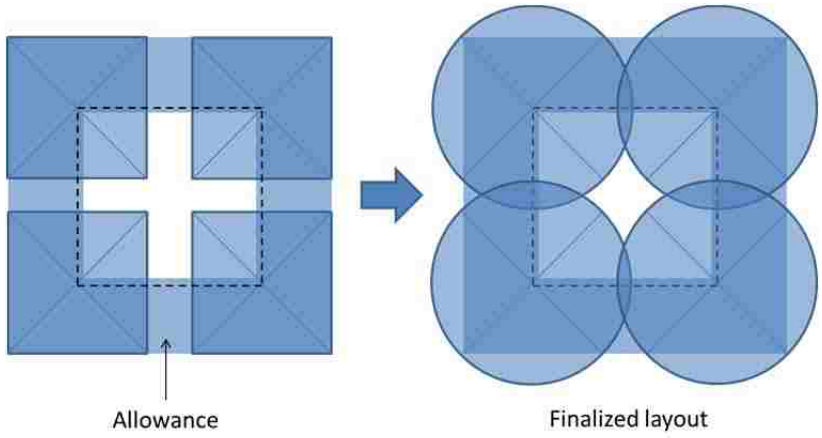

Figure 60: A solution for a SSPP instance where four sensors suffice to provide apt coverage

If the condition is not verified, that means that the amount of uncovered surface $[A_{segment} \cdot (1 - a) - CC_T]$ is positive and that more coverage is needed; and to increase it, the algorithm determines the N new assets that are to be added to the segment (whose width W is larger than its height H, without loss of generality) by computing:

$$N = \lceil [A_{segment} \cdot (1 - a) - CC_T] / (\min(H, s) \cdot s) \rceil$$

Note that the amount (min(H, s) . s) is the maximum coverage that a sensor can contribute.

The process then continues by filling the uncovered surface by positioning the centroids of the newly added camera ranges on the line that passes through the centroids of the corner sensors' ranges (referred to as the *centroids line*) such that the adjacent ones are equidistant from one another, like shown in Figure 61.

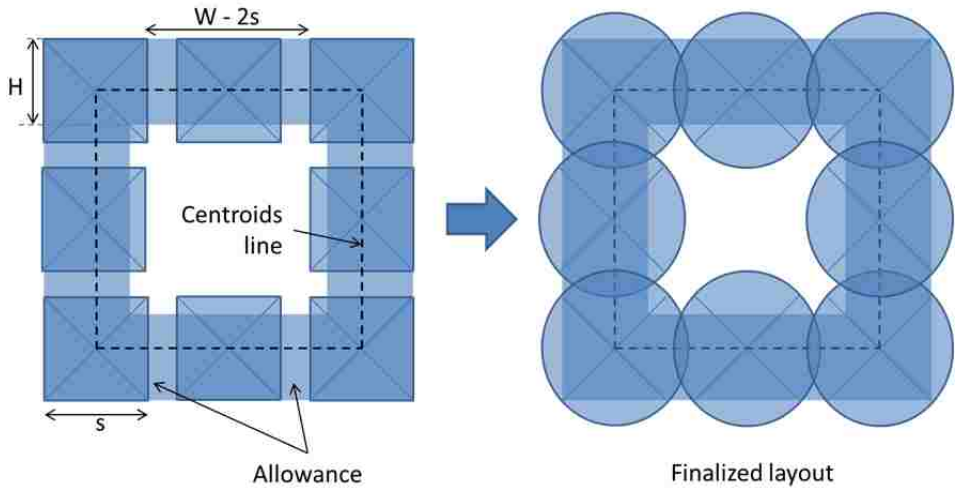

Figure 61: A solution for a SSPP instance where a layer of FOVs suffices to provide apt coverage

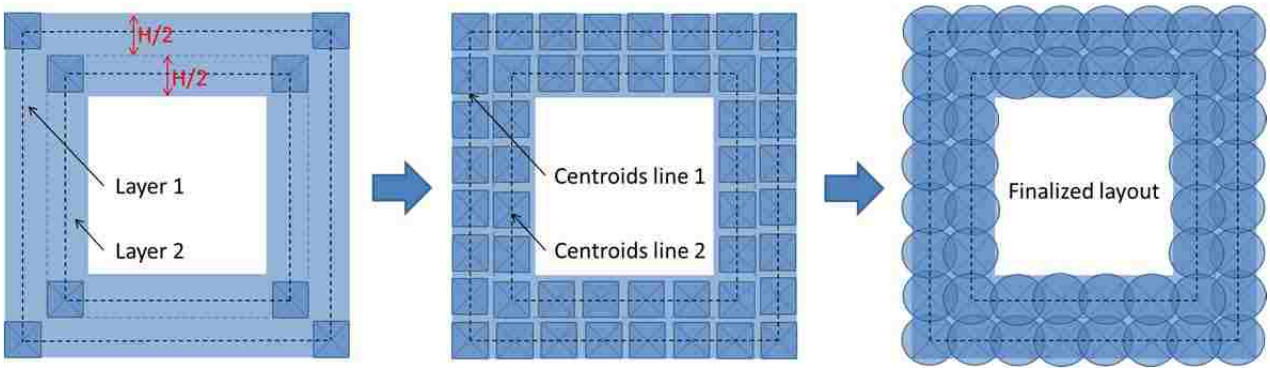

Figure 62: A solution for a SSPP instance where two layers of FOVs are needed for apt coverage

Now, consider the case where the thickness of the surveillance tape is larger than the side length of the sensor's range. In such a setting, if the coverage demand (after incorporating the allowance) is not met, then the virtual wall is divided into a set of layers that have the same thickness. Given this division, each layer can be considered as a mini virtual wall. Subsequently, the above described algorithm can be applied to each layer separately to acquire the set of ranges whose centroids

are all aligned in each segment (see example in Figure 62). Note that there may be some overlap between FOV projections that belong to different layers.

Let M be the minimum number of layers required to meet the coverage demand. By holding true the previously established assumption $W \geq H$, M is acquired by dividing H, the height of the segment (which is the same as T, the thickness of the virtual wall) by s, the length of the sensor's range side:

$$M = H / s = T / s$$

Notice that the quotient of the division can contain a fraction whereas M needs to be an integer. In that case, the floor of the number is used first; if it does not meet the coverage demand, then it is replaced by the ceiling.

Finally, the holistic algorithm that positions the sensors in the virtual wall, and consequently reveals the number of assets needed for the A1 investment alternative, can be compiled in a concise way through the following steps (and whose corresponding pseudo-code can be found in Section 8 of the Appendix):

(0) Obtain the length of the sensor's range (s) and the virtual wall's established parameters—the width (W) and height (H) of the outer rectangle, the coordinates (x,y) of its center, the allowance (a) for lack of coverage, and the thickness (T) of the tape (which is used as the width or height for the four segments of the surveillance tape).

(1) Assign M the floor of the division T/s. If this calculation gives 0 as a result, assign M a value of 1.

(2) Divide the virtual wall into M layers. Note that it is possible to have one layer only.

(3) For each of the M layers, perform the following substeps:

    a. Place a sensor at each corner in a way that maximizes O1 overlaps.

b. For each of the four mini-segments of the layer, compute the total coverage contribution CC_T (as instructed in Section 2.3.3), and check if the coverage demand is met (i.e. verify the condition $CC_T \geq A_{segment} \cdot (1 - a)$).

c. For each of these mini-segments that have insufficient coverage, place N additional sensors equidistantly along the centroids line. N is computed as instructed above.

(4) If the overall coverage demand is met, finalize the constructed layout and return it as output. Otherwise, assign M the ceiling of the division T/s and re-execute the steps (2)-(4).

2.5. Sensor Placement with Aerostat

Assume that the A2 investment alternative is justified, and that it requires the use of mounted sensors to complete the coverage for the surveillance tape. The approach taken to build a hybrid surveillance system that comprises one aerostat and a set of wall/tower cameras can be divided to three steps:

(1) As described in the previous section, build a layout solution for investment alternative A1 (which contains only mounted cameras).

(2) Position the tethered platform such that the center of its range coincides with the centroid of the virtual wall. Note that unlike the first step, the sensors ranges are not modeled as squares due to the large scale of the aerostat's FOV projection (i.e. each one of the four trimmed parts of the disk that lead to a squared shape, as shown in Figure 56, may contain several camera ranges in the aerostat's case, which is inappropriate to ignore).

(3) Remove the cameras whose coverage is deemed unnecessary, i.e. whose overlap with the aerostat's range is significant. An example of such a situation is depicted in Figure 63. While this step can be carried out manually using best ocular judgment, a more advanced method is

introduced below so that the solution can be implemented programmatically and hence, can offer a scalable way to address a large number of surveillance assets.

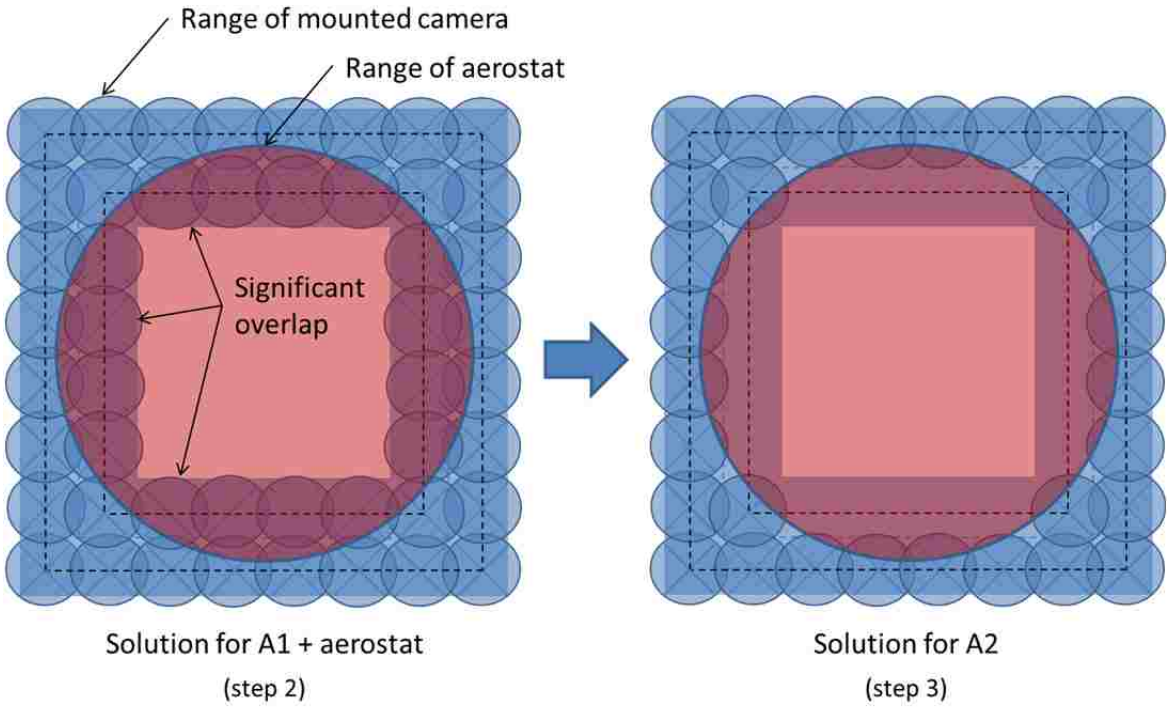

Figure 63: The transition from a solution for A1 to a solution for A2

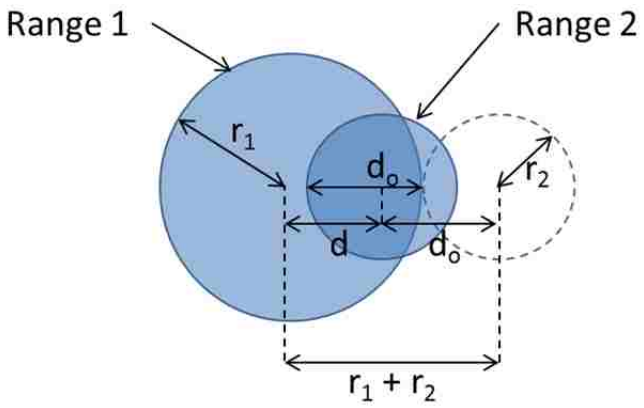

Figure 64: The geometric situation of two overlapping circular ranges with r_1 and r_2 as radii

To comprehend this method, consider two circular sensor ranges that overlap, as depicted in Figure 64. The line that passes through their centers intersects with their overlap in a segment whose length shall be called d_o. d_o can be obtained by subtracting d, the distance between the two ranges'

centers, from (r_1 + r_2), the sum of their radii. Note that d_o is positively correlated with the intersection between the sensors and can therefore be employed as a means to determine whether or not the amount of overlap is considerable. To be more specific, if d_o is larger than a certain threshold (k . r_2)—where k can be specified as an input to this framework, but would have a default value of 90%—then the intersection area between the FOV projections is regarded as significant. If such a situation is identified when evaluating the overlap between the aerostat's range and the mounted-camera's, the latter asset is removed from the hybrid surveillance system. This approach is iterated through for each mounted sensor until the coverage provided by investment alternative A2 is optimized.

3. Strategic Systems

Due to the high cost of purchase, maintenance, and manning of strategic systems, the efficient use of these resources requires timesharing them amongst outposts, bases, and named areas of interest (NAIs). Thus, in order to make a decision about the portfolio of assets to purchase, certain aspects have to be considered at a larger scope (i.e. region-based, as opposed to outpost-based). In this section, a different modeling notion is presented, its constraints are discussed, and their impact on the scope of this study is explained.

The surveillance coverage model of this section is based on the nature of strategic vehicles. They operate from high altitudes, they are much less vulnerable to obstructions and weather conditions, and their swath can cover large areas (e.g. AWAPSS can provide a FOV projection of 8x8km^2). Consequently, their utilization can be carried out in a comprehensive fashion by scanning an entire region (as opposed to visiting specific routes or spots only), which would strategically comprise several OPs, COPs, NAIs, and one or more FOBs. As a result, activities and changes on the geo-surface can be monitored in a way that allows outposts and their backups to be alerted when the threat is still at a great distance from its destination. This mindset leads to modeling the coverage of strategic systems as a surface area—

referred to as a *search-box*, usually a rectangle or a circle—that is much larger than the FOV projection of the sensors carried onboard. Moreover, the search-box concept also comes with the *flight patterns* element that is concerned with the efficient fulfillment of the comprehensive scan; see Figure 65 and Figure 66 for illustration. This paradigm can also be extended to watch over a set of search-boxes.

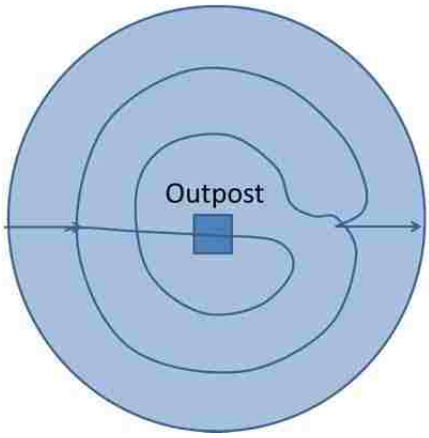

Figure 65: A circular surface area with a spiral flight pattern

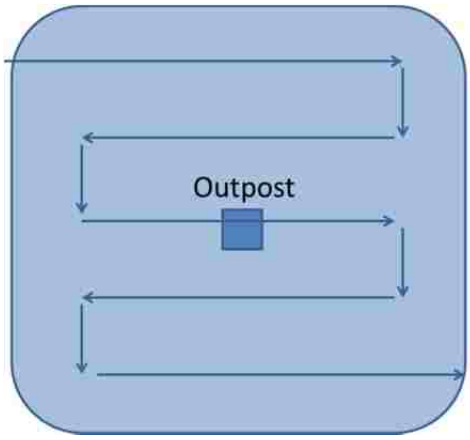

Figure 66: A rectangular surface area with a serpentine flight patterns

This search-box notion seems to provide a model to solve the strategic surveillance problem with more flexibility as some of the restrictions are not applied (e.g. payload limits, communication ranges, wind resistance, terrain elevation, need for stealth or protection). Nonetheless, this model does submit to certain restraints and performance measures that notably influence the process of finding an adequate procurement solution. First, during the cyclical tours of a large platform, the source and destination at the start and end of each surveillance cycle ought to have a runway as well as special accommodations, equipment, and manpower to perform maintenance and refueling; such places are most likely to be FOBs. Second, due to the vastness of the area scanned, and hence, the long distance traveled, the endurance of the aerial vehicle has to be considered when designing the flight pattern. In particular, the higher the endurance is, the longer the tours can be before returning to the origin in order to refuel. Third, a combination of the following three parameters helps determine how fast a certain region can be scanned: (1) the image resolution of the onboard sensors, as this defines the

maximum altitude that enables accurate data collection; (2) the operating altitude—which should be held constant as much as possible so as to minimize fuel consumption—as this determines the FOV projection of the sensor, and as a result, the area covered at a certain point in time; (3) the cruise velocity, which ought to be slow enough to allow accurate terrain scanning. Finally, the intermittence in service requires a robust schedule so that the revisit-time for each region does not create an opportunity for the threat to traverse the monitored terrain unnoticed. The issue here, as mentioned in Chapter 4, is that revisit times depend on the risk level of the target area. (Note that although it may be measured differently, the aspect of risk in this context still pertains to the potential of insurgents' presence; on the other hand, risky regions are different from hotspots, mainly due to the difference in surface size.)

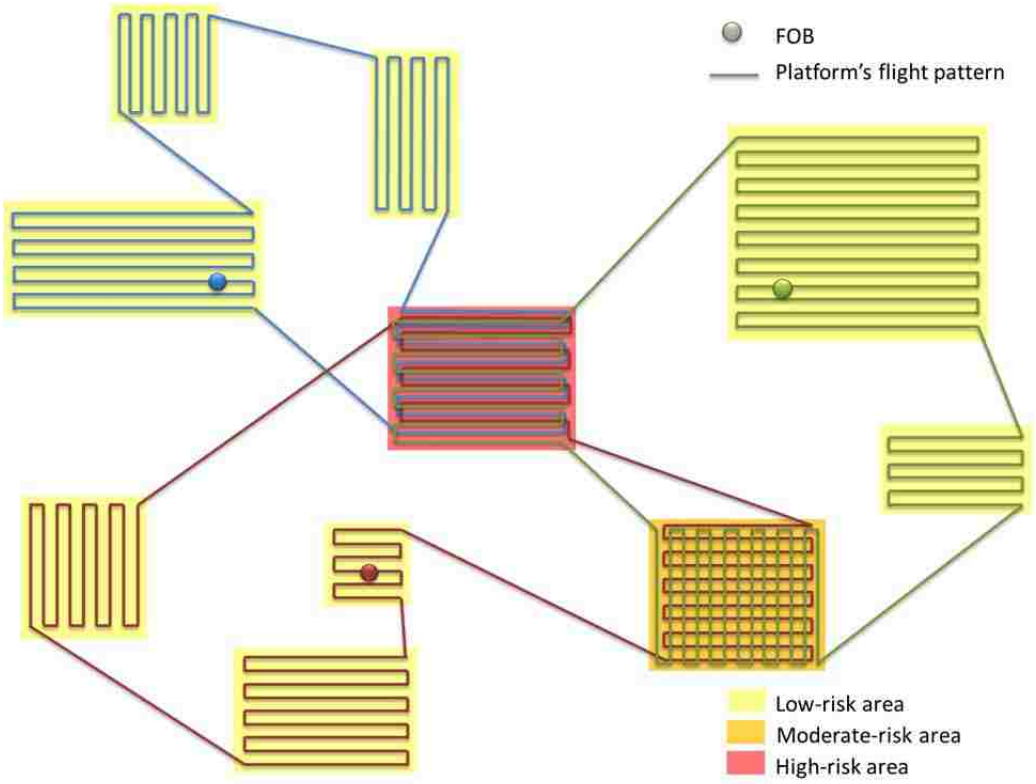

Figure 67: A surveillance plan for ten regions serviced by three strategic vehicles

Riskier regions would have to be scanned more often, which would require adjusting the flight patterns to achieve redundant subtours for riskier search-boxes or adding more assets to the portfolio to lessen

the service intermittence. Figure 67 for instance shows a map with ten risky regions, one of which is high-risk and another with a moderate risk level; the surveillance plan is designed so that the high-risk search-box is visited three times and the moderate-risk one is visited twice in each cycle. In this example, the plan takes advantage of three neighboring FOBs to achieve redundancy through shared resources, as opposed to employing the same asset to service the same (high- or moderate-risk) search-box more than once in the same cycle, and subsequently, increasing the revisit-time for other regions. From this preliminary analysis, it is apparent that constructing a coverage plan for large-scope surveillance—i.e. monitoring the vast regions around a set of outposts—requires a considerably different kind of analysis than the (micro-level) ones performed in the previous sections; it requires a macro-level analysis.

This macro-level analysis also necessitates the evaluation of input data whose scale is beyond the data provided for a specific outpost. To elaborate, risk has to be evaluated from a strategic perspective, not a tactical one. Also, the size and shape of the search-boxes have to be designed in a way that accounts for the geo-locations of the places needing protection, the distance traveled and endurance requirements, historical attacks and battles, proximity to other bases or outposts that have a runway, and some other factors mentioned in Chapter 4 such as trafficability and population. Next, the sequence of visits for each search-box (including redundancy if necessary) has to be determined before constructing an efficient flight patterns that starts and ends at the adequate origin. Furthermore, if the number of assets assigned to the set of target regions does not provide the desired revisit-time, the macro-level analysis might have to be reiterated through with a larger set of resources until the desired level of revisit-frequency is achieved. Because of the difference in scale and complexity between the macro- and micro-level analyses, the allocation of strategic systems and their associated purchase recommendations are considered out of the scope of this study.

CHAPTER 7: CASE STUDY

At this point, the target analytical framework is fully introduced and thoroughly explained. The next step is to apply it on a real-life scenario in order to validate its output solution as well as showcase its powerful utility. The chosen study case is COP Kahler, the outpost that sustained heavy damage during the *Battle of Wanat*, a conflict started by an attack by approximately 200 insurgents on July 13th, 2008. A summary of facts is extracted from *Wanat: Combat Action in Afghanistan, 2008*, a report released by The Staff of the U.S. Army Combat Studies Institute (SUSACS) in 2010. Next, gleaned information and provided data about the Wanat region are inserted and run in the analytical engine, and the output of each phase in the analysis is exposed and adequately discussed. Last but not least, conclusions about the recommended surveillance plan are concisely conveyed.

1. Background

COP Kahler was founded on July 8th 2008 in the village of Wanat, near the Waygal river, in order to establish stability and allow the Coalition Forces to improve their relationship with the local inhabitants. This establishment was also a way to relocate the forces from COP Bella, which was situated in a region considered hostile after July 4th, where an attack from the *Anti-Afghan Forces* (AAF) was expected. This relocation mission was called *ROCK MOVE*, and was performed under high time pressure, which led to a lack of settlement preparations in Wanat. COP Kahler comprised a total of 72 soldiers, 48 of which were Americans, and the rest were part of the Afghan National Army (ANA). The group was very well equipped with weapons and firepower. They also set an observation post—named *OP Topside*—about 75 meters from the COP command post (approximately 50 meters from the COP's perimeter) on a ridge in the eastern side. The combat outpost was about 110x50 meters squared, at an elevation around 3350 feet, and adjacent to the village's mosque, hotel/restaurant, and bazaar as shown in the proposed outpost layout depicted in Figure 68.

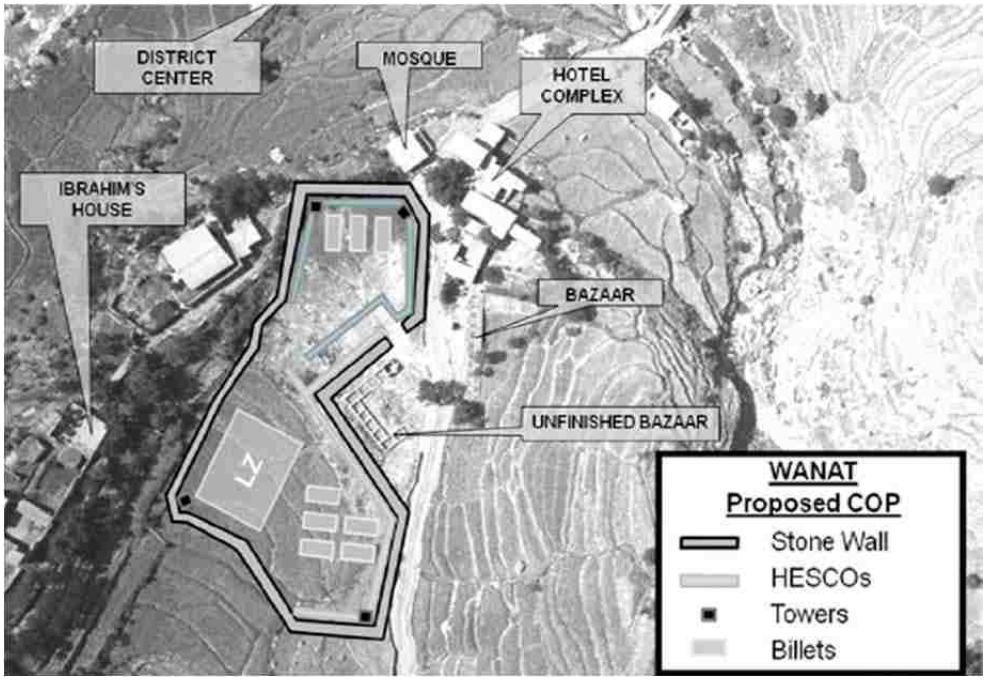

Figure 68: The proposed layout of COP Kalher
Source: SUSACS (2010)

The stone walls as well as other parts of the proposal were however not yet implemented before the insurgents' attack due to the early displacement of the forces, as well as the arrival latency of the contracted builders. This was one of the main factors that contributed towards the COP's vulnerability. Furthermore, there were three other factors that gave the adversary a great advantage. Firstly, the insurgents were able to infiltrate the village as civilians and collect intelligence about the outpost, although some of the villagers approached the COP forces and warned them about an imminent attack. Secondly, there was a severe lack of water, whereas the heat during the daytime was high; this crippled any labor efforts towards building protective obstacles. Lastly, the outpost was located in a low-level geo-platform that was surrounded by higher grounds and ridges (see Figure 69). Due to this setting, as well as the presence of long trees, visibility was obstructed even for OP Topside, a fact that greatly limited the situational awareness of the soldiers.

Figure 69: A view of the Wanat village from the north
Source of original picture (without the labeled white overlays): SUSACS (2010)

On the other hand, the outpost did have some access to ISR assets, but it was either insufficient or utilized ineffectively. To be more specific, *Combined Joint Task Force CJTF-101* did not provide ROCK MOVE with the few agreed upon FMV (by the Predator) and SIGINT hours due to reduction in service and relocation of assets to serve higher priority areas such as the Tangi Valley. Moreover, *TF Bayonet* from the nearby FOB did not send their Shadow and Hunter UAVs to Wanat because of the rugged nature of the Waygal Valley, which was believed to block the radio signals responsible for controlling the aerial vehicles. Also, the personnel at Kahler were not assigned any organic UAVs—not even the highly portable Raven—because it was believed that the hassle of employing the device—which was vulnerable to crosswinds and blocked signals—as well as the trouble associated with finding it and repairing it if lost during a flight mission were not worthwhile. Finally, there was the discouraging aspect associated with the intimidating difficulty of finding insurgents using a small UAV in a large area of mountainous and forested terrain. Lieutenant Colonel Pierre Gervais described it as "looking through a

soda straw… it's hit or miss" (SUSACS, 2010). This lack of long-distance surveillance service impaired the COP's ability to attain appropriate situational awareness and debilitated their readiness for insurgents' attacks. The Battle of Wanat eventually ended with 9 dead and 31 wounded from the coalition forces side, and that is with the reinforcements they received an hour after the conflict started. The backup support was powerful and effective, and ultimately forced the insurgents to withdraw.

2. Analysis

By contemplating the different facts of this event, valuable inferences can be derived and adopted as the foundation of this case study. First, there was no utilization of organic UAVs due to the inconvenience of their employment. There was also a dangerously exceeding reliance on shared non-organic resources whose service was not only short and intermittent, but also uncertain due to rising emergencies in neighboring regions. Furthermore, even if there was a detection of a potential threat by an ISR/FMV asset, there was little or no chance of performing cross-cueing for identification purposes. The analytical framework developed in this study could have helped reduce the reliance on non-organic assets by promoting the efficient utilization of small UAVs in two ways: (1) omitting the risk of losing the aerial vehicle due to crosswinds or lack of electromagnetic signals by specifying feasible paths using constraint analysis; and (2) mitigating the *soda straw* problem (i.e. finding threats by scanning large areas using a small FOV) by drastically narrowing the search to a select of hotspots.

Second, the way that the approximately 200 insurgents snuck into the whereabouts of the COP with their heavy ammunition shows their shrewdness and mastery of exploiting environmental aspects. Even with a mediocre situational awareness, it would be hard not to notice the approach of such a large number of well-equipped individuals (relatively to the village population size). This is one more reason why surveillance has to be conducted intelligently rather than exhaustively. In this case, the analytical

tool could have assisted with rendering this practice possible by evaluating the terrain characteristics and providing the list of places where the adversary was likely to pass by (i.e. hotspots).

Third, the Battle of Wanat was harsh on the coalition forces during the first hour when the number of soldiers was much smaller than the opponent's. Once the reinforcements arrived, the drift of the conflict changed to the disadvantage of the AAF. This means that if the insurgents' presence was detected one hour before the attack begun, there would have been high chances of omitting this first hour of intense hostility, if not eliminating the attack itself (by keeping the adversary from reaching the outpost). Note that on a rough low-trafficability terrain—which is the case for the Wanat region, it is assumed that dismounts need 30 minutes to cross one kilometer. It is also assumed that the insurgents need about one hour to position themselves, set up their gears and weapons, and communicate any last-minute tactics among each other before the attack starts. Hence, detection and identification has to occur at most one hour after the adversary penetrates the two-kilometer radius from the COP.

2.1. Input Data

This analysis targets the 5x5 km^2 region that surrounds the Wanat village. The terrain is digitized into a 50x50 grid thanks to the accuracy level of the provided input data. The COP, OP, and village are assumed to be in cells (26,19), (26,20), (27,19), and (27,20), which are highlighted in green in Figure 70. The given input data is under the format of 50x50 matrices where each matrix represents a specific characteristic of the geological environment. The set of these characteristics includes the average, maximum, and minimum terrain altitudes in ASL meters (see Figure 71 for data visualization). The data also comprises the average slope (in degrees) and wind-level (in m/s) matrices, both of which are visualized in Figure 72 and Figure 73, respectively. Furthermore, each cell in the grid is bestowed upon three binary specifications: (1) whether or not it belongs to the outpost, (2) whether or not it has an anchor for the aerostat, and (3) and whether or not it has a base for tower cameras. In this case, the

only cells that are assumed to have tower and aerostat anchors are ones that belong to the outpost. As for population data, the only reported inhabitant concentration is in the Wanat village, which is considered in this study to be part of the COP for two main reasons: (1) the villagers need protection as the village has the potential to turn into a battle field, and (2) if insurgents infiltrate the village, it becomes extremely challenging to rid of them.

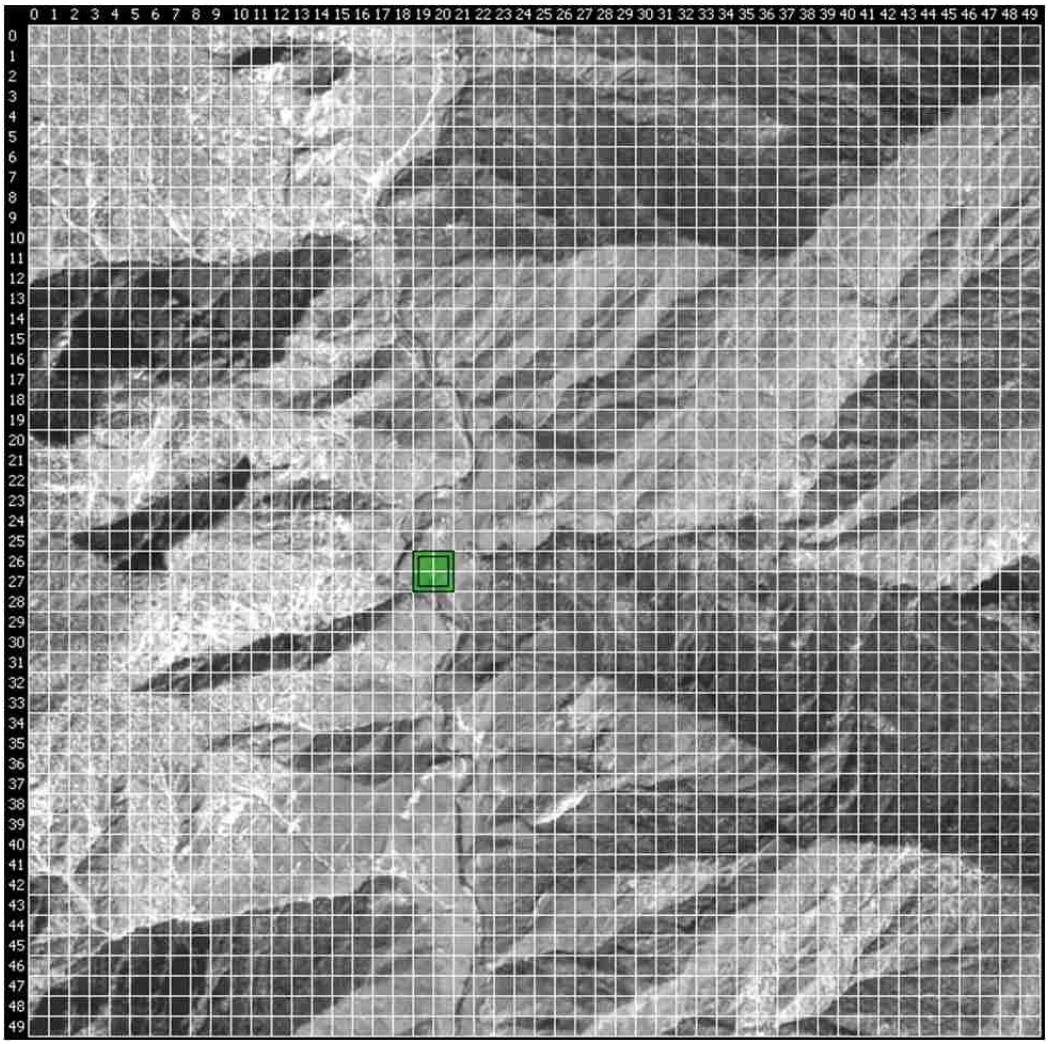

Figure 70: The 50x50 Digitized terrain of the Wanat region

Note that trafficability data is not given, so it is assumed that all cells have poor trafficability t=1 (i.e. there is no unfair advantage or disadvantage amidst the grid cells). The foliage data is also not given, but it is well known that the area around Wanat has a decent amount of foliage. Hence, the

foliage attribute for each cell is randomly generated via the uniform distribution function whose range is [1.5,2.5]. The only exception to this number generation process is the outpost cells, which are given a foliage score of 1, because the village and the COP have a fairly cleared surface that comprises buildings and a terrace, as can be seen from Figure 69.

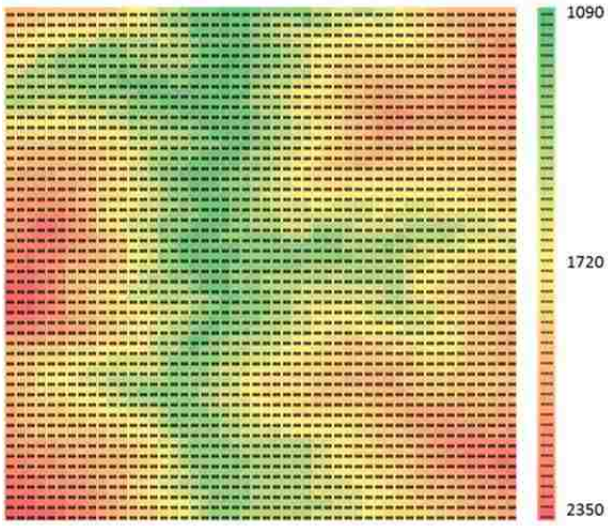

Figure 71: Visualization of average altitude data (m)

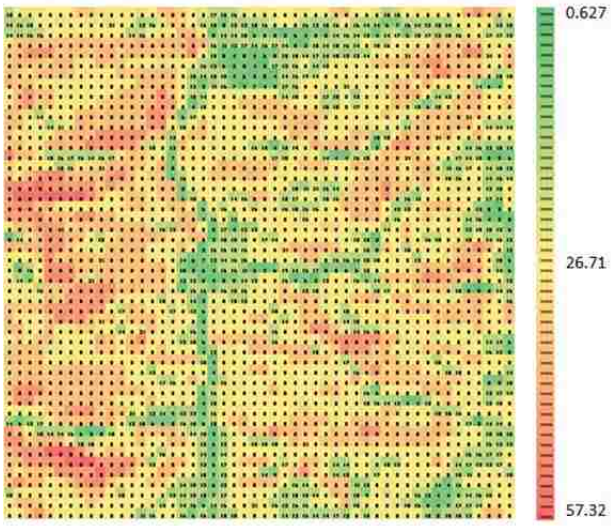

Figure 72: Visualization of slope data (deg)

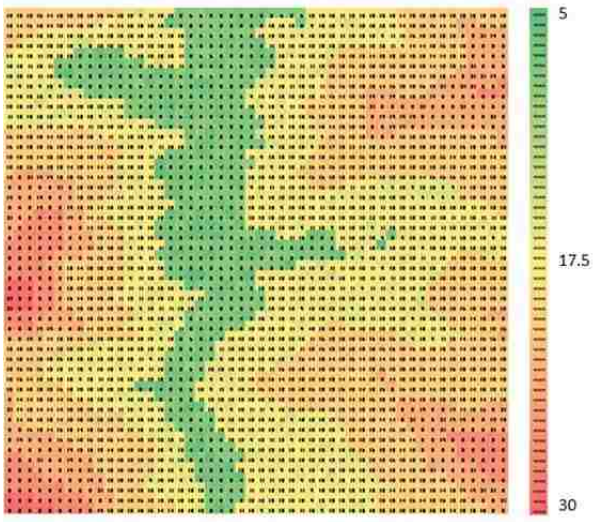

Figure 73: Visualization of wind-levels data (m/s)

For non-matrix parameters, consider the following aforementioned measures along with their values and the risk factor to which they contribute:

Table 10: Parameter values for the Wanat case study

Parameter	Symbol	Risk-Factor	Value
Maximum foliage	f_{max}	Foliage camouflage	3
Ideal elevation for attack	h_o	Long-range attacks	150m
Maximum effective elevation	h_{max}	Long-range attacks	300m
Weapon's maximum range	R	Long-range attacks	3000m
Minimum slope that hinders displacement	β_o	Accessibility to regions	45deg
Maximum slope that hinders displacement	β_{max}	Accessibility to regions	60deg
Slope threshold to consider trafficability	β_{min}	Accessibility to regions	30deg
Trafficability range	n/a	Accessibility to regions	[0,2]

As for the platforms and sensors, the data set used is the same one given in Chapter 2 Section 1.

2.2. Terrain Risk Assessment

The given data is fed into the analytical framework and the terrain risk assessment phase is performed as explained in the Chapter 4. As a result, the risk-factors are generated under a format of matrices; see their associated data visualization in Figure 74 through Figure 77.

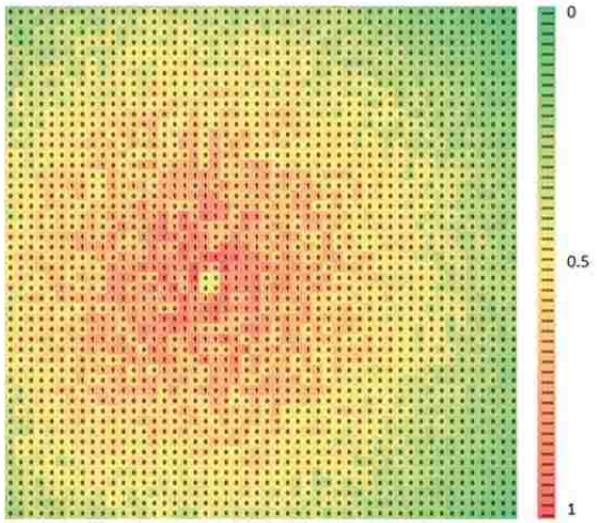

Figure 74: Visualization of the risk-factor associated with foliage camouflage

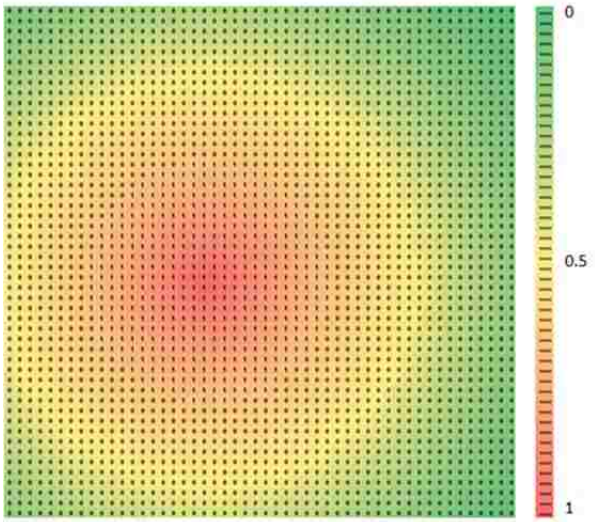

Figure 75: Visualization of the risk-factor associated with proximity to population

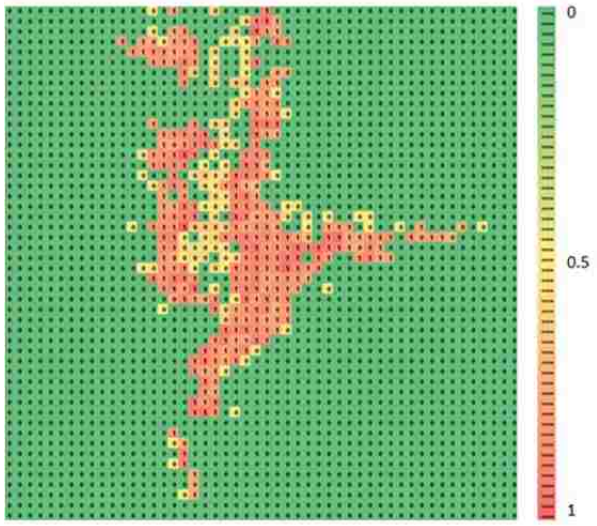

Figure 76: Visualization of the risk-factor associated with long-range attacks

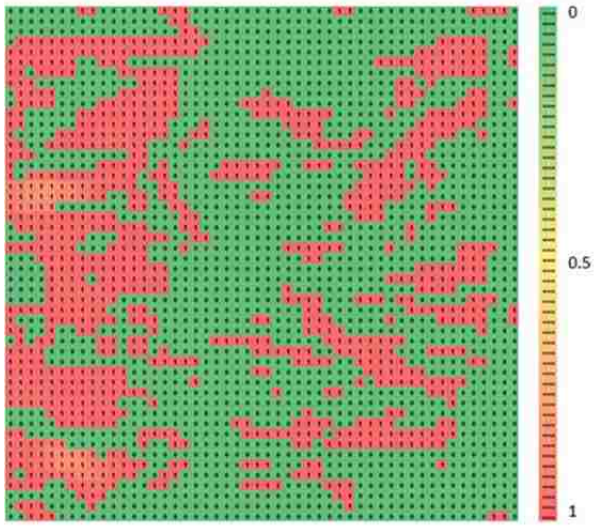

Figure 77: Visualization of the risk-factor associated with accessibility to regions

The framework can also incorporate a fifth optional matrix, as noted in Figure 31, based on another element—the experts input, which is mostly based on historical data and experience in the field. This addition is mainly to adjust the final composite map in case the terrain evaluation process does not capture a subset of the cells that the analyst believes have a high potential for insurgents' presence. In this case study, no such data is provided, and this fifth element is disregarded.

The next item to address is the generation of the composite map and the identification of hotspots. Such a task depends on: (1) the weights associated with each risk-factor—and in this case, three profiles are explored; and (2) the selection process of high-risk cells, whose analysis is a threshold-based one. The first profile assumes no difference in importance between the risk-factors; hence all four of them have a weight of 0.25. The second profile targets places from which long-range attacks can be launched; subsequently, emphasis is placed primarily on the LRA risk-factor (with a weight of 0.5) and secondarily on the accessibility factor (with a weight of 0.25) since the LRA hotspots need to be reached punctually. The third profile focuses on the early stealthy movement of the adversary to reach the outpost and key positions in a smooth and unpredictable fashion. The aspect for earliness in this context is essential as the surveillance aims at detecting the threat as soon as possible. This is modeled by

assigning 50% of the weight to the accessibility risk-factor, 25% to the LRA factor, and 12.5% to both the foliage camouflage and proximity to population factors. Notice that the foliage camouflage risk-factor is deemphasized in this profile despite its potential contribution to the stealthy displacement of insurgents. This is due to the fact that the corresponding matrix risk-scores become greater when the proximity to the outpost increases (as seen in Figure 74), diluting by that the earliness effect. For this same reason, a profile where proximity to population is stressed is not explored. The composite maps for these three profiles are depicted in Figure 78 through Figure 80.

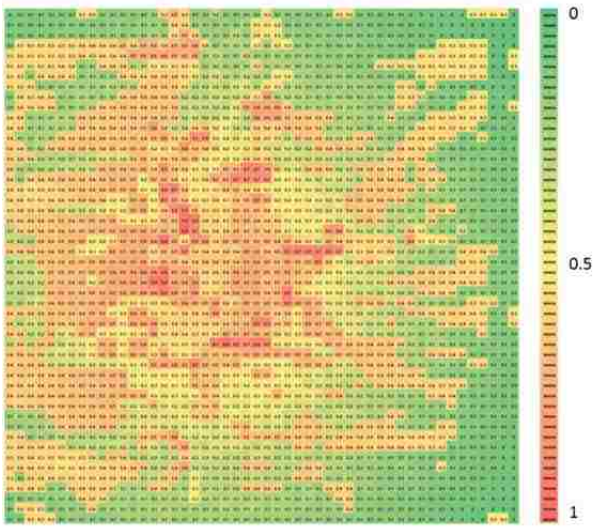

Figure 78: Profile 1's composite map

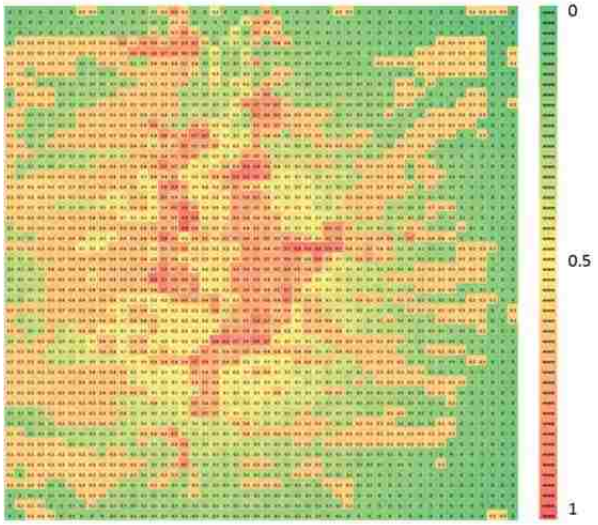

Figure 79: Profile 2's composite map

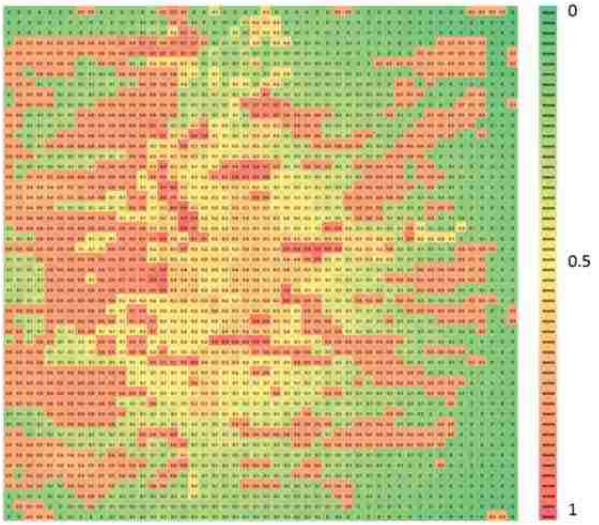

Figure 80: Profile 3's composite map

Profile 1: (w_{FC}, w_{PTP}, w_{LRA}, w_A, w_{EI}) = (0.25, 0.25, 0.25, 0.25, 0)
Description: this profile is unbiased; i.e. it does not discriminate amongst the risk-factors.

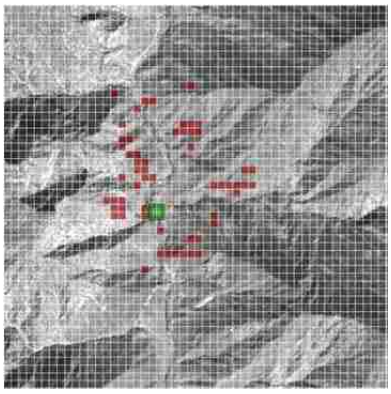

Figure 81: Light surveillance with risk>0.6 yields 57 hotspots

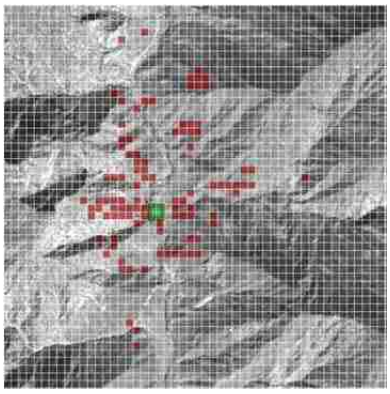

Figure 82: Moderate surveillance with risk>0.55 yields 99 hotspots

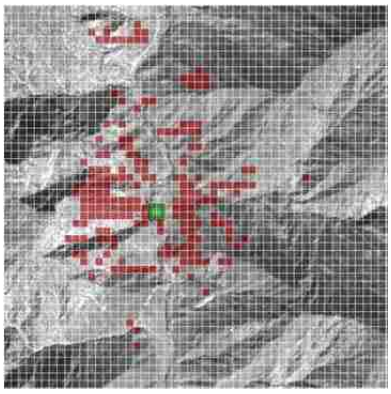

Figure 83: Heavy surveillance with risk>0.5 yields 195 hotspots

Profile 2: (w_{FC}, w_{PTP}, w_{LRA}, w_A, w_{EI}) = (0.125, 0.125, 0.5, 0.25, 0)
Description: this profile targets places from which long-range attacks can be launched.

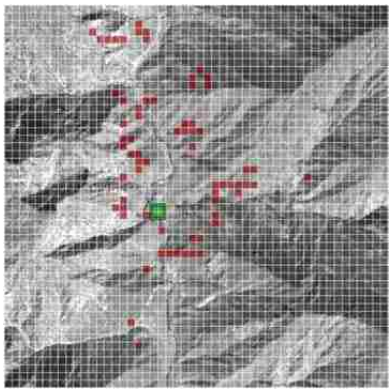

Figure 84: Light surveillance with risk>0.6 yields 68 hotspots

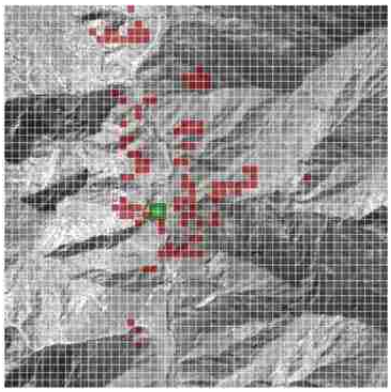

Figure 85: Moderate surveillance with risk>0.55 yields 107 hotspots

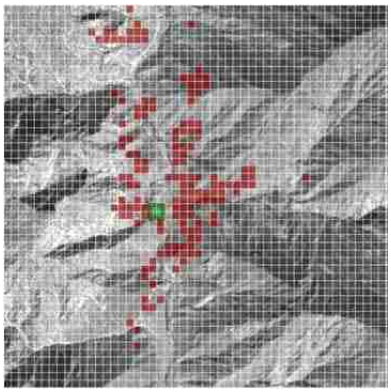

Figure 86: Heavy surveillance with risk>0.5 yields 150 hotspots

Profile 3: (w_{FC}, w_{PTP}, w_{LRA}, w_A, w_{EI}) = (0.125, 0.125, 0.25, 0.5, 0)
Description: this profile targets locations that are unpredictable and lead to strategic positions.

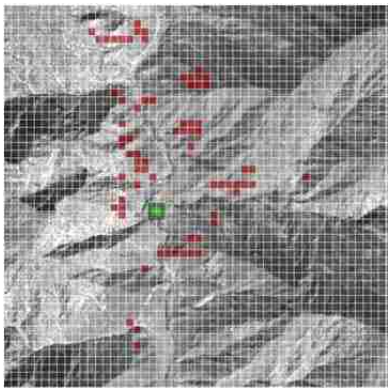

Figure 87: Light surveillance with risk>0.7 yields 72 hotspots

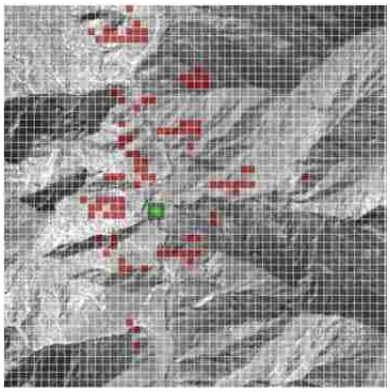

Figure 88: Moderate surveillance with risk>0.65 yields 99 hotspots

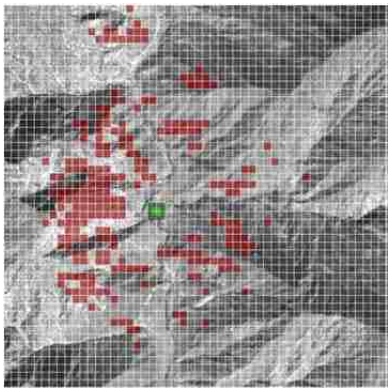

Figure 89: Heavy surveillance with risk>0.6 yields 242 hotspots

As for identifying hotspots, the process is performed by verifying if the final risk-score of each cell—which is a number that belongs to the range [0,1]—is higher than a certain threshold. This selection criterion is relative to the number of identified high-rick cells at a certain level of ISR. To be more specific, three levels of surveillance are considered:

- *Light*: this level corresponds to any threshold that yields a number of hotspots that ranges from 1 to 75.
- *Moderate*: this level corresponds to any threshold that yields a number of hotspots that ranges from 76 to 125.
- *Heavy*: this level corresponds to any threshold that yields a number of hotspots that is greater than 125.

Note: in this study, extra effort is spent to make the second threshold as the mid-point between the other two in order to make the generation of these values more systematic.

Finally, the high-risk cells identification is carried out for all nine combinations, and its outcome is revealed in Figure 81 through Figure 89. The visualization of hotspots on the map helps evaluate the potential presence of threats across the terrain as well as probe the output of the analytical tool. In this case, it is known that the insurgents approached the Wanat village from the west, a fact that prompts the analyst to favor the first and third profiles as Figure 83 and Figure 89 show high concentration of high-risk cells in the western side.

In the rest of this chapter, the heavy-surveillance level of the first profile is chosen for the deeper investigation associated with tour generation and resource allocation due to its larger scale and its more balanced hotspots distribution across all four directions.

2.3. Tour Generation

The tour generation analysis is performed on three UAVs: (1) the Raven (see Figure 90), (2) the shadow (see Figure 93), and the ScanEagle (see Figure 96). Note that because each of these platforms has different capabilities, the feasible space reached by each one is different. This is depicted by a net of thin black lines that represent the UAV-traversable edges, as shown in Figure 91, Figure 94, and Figure 97.

The 3D aspect is portrayed through plotting the optimized operating altitude (also referred to as ASLOA) along with the above-see-level elevation (ASLE), as demonstrated in Figure 92, Figure 95, and Figure 98. The optimization of such altitude is based on giving higher priority to *image quality* and *change in flight altitude* over *stealth* through the weights 0.4, 0.4, and 0.2, respectively. The rationale behind such a decision is the fact that good resolution is critical to detecting and identifying the adversary, and that this task has to be carried out in an easy-to-use and energy-efficient fashion whenever possible. Note that under this configuration, the optimization model is expected to provide a solution in which the UAV tries to stay as close to the ground as possible in order to collect best quality images. The figures below show results that are consistent with this fact. The illustrations are also accompanied by the framework's output data so as to better convey how the flight tours are carried out and show how the three platforms differ in operations.

2.3.1. Raven

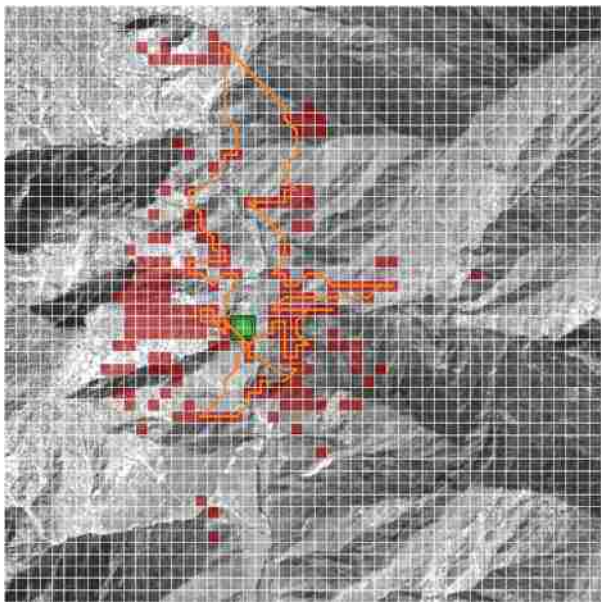

Figure 90: Raven's feasible tour to visit the reachable hotspots

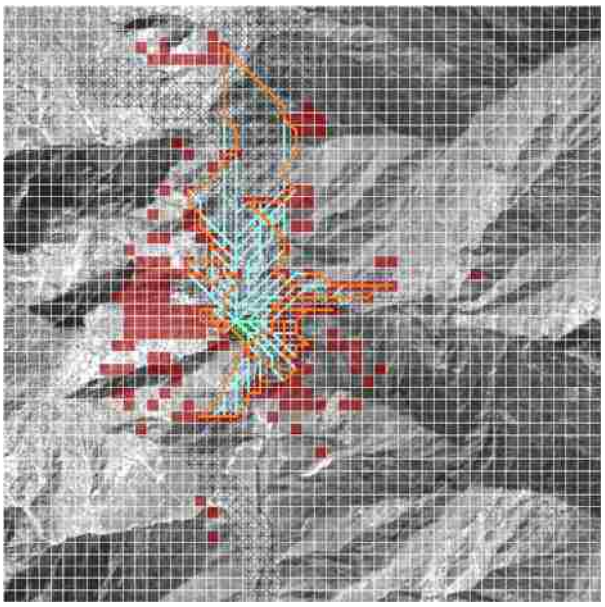

Figure 91: Raven's feasible tour paths (orange), links (cyan), and other traversable edges (black)

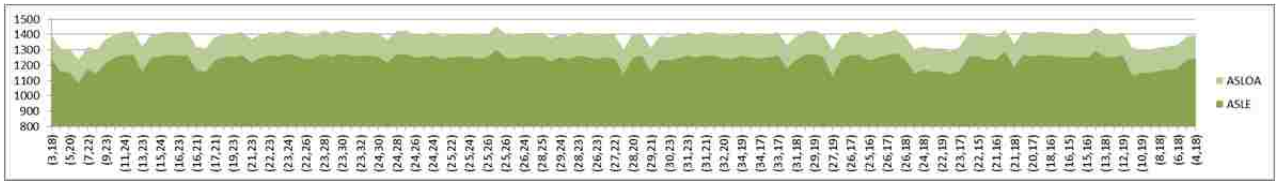

Figure 92: Raven's OA and the terrain elevation (ASL) based on 127 datapoints (in meters)

- Visited hotspots: 76
- Unreachable hotspots: 119
- Tours generated: 1 (no subtours)
- Tour length: 14,291.17m
- Travel time: 793.95s = 13.23min
- Inspection time: 1205.79s = 20.09min
- Flight time: 1999.75s = 33.33min
- Downtime: 300s = 5min
- Cycle time: 2299.75 = 38.33min

2.3.2. Shadow

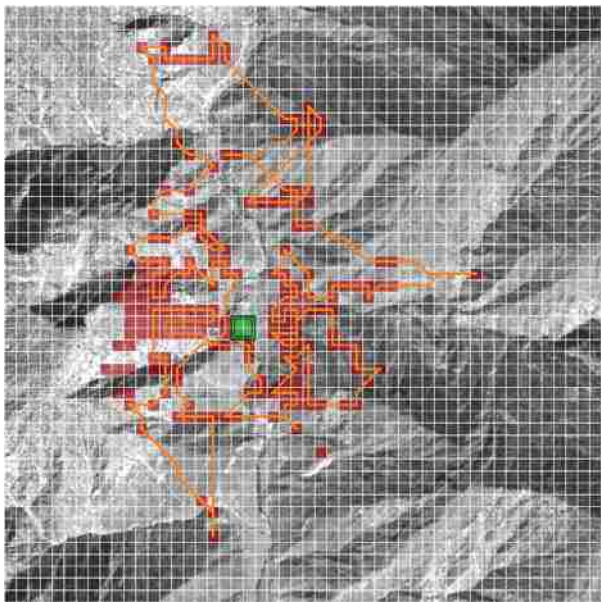

Figure 93: Shadow's feasible tour to visit the reachable hotspots

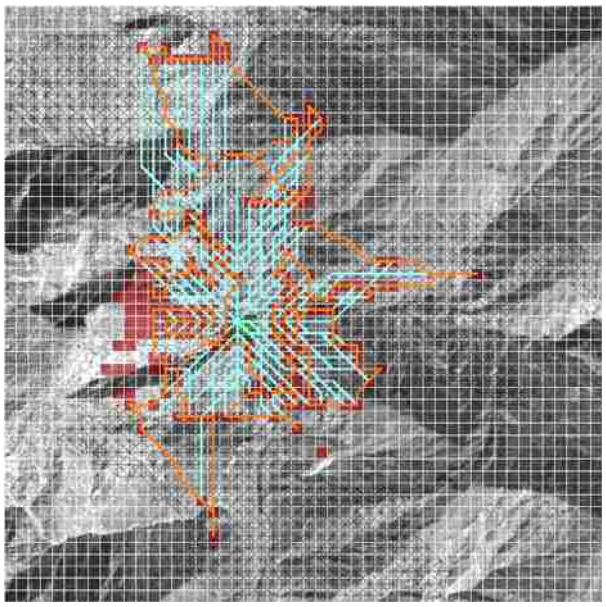

Figure 94: Shadow's feasible tour paths (orange), links (cyan), and other traversable edges (black)

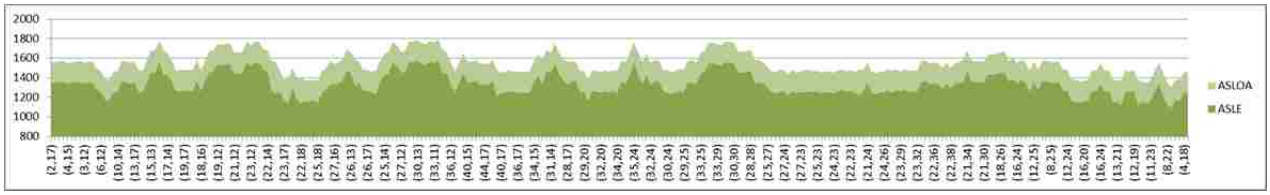

Figure 95: Shadow's OA and the terrain elevation (ASL) based on 274 datapoints (in meters)

- Visited hotspots: 171
- Unreachable hotspots: 24
- Tours generated: 1 (no subtours)
- Tour length: 30,523.76m
- Travel time: 763.09s = 12.72min
- Inspection time: 769.62s = 12.83min
- Flight time: 1532.72s = 25.55min
- Downtime: 3600s = 60min
- Cycle time: 5132.72 = 85.55min

2.3.3. ScanEagle

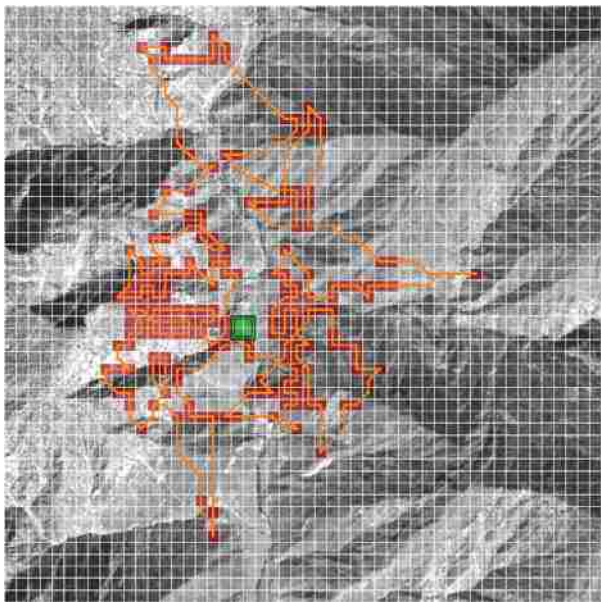

Figure 96: ScanEagle's feasible tour to visit the reachable hotspots

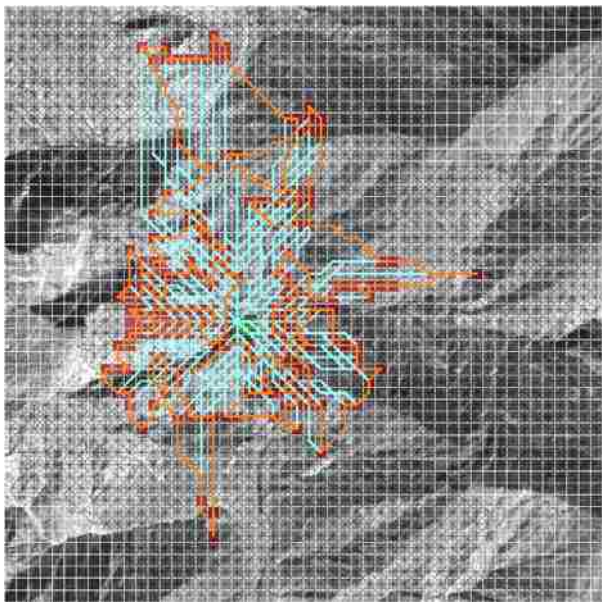

Figure 97: ScanEagle's feasible tour paths (orange), links (cyan), and other traversable edges (black)

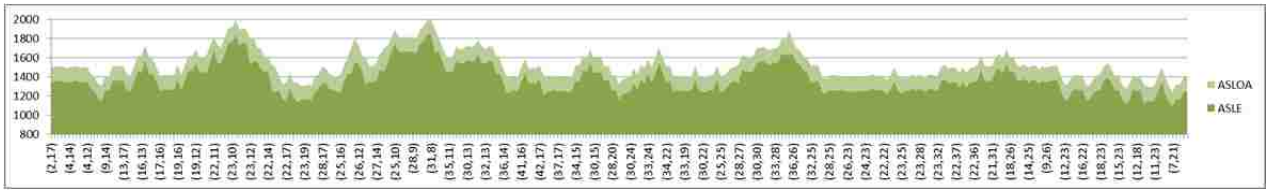

Figure 98: ScanEagle's OA and the terrain elevation (ASL) based on 315 datapoints (in meters)

- Visited hotspots: 193
- Unreachable hotspots: 2
- Tours generated: 1 (no subtours)
- Tour length: 34830.87m
- Travel time: 1393.23s = 23.22min
- Inspection time: 1407.5s = 23.46min
- Flight time: 2800.73s = 46.68min
- Downtime: 900s = 15min
- Cycle time: 3700.73 = 61.68min

2.3.4. Remarks

In this case study, the tour segmentation process returns no subtours. All three UAVs are assigned one complete tour each. From the perspective of the forthcoming resource allocation process, this means two things. First, it signifies that the coefficients matrix of TBSCP has three columns only, which renders solving the problem easy. Second, it implies that there is no need to solve TBVAP as there are no independent subtours to be operated concurrently.

2.4. Resource Allocation

2.4.1. Tactical Systems

To execute the resource allocation procedure for tactical assets, revisit-times are needed for each high-risk cell. In this study, these numbers are attained by a simple systematic conversion of RTM risk-scores that is based on segregating the set of hotspots into two sets: (1) the *first-degree* high-risk cells whose risk-score exceeds 0.6 (depicted in Figure 81), and (2) the *second-degree* high-risk cells, which constitute the remaining of the hotspots (i.e. their risk-scores are within the range [0.6, 0.5]). The former group is assigned a revisit-time of 17 minutes (1012s) whereas the latter is given a value of 27 minutes (1620s). The target function τ_j can be written in terms of the risk-score s_j (for a certain hotspot j) as follows:

$$\tau_j = \begin{cases} 1012s & \text{if } 0.6 \leq s_j \\ 1620s & \text{if } 0.5 \leq s_j \leq 0.6 \\ 0s & \text{otherwise} \end{cases}$$

Note: the rationale behind the 17-minute revisit-time is that it is the sum of half the time needed to cross one kilometer of rough terrain (i.e. 15 minutes, as derived from the assumption established in the beginning of Section 2) and a two-minute leeway. On the other hand, the 27-minute revisit-time reflects

the difference in importance between first-degree and second degree high-risk cells through a 10-minute gap.

After obtaining the flight cycle-times from the tour generation process, and computing the revisit-times using the above function, the TBSCP model is populated and solved. The solution proposes the utilization of four ScanEagles to service the same 193-node tour in a concurrent but asynchronous fashion, with a 15-minute revisit-time for all reachable hotspots. This solution assumes that each vehicle has to go through a 15-minute downtime after finishing a tour. If such a restriction is omitted, the framework gives an alternative that entails employing three ScanEagles only, with a 16.33-minute revisit-time for all reachable hotspots. Note that the removal of such assumption in this particular case is not only safe, but it actually yields a more efficient way to perform surveillance, and that is mainly due to the ScanEagle's 24-hour long endurance. In other words, this UAV can be operated continuously until it needs refueling and servicing, which would happen once a day rather than after every tour.

Another important and appealing feature of the three-ScanEagle solution is that the surveillance schedule is insignificantly affected when one of the three UAVs becomes unavailable. Such an inconvenience can occur for many reasons. It may happen that the UAV has to urgently carry out an identification mission that is triggered by a cross-cueing operation. In addition, the vehicle will eventually require refueling, preventive maintenance, and even unpredicted repair from time to time. It is important to note though that there has to be a change in operation in order to preserve the effectiveness of the surveillance plan during the period of unavailability; this is achieved by switching from the *original configuration*—which is based on visiting all 193 reachable hotspots every 16.33 minutes using all three UAVs concurrently—to the *contingency configuration*. This contingency alternative entails dedicating one UAV to the first-degree high-risk cells only and allocating the other UAV to the remaining hotspots. As a result, the former set of high-risk cells is visited about every 15 minutes, while the latter set is revisited approximately every 30 minutes.

To assess this solution from another perspective, some of the performance measures that can be derived from the provided input data are as follows:

- Manning requirement: 3 at any given moment (one operator per UAV)
- Average slope coverage for scanned hotspots: 100%
- Average slope coverage for all visited cells: 100%

Looking back at the input data, the ScanEagle UAV seems to have had high chances of being selected by a certain analysis or analytical tool as the mobile surveillance asset of choice for COP Kahler (and potentially for other COPs/OPs). When compared to its tactical counterparts, it is clear that it enjoys the longest endurance by far, the widest range of communication, the greatest maximum operating altitude, and the highest resistance to crosswinds. These capabilities allow the vehicle to confront the set of limitations caused by weather and terrain in a much competent fashion. Furthermore, this UAV is one of the two most silent machines and can fly as low as 60 meters, making it ideal for stealth and target identification. Another advantage is the unnecessity of a runaway as the vehicle can be catapulted in the air. The fact that the developed framework picked this asset serves as a good validator of its constraint analysis and optimization structure.

2.4.2. Stationary Systems

Let it be assumed that the analyst is targeting a virtual wall around the center of the outpost with the following properties:

- an outer square that has a side length of 200m
- a thickness of 50 meters with surveillance coverage at the dismount-identification level (i.e. the sensor's FOV is a disk with a radius r_{ID} = 40m as mentioned in Chapter 6 Section 2.3.1)
- a 20% allowance for lack of coverage

The framework gives two solution alternatives, the first of which is based purely on tower-cameras and is referred to as A1. A1 requires the deployment of eight sensors as demonstrated in Figure 99. Such a solution is low-cost and provides full coverage at the dismount-detection level within a 320m-radius.

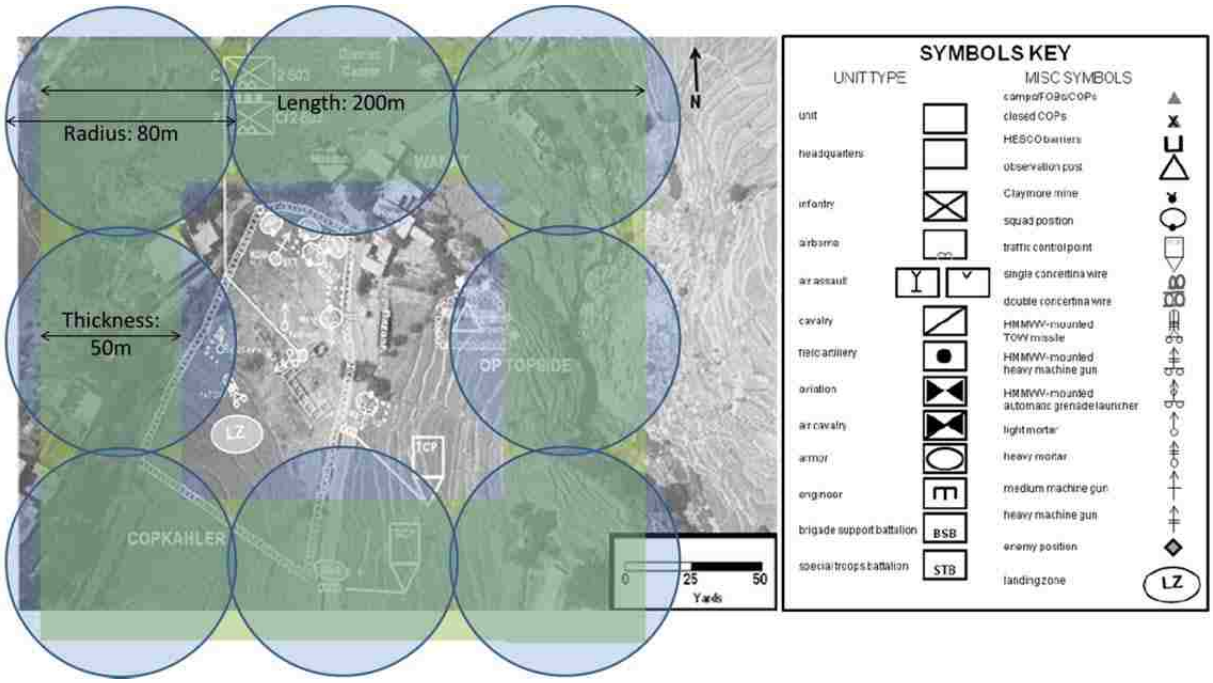

Figure 99: The layout of sensor FOVs (in blue) on the 50-meter-thick virtual wall (in green)
Source of original picture (without the blue and green overlays): SUSACS (2010)

The second alternative, referred to as A2, is an expensive but powerful solution that involves the deployment of one aerostat that is tethered at the center of the outpost. This setup yields coverage radii r_{ID} = 250m, r_{DD} = 1000m, r_{DV} = 4000m for the dismount-identification, dismount-detection, and vehicle-detection levels, respectively.

3. Conclusion

Running the given 5x5km^2 data set for the Wanat region—the past location of COP Kahler—through the developed analytical framework recommends using a combination of tactical vehicles and stationary sensors.

The tactical solution proposes the use of three ScanEagles in a couple of settings:

- The original configuration, a surveillance schedule that requires all three assets to fly concurrently at an adequate distance from each other, such that each of the reachable identified high-risk cells is visited every 16.33 minutes.
- The contingency configuration, which is only used when one of the three assets is unavailable, has a schedule that necessitates that one ScanEagle is dedicated to the subset of high-risk cells whose score exceeds 0.6, while the other ScanEagle monitors the rest of the hotspots. This results in revisit-times of about 15 and 30 minutes for the former and latter subsets, respectively.

The framework also gives two alternatives for the layout of stationary assets. The first alternative involves the utilization of eight tower-cameras placed around the center of the outpost in a squared manner at approximately 80 meters from each other. This low-cost surrogate has reliable coverage of the COP's nearby whereabouts, but has the disadvantage of being ineffective against sniper threats. The second alternative on the other hand is based solely on using an aerostat, a solution that provides much wider coverage at both the detection and identification levels, but might not be justified due to its high cost.

This recommended surveillance plan is constructed in a way to help detect the presence of threats within a two-kilometer radius from COP Kahler at least one hour before the insurgents' attack begins. According to our understanding of the historical facts extracted from the SUSACS (2010) study, the surveillance solution would have prevented the killing and wounding of the U.S. and Afghan soldiers as the reliable enforcement from the neighboring FOB would have arrived before the conflict started.

Finally, it is worth mentioning that this analytical tool is designed to provide a flexible means to analyze complex terrain and provide an adequate solution that meets the different constraints stemmed from the surrounding environment. In general, adding a new data set to the framework (e.g. a newly considered platform, a more recent trafficability matrix, etc) can technically disrupt a currently-obtained solution, a fact that requires performing the analysis with the newly adopted considerations. With this tool however, reconsidering new assets or new environment parameters is as easy as spending few minutes reexecuting the four phases and exploring its output.

4. Dissertation Summary

In the past, the U.S. Army has experienced well-orchestrated attacks that have led to casualties and heavy damage in small combat outposts located in foreign lands where the geo-surface is rough and mountainous. In an effort to reduce or even eliminate this vulnerability, this study assesses the factors that promote risk in such circumstances and addresses an investment problem whose solution specifies the surveillance requirements that can mitigate such risk. To be more specific, this research explores the capabilities of unmanned aerial vehicles and stare-systems to offer a programmatic way that provides a customized, effective, and practical answer to the following question: *given a complex terrain and a smart adversary, what types of surveillance systems, and how many entities of each kind, does a military outpost need to adequately monitor its surrounding environment?*

The addressed problem is difficult to resolve because a mixed portfolio of assets, along with their efficient methods of operation, have to be selected from a myriad of options while considering environmental, technological, and threat aspects. It is certainly extremely challenging, if not impossible, to model and solve as one entity. Although a review of the literature shows that surveillance planning and counter-insurgency have received some attention, no research work is found to be directly applicable to this very problem. Most methods and frameworks that evaluate the geo-surface are

specific to their subject matter, and have as a result limited applicability. In addition, there is a lack of integrated optimization models that take the terrain analysis as an input and produce flight plans and resource allocation solutions as an output. Consequently, the approach followed in this effort is based on constructing a four-phased framework that incorporates planetary surface assessment, constraint analysis, optimization, and space visualization. This analytical tool is effective, novel, and unique with regards to its holistic methodology of answering the research question at hand in a realistic and practical fashion, a procedure that requires good understanding of intelligent insurgency and constrained situational awareness. The developed framework overcomes the entailed challenge by integrating a set of powerful analytical concepts and modeling techniques to address a distinct compilation of real-world restraints during the construction of a customized robust surveillance plan—and hence, a portfolio of systems—for a given COP/OP.

In the first phase, a variant of the Risk Terrain Modeling paradigm is employed to assess the geo-surface for potential insurgent presence and extract a spatially distributed output that aids at determining, visualizing, and converting risk to a group of hotspots along with their practical revisit-times. Besides its inherent nature of evaluating risk, RTM is specifically selected amongst other well-established methods as the tool of choice because it exploits the same space representation infrastructure assumed in this analysis—the two-dimensional grid, which is decided upon due to its simplicity and computational efficiency. Nonetheless, two adjustments to RTM are necessary in order to render it suitable for the context of this study: (1) an extra step is added so as to convert raw weather and terrain attributes to meaningful risk-factors; and (2) risk-scores are allowed to hold any value in the range [0,1] (as opposed to 0 and 1 only) in order to avoid the underestimation of accumulated risk from such factors. The output of this method is the composite map—a matrix whose entries coincide with the 2D grid of the terrain map—in which high-risk grid cells are identified and their corresponding revisit-time is computed.

In the second phase, hotspot-comprising tours and subtours are generated for each considered UAV in an exceptionally feasible manner—using Mathematical Programing with heuristic approaches—in order to adhere to several technological limitations and realistic geological restraints. That is achieved by first ensuring through a process of constraint satisfaction and cell elimination that the digitized terrain infrastructure contains reachable cells only. Next, an efficient sequence of visits to the hotspots identified in the previous phase is acquired by solving the Traveling Salesman Problem. Also, for each consecutive elements of this sequence of cells, the Shortest Path Problem is solved to find the best feasible cell-to-cell route. The MP models associated with these two optimization problems are adopted without adjustment because the reduced network on which they are operating is already constraint-compliant. Conversely, the set of heights at which each UAV is to travel when traversing such paths are specified using a custom-made highly constrained linear program—which is referred to as the Operating Altitude Problem model—due to its context specificity and uniqueness. After obtaining the 3D tours using these aforementioned steps, each tour is checked against its associated UAV's endurance; tours that violate this constraint are segmented into feasible subtours. The set of restraints modeled in this phase pertain to wireless communication range, terrain elevation, wind level, as well as the aerial vehicle's min/max operating altitude, climb-/dive-rates, noise (dB-)distance, captured image resolution, and endurance.

In the third phase, constrained resource allocation is performed on tactical vehicles via Mixed Integer Programming to select the best combination of trajectories and their servicing UAVs, whereas sensor-ground geometry is exploited to construct a lean arrangement of stationary surveillance systems. To be more specific, a variant of the Set Covering Problem is employed to find the best combination of tour and subtours to visit all reachable hotspots in a manner that meets the revisit-times specified in the first phase. To achieve the temporal aspect, the original SCP formulation—which already offers a simple and efficient way to generally pick a subset of options that satisfies the coverage demand—is expanded

to account for the necessary revisit-frequency through one closed-form constraint. Yet, the obtained SCP solution may sometimes be improvable by reducing the number of UAVs needed to perform the selected subtours. This is effectually accomplished via a time-based modification of the classical Assignment Problem whereby two closed-form constraints are added to the involved model in order to keep the revisit-time restraint from being violated. In these optimization cases, Mixed Integer Programming proves very useful in enforcing the refueling, scheduling, and revisit-frequency restraints, which renders the obtained solution easily scalable through repetition to obtain the 24-hour flight schedule. On the other hand, a layout of stare-systems is generated in order to compensate for the intermittency of the UAVs' service. The placement of stationary systems is performed by an ad hoc geometry-based algorithm that minimizes unnecessary overlap in an effort to reduce the cost of the proposed solution. This computational procedure enforces the asset protection and other geometric constraints by relying on a special infrastructure called the virtual wall, which specifies the area around the outpost that is to be covered by FOV projections. Such an algorithm is very fast since it estimates the position coordinates of each sensor using mathematical formulas as opposed to seeking a vector of variable values using a solver or heuristic. After the tactical and stationary aspects of the resource allocation are addressed, the construction process is terminated and the acquired surveillance plan finally reveals the portfolio of assets to be purchased.

The fourth phase does not solve an analytical problem per se; rather, it provides multiple time- and geometry-based measures of effectiveness. The statistics considered in this study pertain to the required manning, surface coverage, slope coverage, captured image quality, tour lengths, travel times, inspection times, flight times, and others. The surveillance plan evaluation also reveals, if any, the set of unreachable hotspots as well as the set of cells where the revisit-time is much higher than required. This not only helps appraise the proposed surveillance plan and identify any vulnerabilities, but it also assists the analyst with any necessary massaging of the final solution for the purpose of adding the human

touch to it and considering any exogenous issues such as contingency responses to asset loss or unavailability.

This ordered comprehensiveness that is associated with risk-based planetary surface evaluation, constrained flight planning, restrained resource allocation, and surveillance plan assessment, is what highlights the contribution of this study, and is what helps the surveillance architect answer the research question without reaching out to other complementary methods.

REFERENCES

[1] Air Combat Command (2007, March 28). *Tethered Aerostat Radar System*. Retrieved August 7, 2010, from http://www.acc.af.mil/library/factsheets/factsheet.asp?id=2359

[2] Ahmadzadeh, A., Keller, J., Pappas, G., Jadbabaie, A., & Kumar, V. (2008). An optimization-based approach to time-critical cooperative surveillance and coverage with UAVs. In O. Khatib, V. Kumar & D. Rus (Eds.), *Experimental robotics, the 10th international symposium on experimental robotics* (pp. 491-500) Springer Berlin / Heidelberg. doi:10.1007/978-3-540-77457-0_46

[3] Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). *Network Flows*, Prentice Hall, Englewood Cliffs, New Jersey.

[4] Army Unmanned Aircraft Systems Center of Excellence. (2010). *US army unmanned aircraft systems roadmap 2010-2035: Eyes of the army.* (Military Report). Defense Technical Information Center.

[5] Balcik, B., Beamon, B. M., & Smilowitz, K. (2008). Last mile distribution in humanitarian relief. *Journal of Intelligent Transportation Systems, 12*(2), 51-63.

[6] Bayraktar, S., Fainekos, G. E., & Pappas, G. J. (2005). Experimental cooperative control of fixed-wing unmanned aerial vehicles. *Decision and Control, 2004. CDC. 43rd IEEE Conference on,* Atlantis, Paradise Island, Bahamas. *4* 4292-4298.

[7] Caplan, J. M., & Kennedy, L. W. (2010). *Risk terrain modeling manual: Theoretical framework and technical steps of spatial risk assessment* Center on Public Security.

[8] Caplan, J. M., Kennedy, L. W., & Miller, J. (2010). Risk terrain modeling: Brokering criminological theory and GIS methods for crime forecasting. *Justice Quarterly*.

[9] Carlyle, W. M., Royset, J. O., & Kevin Wood, R. (2008). Lagrangian relaxation and enumeration for solving constrained shortest-path problems. *Networks*, *52*(4), 256-270.

[10] Carrico, T., & Downer, K. (n.d.). *Commercial air surveillance for sovereign maritime domain awareness.* (Technical Report). AGI.

[11] Chandler, P. R., Pachter, M., Swaroop, D., Fowler, J. M., Howlett, J. K., Rasmussen, S., Schumacher, C., & Nygard, K. (2002). Complexity in UAV cooperative control. *Proceedings of the 2002 American Control Conference,* Anchorage, Alaska, USA. *3* 1831-1836. doi:10.1109/ACC.2002.1023833

[12] Choset, H. (2001). Coverage for robotics–A survey of recent results. *Annals of Mathematics and Artificial Intelligence, 31*(1), 113-126. doi:10.1023/A:1016639210559

[13] Dixon, C. (2009). *COP Keating attack.* Retrieved December 7, 2010, from http://claudesventures.com/?p=1590

[14] Dong, L., Xu-chen, L., Xiang-tao, Y., & Fei, W. (2010). A model for allocating protection resources in military logistics distribution system based on maximal covering problem. *Logistics Systems and Intelligent Management, 2010 International Conference on*, Changsha, China. 1 98-101.

[15] Downs, M. L. (2007). *Rethinking the CFACC's intelligence, surveillance, and reconnaissance approach to counterinsurgency.* (Military Report). Newport, RI: Joint Military Operations Department, Naval War College.

[16] Fok, V. (2005). Supporting airborne cooperative surveillance, engagement and threat-avoidance planning in complex terrain. *SimTect,* Sydney.

[17] Frazzoli, E., Dahleh, M. A., & Feron, E. (1999). A hybrid control architecture for aggressive maneuvering of autonomous helicopters. *Proceedings of the 38th IEEE Conference on Decision and Control,* Phoenix, AZ*., 3* 2471-2476.

[18] Gomez, M., Preece, A., Johnson, M., De Mel, G., Vasconcelos, W., Gibson, C., ... & Pham, T. (2008). An ontology-centric approach to sensor-mission assignment. *Knowledge Engineering: Practice and Patterns*, 347-363.

[19] Grignon, L., Prasetio, Y., Toktas, B., Yen, J., & Zabinsky, Z. (2002). *A basis space-time network formulation for aircraft rerouting with airspace closures.* (Technical Report). Department of Industrial Engineering, University of Washington, Seattle, Washington.

[20] Howard, A., & Seraji, H. (2002). Real-time assessment of terrain traversability for autonomous rover navigation. *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems,* Takamatsu, Japan. *1* 58-63. doi:10.1109/IROS.2000.894582

[21] Jun, M., & D'Andrea, R. (2003). Path planning for unmanned aerial vehicles in uncertain and adversarial environments. *Cooperative Control: Models, Applications and Algorithms* (pp. 95-111). Springer.

[22] Kaplan, E. H., Kress, M., & Szechtman, R. (2010). Confronting entrenched insurgents. *Operations Research, 58*(2), 329-341. doi:10.1287/opre.1090.0728

[23] Kilby, P., Tobin, P., Luscombe, R., Barry, S. I., & Hickson, R. (2007). *The maritime surveillance problem.* (Technical Report No. 128). Defence Science and Technology Organisation.

[24] Kim, J., & Hespanha, J. P. (2003). Discrete approximations to continuous shortest-path: Application to minimum-risk path planning for groups of UAVs. *Proceedings of the 42nd IEEE Conference on Decision and Control,* Maui, HI. *2* 1734-1740.

[25] Klabjan, D., Johnson, E. L., Nemhauser, G. L., Gelman, E., & Ramaswamy, S. (2002). Airline crew scheduling with time windows and plane-count constraints. *Transportation Science*, *36*(3), 337-348.

[26] Lee, J., & Raffensperger, J. F. (2006). Using AMPL for teaching the TSP. *INFORMS Trans. Ed*, *7*(1), 37-69.

[27] Library of Congress Washington DC Congressional Research Service. (2006). *Potential military use of airships and aerostats.* (Report). Defense Technical Information Center.

[28] McCullough, A. (2010, October). Saving outpost Keating. *Air Force Magazine, 93*(10), 38-47.

[29] Mettler, B., Valenti, M., Schouwenaars, T., Kuwata, Y., How, J., Paunicka, J., & Feron, E. (2003). Autonomous UAV guidance build-up: Flight-test demonstration and evaluation plan. *Proc. AIAA Guidance, Navigation, and Control Conference,* Austin, TX.

[30] Miles, D. (2006, December 1). Concept perfected in Iraq, Afghanistan used along U.S. border. *American Forces Press Service*.

[31] Miller, C. E., Tucker, A. W., & Zemlin, R. A. (1960). Integer programming formulations and traveling salesman problems. *J. ACM,* 7, 326-329.

[32] Novy, M. C. (2001). *Air vehicle optimal trajectories for minimization of radar exposure*. (Master's Thesis, Graduate School of Engineering and Management, Air Force Institute of Technology, Wright-Patterson AFB, Ohio).

[33] Peucker, T. K., Fowler, R. J., Little, J. J., & Mark, D. M. (1980). The triangulated irregular network. *Proceedings of the Sixth International Symposium on Automated Cartography (Auto-Carto VI),* Reston, Virginia. 2, 96-103.

[34] Ross, K. (2010, July). What really happened at Wanat. *Proceedings Magazine, 136/7/1,289*.

[35] Royset, J. O., Carlyle, W. M., & Wood, R. K. (2009). Routing military aircraft with a constrained shortest-path algorithm. *Military Operations Research, 14*, 31-52.

[36] Secreat, B. R. (2001). *Traveling salesman problem for surveillance mission using particle swarm optimization*. (Master's Thesis, Graduate School of Engineering and Management, Air Force Institute of Technology, Wright-Patterson AFB, Ohio).

[37] Sheu, J. B., Chen, Y. H., & Lan, L. W. (2005). A novel model for quick response to disaster relief distribution. *Proceedings of the Eastern Asia Society for Transportation Studies,* 5, 2454-2462.

[38] Sidran, D. E. (2007). *Overview of the current state of human-level artificial intelligence in computer.* Retrieved November 21, 2010, from http://www.gilgameshcontrite.com/Computer_AI/

[39] Srikanth, M. B., Mathias, P. C., Natarajan, V., Naidu, P., & Poston, T. (2008). Visibility volumes for interactive path optimization. *The Visual Computer, 24*(7), 635-647. doi:10.1007/s00371-008-0244-x

[40] The Staff of the U.S. Army Combat Studies Institute. (2010). Wanat: combat action in Afghanistan, 2008. *Combat Studies Institute Press*.

[41] The Washington Times. (2008, March 26). BRIEFING: FOBs the closest thing to home in Iraq.

[42] Turbiville, G. H. (2009). *Guerrilla counterintelligence: Insurgent approaches to neutralizing adversary intelligence operations.* (Military Report). Hurlburt Field, FL: Joint Special Operations University.

[43] Vazirani, V. V. (2001). *Algorithm Approximations*. New York: Springer.

[44] Waser, L. T., Ginzler, C., Kuechler, M., Baltsavias, E., & Hurni, L. (2011). Semi-automatic classification of tree species in different forest ecosystems by spectral and geometric variables derived from airborne digital sensor (ADS40) and RC30 data. *Remote Sensing of Environment, 115*(1), 76-85. doi:10.1016/j.rse.2010.08.006

[45] Winkler, R. P. (2006). *The evolution of the joint ATO cycle*. (Master's Thesis, Joint Advanced Warfighting School, 7800 Hampton Blvd, Norfolk, VA).

[46] Zabarankin, M., Uryasev, S., & Pardalos, P. (2002). Optimal risk path algorithms. In R. Murphey, & P. M. Pardalos (Eds.), *Cooperative control and optimization* (pp. 273-298) Springer US. doi:10.1007/0-306-47536-7_13

[47] Zadeh, L. A. (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, *1*(1), 3-28.

APPENDIX

1. AMPL Model for TSP

```
param N integer > 2;                    # Number of nodes
set NODES ordered := {0..N-1};          # set of node indices
set ARCS := {i in NODES, j in NODES: i <> j};  # set of arcs
param c{ARCS};                          # cost of travelling from node i to node j

var x{ARCS} binary;        # 1 if trip from node i to node j is chosen, 0 otherwise
var y{NODES} >= 0;         # the number of so-far visited nodes when at node i

minimize objective:
       sum {(i,j) in ARCS} c[i,j] * x[i,j];

subject to enter_each_node_once {i in NODES}:
       sum{(i,j) in ARCS} x[i,j] = 1;
subject to leave_each_node_once {i in NODES}:
       sum{(j,i) in ARCS} x[j,i] = 1;
subject to no_subtour1 {(i,j) in ARCS: i<>j and i >= 1 and j >= 1}:
       y[i] - y[j] + N * x[i,j] <= N - 1;
subject to no_subtour2 {i in NODES: i >= 1}:
       y[i] + (N - 2) * x[0,i] <= N - 1;
subject to no_subtour3 {i in NODES: i >= 1}:
       y[i] - (N - 2) * x[i,0] >= 1;
```

2. Pseudo-Code for TSP Heuristic

```
sizeThreshold = 25;

function solveDynamically(nodeList, grid)
{
   if(nodeList.size() <= sizeThreshold)
      solution = solveExactly(nodeList); //using a solver (e.g. CPLEX)
   else
   {
      if(grid.columns() > grid.rows())
      {
         (sublis1, sublist2) = divideVertically(nodeList);
         (subgrid1, subgrid2) = divideVertically(grid);
      }
      else
      {
         (sublis1, sublist2) = divideHorizontally(nodeList);
         (subgrid1, subgrid2) = divideHorizontally(grid);
      }

      dummyNode1 = getCentroid(sublist2);
      dummyNode2 = getCentroid(sublist1);

      sublist1.add(dummyNode1);
      sublist2.add(dummyNode2);

      solution1 = solveDynamically(sublist1, subgrid1);
      solution2 = solveDynamically(sublist2, subgrid2);

      solution = join(solution1, solution2); //dummy nodes are removed here
   }
   return solution;
```

```
}
```

3. AMPL Model for SPP

```
set NODES;            # set of nodes
set EDGES;            # set of edges
param c{EDGE};        # cost of travelling through an edge
param b{NODES};       # b[j] = 1 if node j is the origin
                      # b[j] = -1 if node j is the destination
                      # b[j] = 0 otherwise
param a{NODES,EDGES}; # a[j,i] = 1 if node j is the start of edge i
                      # a[j,i] = -1 if node j is the end of edge i
                      # a[j,i] = 0 otherwise

var x{EDGE} binary;   # x[i] = 1 if edge i is selected to be in the path, 0 otherwise

Minimise objective:
       sum{i in EDGES} c[i] * x[i] ;

Subject to enter_each_node_once {j in NODES}:
       Sum{i in EDGES} a[j,i] * x[i] = b[j];
```

4. AMPL Model for OAP

```
set NODES;                # set of nodes visited by UAV through the surveillance tour
set EDGES within (NODES cross NODES);

param elev{NODES} >= 0;    # cell/node's elevation
param dist{EDGES} >= 0;    # edge's length (distance traveled)
param climb_rate >= 0;     # UAV's maximum climb-rate
param dive_rate >= 0;      # UAV's maximum dive-rate
param lb_oa_agl >= 0;      # UAV's operating altitude lower bound (above ground level)
param ub_oa_asl >= 0;      # UAV's operating altitude upper bound (above sea level)
param db_dist_agl >= 0;    # UAV's DB (noise) distance (above ground level)
param w1 >= 0;             # objective 1's weight
param w2 >= 0;             # objective 2's weight
param w3 >= 0;             # objective 3's weight

var oa_agl{NODES} >= 0;    # UAV's operating altitude (above ground level)
var oa_asl{NODES} >=0;     # UAV's operating altitude (above sea level)
var a{NODES} >= 0;         # allowance for violating stealth constraint at a node
var max_oa_change >= 0;    # maximum operating altitude change across all edges
var max_oa_agl >= 0;       # maximum operating altitude across all nodes (AGL)

minimize objective:
       w1 * max_oa_change + w2 * max_oa_agl + w3 * sum{i in NODES} a[i];

subject to
oa_asl_constraint {i in NODES}:
       oa_asl[i] = oa_agl[i] + elev[i];
lb_oa_agl_constraint {i in NODES}:
       oa_agl[i] >= lb_oa_agl;
ub_oa_agl_constraint {i in NODES}:
       oa_asl[i] <= ub_oa_asl;
climb_dive_constraint1 {(i,j) in EDGES}:
       oa_asl[i] - oa_asl[j] <= min(climb_rate, dive_rate) * dist[i,j];
climb_dive_constraint2 {(i,j) in EDGES}:
       -oa_asl[i] + oa_asl[j] <= min(climb_rate, dive_rate) * dist[i,j];

# for objective1 formulation
```

```
max_oa_change_constraint1 {(i,j) in EDGES}:
       oa_asl[i] - oa_asl[j] <= max_oa_change;
max_oa_change_constraint2 {(i,j) in EDGES}:
       -oa_asl[i] + oa_asl[j] <= max_oa_change;

# for objective2 formulation
stealth_constraint {i in NODES}:
       oa_agl[i] + a[i] >= db_dist_agl;

# for objective3 formulation
min_oa_agl_constraint {i in NODES}:
       oa_agl[i] <= max_oa_agl;
```

Example of Input Data:

```
data;

set NODES := 0 1 2 3;
set EDGES := (0,1) (0,2) (0,3);

param dive_rate := 1;
param climb_rate := 1;
param db_dist_agl := 100;
param lb_oa_agl := 50;
param ub_oa_asl := 1500;

param:       elev :=
       0     450
       1     500
       2     400
       3     400;

param:       dist :=
       0 1   50
       0 2   50
       0 3   100;
```

5. Pseudo-Code for the Tour Segmentation Algorithm

```
function segmentTour()
{
   foreach startNode in sortedHotspotsByTSPSequence
   {
       //initiating a subtour
       subtour.add(linkTo(startNode)); //going from outpost to startNode
       subtour.add(linkTo(startNode)); //going from startNode to outpost
       traveledDistance = totalLength(subtour);
       nextNode = startNode;

       while(true)
       {
          currentNode = nextNode;
          nextNode = getNextNode(sortedHotspotsByTSPSequence, currentNode);

          //computing endurance cost of subtour expansion
          distanceChange = -linkTo(currentNode).length()          //- old return link
                         + path(currentNode, nextNode).Length()   //+ new path
                         + linkTo(nextNode).getLength();          //+ new return link

          //expanding subtour if feasible
```

```
            if(enduranceConstraint.isVerified(traveledDistance + distanceChange))
            {
                subtour.remove(linkTo(currentNode));
                subtour.add(path(currentNode, nextNode));
                subtour.add(linkTo(nextNode));
                traveledDistance = traveledDistance + distanceChange;
            }
            else break;
        }
        subtourDatabase.save(startNode, subtour);
    }

    if(subtourDatabase.containsCompleteTour())
        return leastCostTour(subtourDatabase);
    else return allSavedSubtours(subtourDatabase);
}
```

6. AMPL Model for TBSCP

```
param N integer;                 # number of nodes
param M integer;                 # number of subtours
set NODES := {0..N-1};           # set of node indices
set SUBTOURS := {0..M-1};        # set of subtour indices
param c{SUBTOURS};               # cost of selecting a subtour (in distance)
param f{NODES,SUBTOURS};         # visit frequency if tour contains node; 0 otherwise
param t{NODES};                  # required revisit-time of a node

var x{SUBTOURS} integer >= 0;    # needed number of trips for a certain subtour

minimize objective:
       sum{j in SUBTOURS} c[j] * x[j];

subject to revisit_time_constraint {i in NODES}:
       sum{j in SUBTOURS} f[i,j] * x[j] >= 1/t[i];
```

7. AMPL Model for TBVAP

```
param M integer;                 # big-M (a very large number)
param N integer;                 # number of vehicles/subtours
set VEHICLES := {0..N-1};        # set of vehicle indices
set SUBTOURS := {0..N-1};        # set of subtour indices
param d{VEHICLES,SUBTOURS};      # duration that vehicle takes to traverse subtour
param t{SUBTOURS};               # subtour's minimum revisit-time (depends on nodes)

var x{VEHICLES,SUBTOURS} binary; # x[i,j]=1 if vehicle i serves subtour j, otherwise 0
var y{VEHICLES} binary;          # y[i]=1 if vehicle i is to be purchased, otherwise 0

minimize objective:
       sum{i in VEHICLES} y[i];

subject to cover_all_subtours {j in SUBTOURS}:
       sum{i in VEHICLES} x[i,j] = 1;
subject to revisit_time_constraint {i in VEHICLES}:
       sum{j in SUBTOURS} d[i,j] * x[i,j] <= min{j in SUBTOURS} t[j];
subject to x_y_relation {i in VEHICLES}:
       sum{j in SUBTOURS} x[i,j] <= M * y[i];
```

Example of Input Data:

```
data;

param M :=   1000;
param N :=   5;

param d := [*,*] :
              0      1      2      3      4 :=
       0      1.25   M      1      1.1    0.75
       1      1.25   1.75   1      1.1    0.75
       2      M      M      1      M      0.75
       3      1.25   M      1      1.1    0.75
       4      M      M      1      M      0.75;

param t :=    0      2.5
              1      2.5
              2      2.5
              3      2.5
              4      2.5;
```

8. Pseudo-Code for the SSPP Algorithm

```
function placeSensors(virtualWall)
{
    layersList = divideIntoLayers(virtualWall); // this may return one layer only
    foreach(segment in virtualWall.segmentsList)
        segment.computeAllowance();
    foreach(layer in layersList)
        foreach(segment in layer.segmentsList)
            segment.computeAllowance();
    foreach(layer in layersList)
    {
        foreach(corner in layer.cornersList)
            corner.place(new Sensor());
        foreach(segment in layer.segmentsList)
        {
            CC = segment.computeCoverageContribution();
            if(CC < segment.area * (1 - segment.allowance))
            {
                N = ceiling((A * (1 - a) - CC) / (H * (W - 2 * s)));
                for(i = 1; i <= N; i++)
                    layout.placeOnCentroidsLine(new Sensor(), i); // at ith position
            }
        }
    }
    finalize(layout); // recover the original circular shape of the sensor's range
}
```