3-16-2009

Operations-Focused Optimized Theater Weather Sensing Strategies Using Preemptive Binary Integer Programming

Andrew J. Geyer

OPERATIONS-FOCUSED OPTIMIZED THEATER WEATHER SENSING STRATEGIES USING PREEMPTIVE BINARY INTEGER PROGRAMMING

THESIS

Andrew J. Geyer, Captain, USAF

AFIT/GOR/ENS/09-06

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GOR/ENS/09-06

OPERATIONS-FOCUSED OPTIMIZED THEATER WEATHER SENSING STRATEGIES USING PREEMPTIVE BINARY INTEGER PROGRAMMING

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

Andrew J. Geyer, BS

Captain, USAF

March 2009

AFIT/GOR/ENS/09-06

OPERATIONS-FOCUSED OPTIMIZED THEATER WEATHER SENSING STRATEGIES USING PREEMPTIVE BINARY INTEGER PROGRAMMING

Andrew J. Geyer, BS
Captain, USAF

Approved:

Shane N. Hall, Ph.D., Maj, USAF (Co-Advisor) — 16 Mar 09 (date)

James T. Moore, Ph.D., (Co-Advisor) — 16 Mar 09 (date)

AFIT/GOR/ENS/09-06

Abstract

This thesis describes a method that optimally deploys weather sensors of all types in a battlefield environment. Gridded climatology models are used to determine an estimate for the weighted frequency of occurrence of operationally significant inclement weather events. That data is used to formulate a series of preemptive Binary Integer Linear Programs that maximize detection of expected operationally significant inclement weather occurrences within the constraints of feasibility of sensor deployment, sensor operational lifespan and the sensor's ability to detect the operationally significant inclement weather elements. The preemptive Binary Integer Linear Programs are combined into a single objective function that maintains the preemptive nature of the original objective functions. The BILP solutions are described as a meteorology and oceanographic collection plan supporting a particular military campaign. A method for sensitivity analysis of differing BILP optimal solutions is provided. Various realistic instances of the problem are solved to optimality and analyzed to demonstrate that the problem formulation accurately captures all aspects of the problem. This type of analysis was not possible before this methodology was developed.

AFIT/GOR/ENS/09-06

To My Wife

Acknowledgments

I would like to express my sincere appreciation to my faculty advisors, Maj Shane Hall and Dr. James Moore, for their guidance and support throughout the course of this thesis effort. Their insight and experience were invaluable. I would also like to thank Lt Col Jay Desjardins and Maj Jack Evans at Air Force Special Operations Command for agreeing to sponsor this research.

Special thanks go to Dr. Scott Applequist, formerly of the 14th Weather Squadron, for providing the customized gridded climatology model data, to Mr. David Knapp at the Army Research Laboratory for providing insight into current battlefield weather sensing efforts and, finally, to Lt Col Frederick Williams and the men and women of the 28th Operational Weather Squadron for helping keep the model formulation realistic.

Andrew J. Geyer

Table of Contents

Page

List of Figures

List of Tables

Table Page

OPERATIONS-FOCUSED OPTIMIZED THEATER WEATHER SENSING STRATEGIES USING PREEMPTIVE BINARY INTEGER PROGRAMMING

I. Introduction

Background

Throughout history, successful military leaders have recognized that weather conditions on the battlefield can play a significant role in determining the victor. As early as 500 B.C.E., the great Chinese General Sun Tzu wrote in *Art of War* "Know yourself, know your enemy; your victory will never be endangered. Know the ground, know the weather; your victory will then be total......" (Tzu 2005). For this reason, the United States maintains and equips several different types of military units that are tasked to provide dedicated weather support to operational commanders. These units use a variety of types of sensors to collect current weather conditions on the battlefield. These units are also tasked to predict when future weather conditions will present a significant or marginal impact to friendly forces' ability to conduct various types of military operations. For each different type of military operation, unit commanders and technical experts maintain a list of weather condition thresholds that significantly impact a unit's ability to successfully conduct that type of operation (JPub 3-59 2008). There is a similar list of thresholds that marginally impact those same types of operations (JPub 3-59 2008).

United States military doctrine dictates that a senior commissioned officer trained in the regions of Meteorology and Oceanography (METOC) be appointed as the primary weather operations advisor to the overall commander of a military theater (JPub 3-59

2008). This officer is typically referred to as the Senior METOC Officer (SMO). The SMO is responsible for creating a theater weather sensing strategy that makes optimal use of all reliable weather sensing sources within the theater (JPub 3-59 2008). From that strategy, the SMO develops a METOC collection plan. This plan describes where weather sensing equipment and personnel should be located on the battlefield such that, in the SMO's expert opinion, these sensors provide the best depiction of current battlefield atmospheric conditions in support of the commander's Concept of Operations (CONOPS) (JPub 3-59 2008).

Problem Description

Presently, there is no established methodology for determining an optimal weather sensing strategy. SMOs must rely on their individual knowledge of terrain, climate conditions, weather-related impacts to military operations and capabilities of available weather units to create their sensing strategies. Creating this strategy is very difficult for a number of reasons. First and foremost, the number of available weather units and sensors available for allocation is usually very small relative to the size of the battlefield that they are tasked to cover. Obtaining additional weather personnel or sensors in addition to those initially on hand takes significant time and money. Second, different types of weather units have different capabilities. Some units can only be stationed at a friendly installation alongside units from their same branch of service (i.e. Army, Navy, Air Force, etc.). Other weather units is placed into most any environment, but they can only stay in a particular region for a relatively brief time period of time before they must be recovered or rotated with another unit. Third, different types of weather sensors have similar constraints on their ability to operate in a particular region. Some sensors are small and solar-powered. They are placed most anywhere, whereas other sensors are large and require a reliable power source to operate effectively. Additionally, not every sensor can collect all of the weather elements used to determine whether or not conditions will impact friendly forces' ability to conduct a particular type of military operation. Finally, the SMO must also consider the replacement cost and expected time to failure for each type of weather sensor. This is a very labor-intensive process that does not necessarily ensure optimal deployment of all weather personnel and equipment.

Implementing any weather sensing strategy will always require a significant expenditure of time and resources. Once a weather sensing strategy is created, implementing that strategy is further complicated by the fact that the SMO's role is purely that of a subject matter expert. The SMO is rarely in command of the personnel and equipment that the weather sensing strategy is attempting to allocate. The SMO must present the sensing strategy to the overall commander for approval and implementation. When disagreements arise among the commander's staff regarding the benefits of implementing one particular weather sensing strategy versus another, the SMO currently has a very difficult time providing concrete cost-benefit comparisons that clearly advocate for one strategy over another. The SMO must be able to concretely demonstrate to the commander that implementing the weather sensing strategy will contribute to achieving the commander's military objectives in a way that is significant enough to be worth the associated cost or risk.

Scope

In this research, the Battlefield Weather Sensing Strategy Problem (BWSSP) is formulated as a series of preemptive Binary Integer Linear Programs (BILP). The BWSSP finds an optimal METOC collection plan that is focused on collecting the maximum amount of operationally-relevant weather information possible with the resources provided for any potential battlefield environment. The model allocates a set of heterogeneous sensors with varying capabilities and requirements over both space and time dimensions. The space dimension captures the decision on where to place sensors in order to maximize detection of operationally significant weather by each emplaced sensor. The time dimension captures the decision on when to place a sensor at a particular region so as to maximize detection of operationally significant weather conditions over the expected operational lifespan of the sensor.

The usefulness of placing a particular type of sensor at a particular place and time is calculated using a unique combination of historical and model-generated climate statistics, operationally significant weather element threshold parameters and characteristics of the sensor such as detection capability ranges and expected time to failure. An optimal solution of this formulation yields the highest weighted probability of detection of the weather conditions that most significantly impact the operations most critical to the overall commander's battle plan for the longest amount of time, based on the expected time to failure for the various sensor types. The optimal solution tells the SMO both where and when to deploy the available sensors.

The statistical distributions of weather sensor operational lifespans are assumed to be either proprietary or very complex. As such, a sensor may continue to function well beyond its expected lifespan. Also, in the case of clandestine weather sensors, returning to a region after emplacing a sensor may reveal the existence of that sensor to an adversary. To avoid either allocating a sensor to a currently occupied region or compromising a new clandestine sensor, the BWSSP in this thesis assumes that once a weather sensor is allocated to a particular region at a particular time, it will not be replaced for the duration of the campaign planning time period. The BWSSP can be forced to allocate sensors to user-defined regions. Therefore, once a sensor stops functioning, it is assumed that the SMO will remove the malfunctioning sensor from the current BWSSP solution. If the region is now ineligible for sensor deployment, it can be removed from the BWSSP instance. The SMO is then be able to re-optimize the BWSSP with the remaining unallocated supply of sensors while keeping all currently functioning weather sensors in their respective places. This new solution remains optimal for the given battlefield conditions.

In practice, this model will be applied by SMOs in deployed regions around the globe. SMOs are highly skilled atmospheric scientists and military officers, but very few are trained in operations research techniques. As such, all computer code is written in Visual Basic (VBA®) for Microsoft Excel® with an emphasis on minimizing the number of decision variables at each step (without loss of fidelity) in an attempt to keep the problem solvable by Excel® Premium Solver® Version 9 in less than 24 hours on a standard laptop computer capable of running Microsoft Office 2007®.

Contribution of Research

There are a variety of approaches to sensor allocation in current literature. This is the first approach that considers historical behaviors of the detection targets as a series of preemptive objectives to form a measure of the usefulness of a particular sensor placement. Gridded climatology models and historical weather observations provide reasonable estimates of when and where operationally significant weather conditions are expected to occur. Since the objective of the BWSSP is to maximize detection of these conditions, these estimates of past behaviors provide a reasonable estimate of target behaviors for the weather sensors allocated in the METOC collection plan (BWSSP optimal solution). The methodology applied to the BWSSP can be expanded to other resource allocation problems with multiple, preemptive objectives and where reasonable estimates of target behaviors are known and where time, space and the lifespan of the allocated resource are factors for consideration.

Additionally, this research creates the first mathematical model for a military METOC collection plan. The BWSSP model, as outlined in this research, can be used to compare METOC collection plans for differing supplies of weather sensors, thereby providing the SMO and overall commander with the ability to create a cost-benefit analysis for weather sensor procurement or deployment decisions.

Overview

This thesis comprises five chapters and three appendices. Chapter 2 describes the types of data collected in a weather observation and the methods used to collect that data. Chapter 2 also describes how that data is used to create gridded climatology models.

Chapter 2 then describes the United States military doctrine that governs military campaign planning and the role of military weather personnel and weather data in that process. Chapter 2 concludes with a review of current research literature in the areas of ad-hoc wireless sensor network design and optimization. Chapter 3 outlines the mathematical model for the BWSSP. The results of multiple, simulated instances of the BWSSP are presented in Chapter 4. Sensitivity analysis is performed on a BWSSP optimal solution (METOC collection plan) to demonstrate how changes in sensor availability can affect the usefulness of the METOC collection plan. Chapter 5 outlines the conclusions of this research, other possible applications for the BWSSP and areas for future research into the BWSSP. Appendix A is the Microsoft® Visual Basic® computer code that was used to create and evaluate the instances of the BWSSP in Chapter 4. Appendix B is an Op-Ed column on the value of this research for the Air University "Blue Dart" program. Appendix C contains an image of a story board PowerPoint® slide that briefly describes this research.

II. Literature Review

Introduction

This chapter discusses the literature that provides the foundation for this research. The main areas of interest are weather sensors, weather data collection, the role of weather in military campaign planning and ad-hoc wireless sensor network design schemes.

Surface Weather Sensors in the BWSSP Model

When meteorologists talk about the weather, they are talking about the state of one or more of these weather elements: air temperature, air pressure, humidity, cloud height, total cloud coverage, precipitation, visibility, wind speed and wind direction (Ahrens 2000). A weather observation is a report of the current state of these elements. In general, weather observations are divided into two types: surface and upper-air. Surface weather observations are measurements of the current state of weather elements as observed by manned instruments or automated sensors located on the surface of the Earth (AFMAN 15-11 2008). Air Force Manual 15-111 *Surface Weather Observations* (2008) dictates that surface weather observations contain current measurements of all of the aforementioned weather elements. They transmitted with no delay at a minimum of once an hour on the hour. If possible, surface weather observations should also be transmitted when weather conditions change significantly either for the better or worse (AFMAN 15-111 2008).

The military forces of the United States and its allies use a variety of different types of sensors to gather the essential weather elements for a surface weather observation. For the purposes of the BWSSP model, these sensors are placed into four general sensor categories: automated fixed base, manned fixed base, automated tactical, and manned tactical. In the BWSSP model, an automated fixed base sensor is a surface weather sensor that requires a secure installation with a reliable source of power to operate. This type of sensor does not require augmentation from a trained weather observer in order to transmit a valid surface weather observation, though it may require occasional maintenance to remain operational. An example of this type of sensor commonly in use today is the TMQ-53 Tactical Meteorological Observation System (TMOS) manufactured by Vaisala® Corporation (see Figure 1).

Figure 1. TMQ-53 TMOS Undergoing Maintenance Check (Kuykendall 2007)

Similarly, a manned fixed base sensor requires a secure installation with a reliable source of power to operate. Typically, a sensor of this type is actually a trained military weather technician with a set of approved handheld weather sensors with access to a radio or internet connection. These personnel are part of the conventional military forces and are not trained to operate independently on the battlefield. Therefore, they are restricted to operating at regions secured by friendly forces such as bases and airfields. Typical handheld sensors used as part of this type of "sensor" are the Kestrel 4000® handheld wind, temperature, dew point and pressure sensor; plastic rain gauges; handheld lightning detection systems and laser rangefinders for determining cloud heights (see Figures 2-5).

Figure 2. Kestrel 4000®

(REI Outdoor 2008)

Figure 3. Rain Gauge

(National Weather Service 2002)

Figure 4. Lightning Detector

(Electronics 2007)

Figure 5. Laser Rangefinder

(Sniper's Paradise 2006)

As defined for the BWSSP, automated tactical weather sensors are sensors that can operate and transmit surface weather observations from most anywhere on the battlefield. They are typically deployed clandestinely and can operate for a significant time period of time under their own power. The last type of sensor, tactical manned weather sensors, are the same as the manned fixed base sensors except that the personnel using the equipment are weather personnel who are trained to operate and survive most anywhere on the battlefield either on their own or as part of a small team. These personnel are ground combat forces trained as Forward Region Limited Observers (FALOPs), friendly indigenous personnel trained and equipped as part of a clandestine weather observation network or they may be forward-deployed United States Air Force Special Operations Weather Teams (SOWTs).

Figure 6. SOWT Personnel (United States Air Force 2008)

Upper-Air Weather Sensors in the BWSSP Model

Upper-air weather observations are measurements of the current state of weather elements as observed by a sensor as it ascends through the atmosphere. For the purposes of the BWSSP model, the primary upper-air weather sensors are weather balloons (also known as "rawinsondes") (Ahrens 2000). Federal Meteorological Handbook No. 3 - *Rawinsonde and Pibal Observations* (FMH-3 1997) mandates that when possible, upper-air weather balloon observations should be taken at 0000 and 1200 Greenwich Mean Time every day. These observations are to be transmitted as rapidly as possible to the global weather information network. FMH-3 (1997) mandates that all weather balloon sensors must be calibrated with known reliable surface weather sensors prior to launch. As such, the BWSSP requires the co-location of surface and upper-air sensors.

A typical weather balloon sensor package consists of a disposable sensor bundle attached to a helium balloon. The sensor package measures the same weather elements as a surface weather sensor, but at multiple heights as the balloon lifts it through the atmosphere. The sensor package is equipped with a radio transmitter that sends its measurements to the surface in real-time. Most modern weather balloon sensor packages are also equipped with a GPS receiver so that it can also transmit its current position. Most weather balloon systems require a secure installation for operation. United States and allied artillery meteorological (ARTYMET) teams also launch weather balloons to aid in artillery targeting. ARTYMET weather balloon data can also be included in the BWSSP model. Additionally, SOWT personnel sometimes are able to launch weather balloons from other regions on the battlefield for very short periods of time, depending on the equipment available and the tactical situation they face (see Figure 8).

Figure 7. SOWT Operator Releases Weather Balloon (Emery 2006)

Weather Radars in the BWSSP Model

In the BWSSP model, weather radars are treated much like upper-air weather sensors because they are used to continuously examine the behavior of clouds, wind and air masses at levels well above the surface of the Earth by bouncing microwave radiation off of water particles (Ahrens 2000). In order to be effective, tactical weather radars need continuous network connectivity and a reliable power supply. As such, they can only be deployed to secure installations (see Figure 7).

Figure 8. E600 Tactical Weather Radar Tower (EWR Weather Radar Systems 2006)

All weather radars require trained weather personnel to operate them. For this reason, the BWSSP requires that a surface weather sensor be co-located with any weather radar in order to make the best use of military weather personnel.

Combined Weather Sensors in the BWSSP Model

Depending on the terrain and resources available, it may be advantageous to combine multiple sensors together in one region to obtain additional useful weather information. For example, one may wish to place a series of surface weather sensors at intervals scaling up the side of a mountain. This combination of sensors will then provide useful information on the profile of the atmosphere and would behave like an upper-air sensor. In this case, the SMO would enter the sensor placed at the foot of the mountain as a surface sensor in the BWSSP model. The remaining set of surface sensors placed on the mountain slope would be input as one upper-air weather sensor in the BWSSP model.

Weather Observation Collection and Dissemination

Once collected, military surface and upper-air weather observations are used for two main purposes. The first purpose is to provide timely data to theater weather forecasting centers for Numerical Weather Prediction (NWP) computer model initialization. Timely delivery of observations that provide an accurate depiction of the current state of the atmosphere are vital to proper NWP model initialization, thereby ensuring it provides accurate weather forecast data (Holton 1992). Higher-resolution NWP models, like those currently used in military weather forecasting, are generally initialized in two steps: objective analysis of the observations and data initialization (Holton 1992). In the objective analysis step, all of the irregularly spaced surface and upper-air weather observation data taken at the NWP model's initialization time are checked for accuracy and are then interpolated to determine the state of the weather elements at standard levels of the atmosphere over evenly-spaced latitude and longitude points on the surface of the Earth (Holton 1992). In the data initialization step, the evenly-spaced interpolated data points are then further smoothed to eliminate noise in the NWP model due to interpolation and rounding errors. These smoothed, evenly-spaced data points are then used as the input data for the NWP model (Holton 1992).

The second purpose for taking weather observations is to provide friendly force weather personnel with a well-developed picture of the current state of the weather conditions across the battlefield. This data is used to validate NWP model output as part of the weather forecasting process. Additionally, a well-developed picture of current battlefield weather conditions allows forecasters to inform unit commanders of current or approaching weather conditions that may impact ongoing operations. This process is known as MISSIONWATCH in United States Air Force weather doctrine (AFMAN 15-129 2004). In order to conduct MISSIONWATCH effectively, friendly force weather personnel need to be fully aware of all weather conditions that can impact military operations through the duration of a campaign.

Role of Weather in Military Campaign Planning

United States military doctrine defines a military campaign as "a series of related military operations aimed at accomplishing a strategic or operational objective within a given time and space" (JPub 3-0 2008). When planning for a military campaign, doctrine states that the overall commander should create a draft mission statement, commander's intent, and the pertinent CONOPS. The mission statement is usually a short and clear statement of what needs to be done during the campaign and why these actions are important for overall mission success. The commander's intent is a concise expression of the campaign's purpose and the desired military end state when the campaign is concluded (JPub 3-0 2008). The CONOPS describes how the efforts of all subordinate units will be integrated, synchronized, and phased to achieve the desired military end state at the conclusion of the campaign (JPub 3-0 2008).

United States military campaigns generally follow a six phase model. These phases are: shape, deter, seize initiative, dominate, stabilize, and enable civil authority. Depending on the mission statement and commander's intent, a campaign plan may not feature all six phases. In the campaign plan, each phase of the campaign has an anticipated length of time to completion as well as a set of military objectives that must be met before that phase is completed. Depending on the conditions on the ground, anticipated times to completion for phases of a campaign can shift drastically in either direction on the calendar. Therefore, as campaigns progress, the campaign plan is continuously refined to ensure it accurately fulfills the commander's intent (JPub 3-0 2008).

Prior to the launch of a military campaign and during the refinement of an ongoing campaign, commanders and their staff conduct Joint Intelligence Preparation of the Operational Environment (JIPOE). JIPOE is a four step process: define the battlefield environment, describe the battlefield's effects, evaluate the adversary and determine adversary potential courses of action (JPub 2-01.3 2000).

In order to define the battlefield environment, one must examine the climate of the different regions on the battlefield. For almost every potential conflict region in the world, historical weather observation data (climatology) is not available or reliable (JPub 3-59 2008). To compensate for this lack of critical information, the United States Air Force Weather Agency's 14th Weather Squadron located in Asheville, North Carolina created the Advanced Climate Modeling and Environmental Simulation (ACMES) program. Using all reliable historical weather observations available, the ACMES model uses NWP techniques to model historical weather conditions over time periods of several years. The ACMES model interpolates climatological statistics for evenly-distributed points on the surface of the Earth at 100 km, 45 km, 15 km and 7.5 km resolutions in the same manner as NWP models predict future conditions (14th Weather Squadron (AFWA) - USAF Accessed: September 11, 2008).

The ACMES model provides a wide variety of climatological output statistics in a variety of formats. For example, Figure 9 depicts the mean daily surface temperature for the Korean peninsula in the month of January over a ten year time period. The flexibility of the ACMES model and its ability to provide a reasonable estimate of climate conditions in regions where little to no data is available makes it a very popular tool among military weather forecasters for use in the first step of JIPOE.

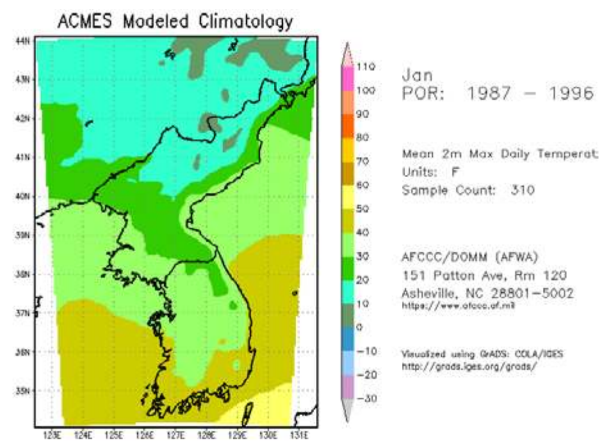


Figure 9. ACMES Example Graphical Output
(Accessed: 14th Weather Squadron (AFWA) - USAF September 11, 2008)

For the last three steps of JIPOE (describe the battlefield's effects, evaluate the adversary and determine adversary potential courses of action), United States military doctrine states that "critical parameters should be established for each weather aspect in order to define the thresholds at which deteriorating weather conditions is expected to have favorable, marginal, or unfavorable effects on specific types of operations and equipment" (JPub 2-01.3 2000). These critical parameters are usually collected by the senior weather officer from various sources to include technical manuals, equipment operators, intelligence experts and unit commanders (JPub 2-01.3 2000). Once compiled, the list of these critical weather element thresholds is commonly referred to as a Weather Effects Matrix (WEM). Though there is no specified format for the WEM, many units organize their WEM into a format similar to the example given in Table 1.

Table 1. Example WEM for One Military Campaign Phase

OPERATION	FAVORABLE (No Degradation)	MARGINAL (Some Degradation)	UNFAVORABLE (Significant Degradation)
SEA PORTS	WIND < 20 KTS	WIND 20 - 35 KTS	WIND > 35 KTS
AIR PORTS	CEILING > 1500 FT	CEILING 200 - 1500 FT	CEILING < 200 FT
	VISIBILITY > 4800 METERS	VISIBILITY 900 - 4800 METERS	VISIBILITY < 900 METERS
BRIDGING	WIND < 10 KTS	WIND 10 - 34 KTS	WIND > 34 KTS
SIGINT	WIND < 30 KTS	WIND 30 - 45 KTS	WIND > 45 KTS
		TEMPERATURE 85 - 120 F	TEMPERATURE < 32 F
HELO LIFT	CEILING > 500 FT	CEILING 300 - 500 FT	CEILING < 300 FT
(NO SPECIFIC AIRFRAME)	VISIBILITY > 1600 METERS	VISIBILITY 800 - 1600 METERS	VISIBILITY < 800 METERS
(FLT LVL < 10000 FT)	NO TURBULENCE / ICING	LGT - MDT TURBULENCE / ICING	SVR TURBULENCE / ICING
TRAFFICABILITY	NO PRECIPITATION	LGT - MDT PRECIPITATION	HVY PRECIPITATION
NBC OPS		WIND < 10 KTS	WIND > 30 KTS
			WIND CALM
	NO PRECIPITATION	LGT PRECIPITATION	MDT - HVY PRECIPITATION
CAS	CEILING > 2000 FT	CEILING 1000 - 2000 FT	CEILING < 1000 FT
(PLANNING PURPOSES)	VISIBILITY > 8000 METERS	VISIBILITY 3200 - 8000 METERS	VISIBILITY < 3200 METERS
STRAT RECON	VISIBILITY > 8000 METERS	VISIBILITY 4800 - 8000 METERS	VISIBILITY < 4800 METERS
(FLT LEVEL > 25000 FT)	CLOUD COVER: SKC OR FEW	CLOUD COVER: SCT	CLOUD COVER: BKN OR OVC
HIGH RECON	VISIBILITY > 8000 METERS	VISIBILITY 4800 - 8000 METERS	VISIBILITY < 4800 METERS
(FLT LEVEL > 8000 FT)	CLOUD COVER: SKC OR FEW	CLOUD COVER: SCT	CLOUD COVER: BKN OR OVC
LOW RECON	VISIBILITY > 8000 METERS	VISIBILITY 4800 - 8000 METERS	VISIBILITY < 4800 METERS
(FLT LEVEL > 3000 FT)	CLOUD COVER: SKC OR FEW	CLOUD COVER: SCT	CLOUD COVER: BKN OR OVC
GROUND RECCE	VISIBILITY > 3000 METERS	VISIBILITY 1000 - 3000 METERS	VISIBILITY < 1000 METERS
AIRBORNE	WIND < 13 KTS	WIND 13 - 18 KTS	WIND > 18 KTS
			CEILING < 1000 FT
	NO PRECIPITATION	LGT PRECIPITATION	MDT - HVY PRECIPITATION
	DENSITY ALTITUDE < 4000 FT	DENSITY ALTITUDE 4000 - 6900 FT	DENSITY ALTITUDE > 6900 FT
PERSONNEL	NO PRECIPITATION	LGT PRECIPITATION	MDT - HVY PRECIPITATION
(TEMP/HEAT/WINDCHILL)	TEMPERATURE 20 - 85 F	TEMPERATURE -15 - 20 F	TEMPERATURE < -15 F
		TEMPERATURE 85 - 95 F	TEMPERATURE > 95 F

Regardless of format chosen, the WEM consists of a list of the different types of military operations planned in the CONOPS for a given phase of the campaign. If the CONOPS change from one phase to another, the WEM must change, as well. Though not necessary, this list of operations may be put in order of precedence with respect to the commander's main effort for that phase. For example, if building up supply stores was the commander's main focus for a particular phase, sea port and air port operations would be listed first in the WEM, as they are in Table 1 above.

For each type of operation listed in the WEM, there are three associated lists of weather element thresholds. The unfavorable or "red" weather element threshold list consists of a list of weather conditions that significantly degrade friendly (or possibly hostile) forces' ability to conduct that particular type of operation. The marginal or "amber" weather element threshold list consists of a list of weather conditions that slightly degrade friendly (or possibly hostile) forces' ability to conduct that particular type of operation. The favorable or "green" weather element threshold list consists of a list of weather conditions that provide ideal conditions for friendly (or possibly hostile) forces' to conduct that particular type of operation.

For example, if Close Air Support (CAS) was a concern to a commander following the CONOP depicted in Table 1, a military weather forecaster would focus primarily on predicting when either the visibility will be less than 3,200 meters or when the cloud cover will exceed one half of the total sky (known as a ceiling) at an altitude of 1,000 feet or less. If those conditions were not likely to occur, the weather forecaster would then focus on predicting when either the visibility could be between 3,200 meters and 8,000 meters or where a cloud ceiling could form between 1,000 to 2,000 feet altitude. If none of these conditions are forecast, a commander will expect to have no degradation of his CAS capabilities due to the weather conditions.

All of these weather element thresholds are calculated from the weather element parameters found in climatology, weather observations and NWP forecast data. For short term operational planning, the WEM thresholds are compared with weather observations, NWP model output and man-made weather forecast products to determine the anticipated weather impact to the planned operations (JPub 3-59 2008). For long-term campaign planning, climatology is used to determine potential weather impacts to planned military operations (JPub 2-01.3 2000).

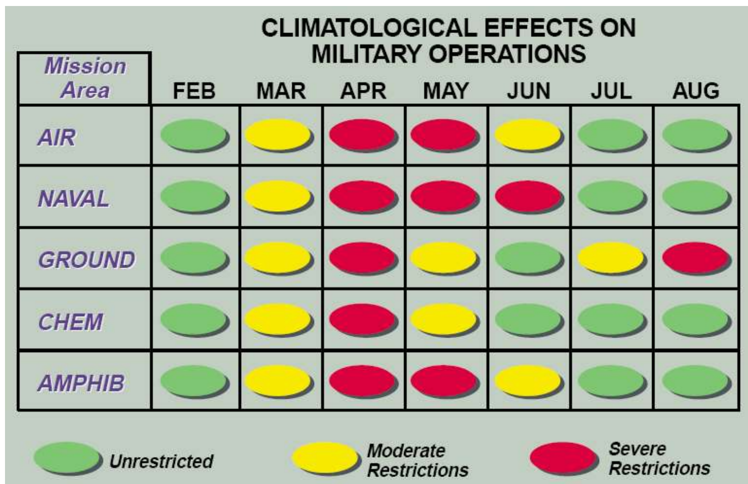


Figure 10. Weather Impacts to Operations Based on Climatology (JPub 2-01.3 2000)

Since large-scale weather patterns are seasonal and the months of the calendar year approximately follow the seasons; climatological statistics are typically calculated by month (Ahrens 2000). Therefore, weather impact forecasts for long-term planning are typically divided in the time dimension by changes in phase of the campaign and then further subdivided by any changes in the month of the calendar year that fall during that phase of the campaign. As demonstrated in Figure 9, climatological conditions also change depending on region. The common practice among military weather forecasters is to divide the battlefield map up into large climatological regions where conditions are expected to be generally similar. Products like the example in Figure 10 are then created for each climatological region using the WEM for each planned campaign phase.

Creating products in this manner provides campaign planners with the level of detail required to aid their efforts while minimizing the workload of the usually small military weather forecasting section.

The gridded output of the ACMES model also permits another approach to using climatological data as an aid in military campaign planning. Though not as efficient as the current large climatological region approach, it is possible to treat each individual data point in the ACMES model output as a separate climatological region. Just as in the current process, analysis can be performed for each data point to determine the monthly by-hour percent occurrence of when that region is in the "red" category for a particular operation. Similar analysis can be done to determine the monthly by-hour percent occurrence of conditions that are in the "amber" category for a particular operation. This analysis can be repeated for every type of operation listed in the WEM for every planned phase of the campaign. For the ACMES model at 15 km resolution, this could mean performing this analysis for a very large number of 225 km^2 roughly square-shaped climatological regions. For an average-sized country like Afghanistan, this process would mean dividing the country into 647,500 $km^2 \div 225\ km^2 = 2{,}878$ climatological regions (Central Intelligence Agency 2008). In realistic cases, many of those 2,878 regions probably would not be considered for climatological analysis because they would either not be reachable by friendly forces or because friendly forces would have no plan to enter that particular region throughout the duration of the campaign plan. Even with those constraints eliminating possibly hundreds of regions, it is easy to see why this method is not standard practice. However, performing climatological analysis with this approach will lead to a novel and efficient formulation of the BWSSP model.

The BWSSP as an Ad-Hoc Heterogeneous Wireless Sensor Network

A wireless sensor network consists of a set of sensors where each sensor has its own power supply and transmission capability (Shih et al. 2006). These sensors are deployed over a region and transmit data to a central data collection node (called a sink) either directly or by using other sensors in the network as intermediaries (Shih et al. 2006). A wireless sensor network that consists of different types of sensors with varying capabilities is said to be heterogeneous (Shih et al. 2006). An ad-hoc wireless sensor network is a wireless sensor network where each sensor is not necessarily intended to be permanent and sensors usually have an expected operational lifespan (Shih et al. 2006). This type of network is usually constructed when sensors are required to provide information from regions that are too difficult, dangerous or costly to access frequently (Shih et al. 2006). An optimal solution to the BWSSP creates a METOC collection plan which is an ad-hoc heterogeneous wireless sensor network, since it allocates all available weather sensors of any type to regions on the battlefield at a time that maximizes detection of "red" and "amber" weather conditions over the expected lifespan of the weather sensors.

The variety of wireless sensor types available for practical use in networks continues to increase significantly as batteries improve and electronic sensors become smaller and more effective (Rowaihy et al. 2007). The best combination of sensors to select for any wireless sensor network is determined by solving the Sensor Selection Problem (SSP) (Rowaihy et al. 2007). Given a set N of sensors, the SSP seeks the "best" subset of sensors $N' \in N$ that provides a desired level of gathered information within a particular budget. The SSP is at least as hard as the weakly NP-hard Knapsack problem (Garey and Johnson 1979). In instances of the BWSSP with only one time period, the BWSSP seeks the "best" types of sensors to allocate to a set of regions in order to gather the required weather information with a budget of available sensors. Therefore, the BWSSP is at least as hard as the SSP, meaning the BWSSP is also NP-hard. This means that, in the worst case, optimal solutions for instances of the BWSSP cannot be found in polynomial time (assuming $P \neq NP$).

In their survey of wireless sensor network designs, Rowaihy et al. (2007) stated, that once the optimal sensor types are selected, these networks are constructed utilizing one of four schemes: target tracking and localization, single mission assignment, multiple missions assignment and coverage. Target tracking and localization networks are primarily concerned with the network's ability to detect one or more targets that pass within its detection region. Optimization schemes for this type of network usually focus on maximizing target detection probability at minimum cost, however, there is no estimate of target behavior. Sensors are assumed to have a probability distribution governing their ability to detect a target when it enters its detectable region. As such, various statistical methods are used to determine either sensor placement or when to activate (or deactivate) sensors within the network (Rowaihy et al. 2007).

The BWSSP is focussed on detection of bad weather over and around the region of each sensor. Air Force Manual 15-129 *Air and Space Weather Operations - Processes and Procedures* (2004) mandates all weather sensors be certified as fully functional and accurate prior to their use. It also requires weather personnel be fully trained on the correct installation and placement of weather sensors. This requirement ensures sensors are providing an accurate picture of the environment and they are able to relay observational information in a timely fashion. Therefore, the BWSSP assumes all sensors accurately detect and report on the weather elements occurring within the sensor's respective detectable area. Any weather sensors in the BWSSP requiring battery power are usually unmanned tactical surface weather sensors. These sensors continuously conduct MISSIONWATCH, so they transmit on a fixed schedule (AFMAN 15-111 2008). The BWSSP assumes the operational lifespan of a weather sensor is the expected time to failure of one or more of the sensor components, regardless of the target environment. As such, battery life optimization schemes are not applicable to the BWSSP. Operationally significant inclimate weather conditions are the "targets" to be detected by the BWSSP wireless sensor network and the ACMES model data provides a reasonable estimate of target behavior. Since target behavior is known with a reasonable level of certainty in the BWSSP, the BWSSP does not fit into a target tracking and localization wireless network scheme.

In single and multiple mission assignment schemes, usefulness value functions are created to determine which types of sensors should be placed where to best detect known target(s). Multiple targets may arrive simultaneously and must be dealt with by the available sensors. The missions are individual sensor tasks that may or may not be shared by other sensors in the network. This wireless sensor network design scheme is concerned with optimizing the configurations of individual, multi-modal sensors (Rowaihy et al. 2007).

With the exception of a weather radar, the weather sensors in the BWSSP operate in only one mode and are always attempting to detect the same weather elements. A weather radar is capable of detecting different weather phenomena depending on the current state of the atmosphere (Ahrens 2000). However, the decision to change radar settings does not significantly affect its operational lifespan nor does it affect the decision on its placement. As such, the BWSSP does not fit into the single and multiple mission assignment schemes for wireless sensor network design.

The coverage wireless sensor network design scheme is concerned with conserving energy consumption while maintaining total coverage of a geographic area. Redundant sensors are placed within the geographic area in a manner such that sensors is activated and deactivated to maximize the useful lifespan of the sensor network (Rowaihy et al. 2007). Since battery life is not a concern in the BWSSP, the BWSSP does not fit into the coverage wireless sensor network design scheme.

In addition to the schemes reviewed by Rowaihy et al. (2007), recent research efforts have approached wireless sensor network design as a resource allocation problem. Xu and Sahni (2006) use a deterministic sensor deployment scheme formulated as a BILP. Their formulation ensures a predetermined level of sensor coverage at minimum cost and hence, their objective function is focussed solely on minimizing cost. Their methodology chooses sensor deployment regions from a gridded set of pre-determined sites. They assume that the available sensors are able to communicate with the central data collection point from any of the feasible deployment regions. Their first set of constraints ensures the required amount of coverage is met for all regions of the network. Their second set of constraints ensures that no more than one sensor is deployed to any region. This gridded BILP formulation is efficient, as it takes only $O(sn)$ operations to create the set of constraints, where s is the number of sensors and n is the number of feasible regions for sensor deployment. This rudimentary formulation provides a foundation for a resource allocation approach to the BWSSP, however, Xu and Sahni's formulation only considers homogeneous sensor networks. Furthermore, their formulation does not account for sensor lifespan or targets that change over time, which are critical aspects that must be considered in any formulation of the BWSSP.

Ramadan et al. (2006) describes a sensor deployment tool called *SensDep* that allocates sensors in a fashion similar to Xu and Sahni's. The *SensDep* tool also uses a BILP to allocate sensors to regions, considers heterogeneous wireless sensors and considers sensor lifespan. However, the *SensDep* tool assumes sensors is activated (or deactivated) in order to extend battery life. The *SensDep* objective function maximizes coverage of the network while also minimimizing energy consumption. This is accomplished by subtracting energy consumption parameters from the coverage amount in the objective function. This is the coverage scheme as described in Rowaihy et al. (2007). Though the use of sensor lifespan in the BILP formulation of *SensDep* provides some insight into how sensor lifespan may be incorporated into a resource allocation approach to the BWSSP, the *SensDep* tool follows the coverage sensor network design scheme. The BWSSP seeks to maximize detection of occurrence of inclement weather at each region while *SensDep* seeks to maximize coverage over an entire geographic region. Therefore, the BWSSP does not fit into the methodology in use in *SensDep*.

Pizzocaro et al. (2007) use a Multiple Knapsack approach to a sensor assignment and deployment problem for intelligence-gathering sensors on a virtual battlefield. In their model, they use a virtual battlefield divided into a set of regions. Each region has its own deterministic surveillence requirement. The "coverage" given by each sensor type is its respective value in the Multiple Knapsack problem. Their objective function maximizes coverage of the battlefield using the minimum number of sensors from a heterogeneous set by subtracting the number of sensors from the coverage parameters. The first set of constraints in this model are similar to those in Xu and Sahni's model; they ensure that no sensor is deployed to more than one region. In addition to these constraints, Pizzocaro et al. (2007) add a set of constraints that ensure sensors deployed into a particular region are able to satisfy the sensing requirements in that region.

Pizzocaro et al. (2007) provides further insight into a formulation for the BWSSP. It demonstrates that a Multiple Knapsack approach provides a weakly NP-hard formulation for the sensor selection and deployment problem. However, this model does not account for the operational lifespan of the sensors nor does it account for sensing requirements that change over time. Adding a time dimension to the Multiple Knapsack formulation provides a mechanism for incoorporating the operational lifespan and the changing nature of the sensing requirements into the model. With the addition of the time dimension, this formulation takes on the form of a Binary Multiple Knapsack resource allocation problem with multiple resources. This formulation provides a novel (though NP-hard) formulation for the BWSSP.

Conclusion

The foundational literature for this research comes from a varied list of sources to include works discussing the attributes of weather sensors, weather data collection, the role of weather in military campaign planning and ad-hoc wireless sensor network optimization schemes. The literature for ad-hoc wireless sensor network optimization schemes is vast and continues to grow as sensing technology improves. The various ad-hoc wireless sensing network optimization schemes all provide a foundation for the BWSSP, but no single methodology captures it entirely. Specifically, there does not appear to be any wireless network sensing scheme in the literature with a set of preemptive objectives. Furthermore, there is a lack of significant research focused on detecting targets whose behavior has a known pattern of behavior that changes over time. In the next chapter, a formulation that captures all aspects of the BWSSP is presented. Therefore, this research expands the current literature by presenting a scheme for wireless sensor networks that considers historical behaviors of the detection targets as a series of preemptive objectives to form a measure of the usefulness of a particular sensor placement. In addition, this research provides the first mathematical model for developing a realistic METOC collection plan for any type of military campaign anywhere on the globe.

III. Methodology

Introduction

This chapter presents the general formulation of the BWSSP. The parameters, sets and decision variables in the BWSSP are explained in terms of readily available data. These entities are then formulated first as a series of preemptive BILP objective functions and then as a combined, single BILP that maintains the preemptive nature of the original objectives. The constraints on the BWSSP BILP are described in detail in terms of the attributes of the BWSSP parameters. BWSSP complexity is then discussed with a preview of the approach taken in Chapter IV to solve several BWSSP instances to optimality. Optimal solutions to the BWSSP are described as a METOC collection plan in terms of the BWSSP parameters. The chapter concludes with an approach to sensitivity analysis of differing BWSSP optimal solutions based on the improvement or degradation of the METOC collection plan in terms of the BWSSP parameters.

BWSSP Sets, Parameters and Decision Variables

While conducting JIPOE as part of initial campaign planning, the SMO will know the layout of the battlefield, the planned phases of the campaign and the expected timetable for execution of these phases. Therefore, the entire expected duration of the campaign is divided into a set $T = \{1, 2, ..., \tau\}$ of time periods where transition from one time period to the next is denoted by either the expected change in the phase of the campaign or a change in the calendar month (a reasonable estimate of change in climate conditions). The time periods in the set T are not necessarily of the same duration, but no single time period will exceed the length of one calendar month.

As noted previously, any battlefield can be subdivided into a set of roughly square-shaped climatological regions, each with a data point of the ACMES gridded climatology model in its center. The size and number of these regions depends on the resolution of the ACMES model and the size of the battlefield. Occasionally, these regions may need to be further subdivided because they are bisected by a climatologically significant terrain feature such as a mountain range or large river. Regions may also need to be further subdivided because they feature a friendly installation or airfield that could require a weather observation, but a weather observation at that installation or airfield would not provide an accurate depiction of the weather occurring in the rest of the region determined by the ACMES data point. Additionally, entire regions may be eliminated from consideration in the BWSSP because they either lay out of the region of interest or they are not feasible for deployment of any of the available weather sensor types. After applying all of this analysis to the ACMES model output, the battlefield will naturally be divided into a set $M = \{1, 2, ..., \mu\}$ of climatological regions that are viable candidates for deployment of at least one of the available weather sensor types. For any two regions $m \in M$ and $q \in M$, let d_{mq} be the physical distance between the center of region m and the center of region q.

Another consideration in the formation of any sensing strategy is the region of friendly installations, outposts, field headquarters, villages, etc. Since some sensor types require external support and security in order to operate, the BWSSP must account for these regions. Let $H_t \subseteq M$ be the set of regions that contain secure friendly installations at time period $t \in T$. Sensors requiring a secure installation must be deployed to regions in the set H_t. For sensors that do not require a secure installation, it is advantageous to deploy them to the region $m \in M$ with the greatest frequency of inclement conditions that is furthest away from the nearest friendly region, as weather conditions progress across the battlefield from one direction to another. Let $h_{mt} = \min_{q \in H_t \subseteq M} \{d_{mq}\}$, the distance to the nearest region in set H_t from region $m \in M$.

As a part of the SMO's campaign planning, he or she will create a set $N = \{1, 2, ..., v\}$ of all weather sensor types available for deployment during the expected duration of the campaign. As previously noted, the position of upper air sensors requires the co-location of a surface sensor in order to ensure either proper calibration of the upper air sensor or optimal use of weather personnel. Since these weather personnel must be present to operate an upper air weather sensor or tactical weather radar, they will also want to take a surface weather observation. Other than the co-location requirement, the positioning of upper-air weather sensors or weather radars does not necessarily affect the positioning of surface weather sensors. Furthermore, the positioning of upper-air weather sensors does not affect the positioning of weather radars and vice-versa. For example, the positioning of a tactical weather radar does not necessarily affect the positioning of a weather balloon launch site, but the positioning of a tactical weather radar effects the positioning of all other tactical weather radars. Therefore, let $B \subset N$ be the set of upper-air weather sensors that gather upper-air profile data (e.g., weather balloons) and $E \subset N$ be the set of weather radars.

For each sensor type $n \in N$, the SMO is able to calculate an effective sensing range (or minimum spacing distance) δ_n. This parameter is determined by the behavior of the sensor type, the effect of terrain on the sensor or by the desired minimum spacing between sensors that perform the same function (surface, upper-air or radar). During each time period $t \in T$, the SMO will have a reasonable estimate of the number of sensors of type $n \in N$ available for use in the campaign weather sensing strategy. Let S be a $\nu \times \tau$ matrix where the parameter s_{nt} in S signifies the number of sensors of type $n \in N$ that will arrive in theater at the start of time period $t \in T$.

Once the number of sensors available is determined, the SMO can also determine the sets $L_{nt} \subseteq T$, where L_{nt} is the set of time periods sensor type $n \in N$ remains operational provided it is deployed at the start of time period $t \in T$, and hence L_{nt} defines the operational lifespan of sensor type $n \in N$. Finally, let A be a $\nu \times \mu \times \tau$ tensor of binary parameters where the parameter a_{nmt} in A is 1 if sensor type $n \in N$ is deployable to region $m \in M$ at the start of $t \in T$ and 0, otherwise. A region may no longer be reachable after a certain time period (e.g. $a_{nmt} = 1$, but $a_{nm(t+1)} = 0$), but a properly deployed sensor should continue to operate until its components fail, barring discovery by the enemy or any other unforeseen calamity. Therefore, if sensor type $n \in N$ is deployed into a particular region $m \in M$, the BWSSP assumes that the sensor will continue to operate over its expected operational lifespan.

For every time period $t \in T$, there is a corresponding WEM derived from the CONOPS. In each WEM, there is listed a set of planned operation types O_t. If provided in the CONOPS, the SMO can assign a priority p_{ot} to each operation type $o \in O_t$ where the highest priority operation type $o \in O_t$ has $\overline{p}_{ot} = \max_{o \in O_t} \{p_{ot}\}$ and the operation type $o \in O_t$ with the lowest priority has $\underline{p}_{ot} = \max_{o \in O_t} \{p_{ot}\}$. The weights should be scaled so that $p_{11} = \max_{o \in O_1} \{p_{o1}\} = p_{12} = \max_{o \in O_2} \{p_{o2}\} = ... = p_{1\tau} = \max_{o \in O_\tau} \{p_{o\tau}\}$ for all $t \in T$ to prevent time periods with more operations being weighted as more important than other time periods. Operations may change priority from one time period to another. For example, an operation such as CAS may correspond to operation $o = 1$ where $o \in O_t$ and CAS may also correspond to operation $o = 8$ where $o \in O_{t+1}$.

For each operation type $o \in O_t$, there is a list of weather element thresholds that make that operation type "red," should the weather conditions meet or exceed those thresholds. There is a separate list of weather element thresholds that make that operation type "amber," should the weather conditions fall within those thresholds. Using the ACMES gridded climatology model output, it is possible to obtain a reasonable estimate of the frequency of occurrence of these "red" and "amber" weather element thresholds.

Let r_{onmt} be the ACMES-derived percent hourly occurrence of one or more of the weather conditions that meets or exceeds one or more of the "red" weather element thresholds for operation $o \in O_t$ during time period $t \in T$ in region $m \in M$ detectable by sensor type $n \in N$. For example, suppose Table 1 is the WEM for the first time period of a military campaign ($t = 1$ where $t \in T$). CAS is the eighth highest priority operation type in the list ($o = 8$ where $o \in O_1$). If sensor type $n \in N$ can detect cloud ceilings below 1,000 feet altitude as well as surface visibility less than 3,200 meters and the

ACMES model calculates that one or both of those weather conditions occurs, on average, during 75% of the hourly intervals in the time period $t = 1$ at region $m = 4$ $(m \in M)$, the parameter $r_{8n41} = 75$. Conversely, if sensor type $b \in N$ cannot detect cloud ceilings below 1,000 feet altitude, but can detect surface visibility less than 3,200 meters and the ACMES model calculates that surface visibility drops below 3,200 meters, on average, during 50% of the hourly intervals in the time period $t = 1$ at region $m = 4$ $(m \in M)$, then the parameter $r_{8b41} = 50$.

Similarly, let y_{onmt} be the ACMES-derived percent hourly occurrence of one or more of the weather conditions that falls within one or more of the "amber" weather element thresholds for operation $o \in O_t$ during time period $t \in T$ in region $m \in M$ detectable by sensor type $n \in N$. Returning to the previous example, again suppose Table 1 is the WEM for the first time period of a military campaign ($t = 1$ where $t \in T$). CAS is the eighth highest priority operation type in the list ($o = 8$ where $o \in O_1$). If sensor type $n \in N$ can detect cloud ceilings between 1,000 and 2,000 feet altitude as well as surface visibility between 3,200 and 8,000 meters and the ACMES model calculates that one or both of those weather conditions occurs, on average, during 15% of the hourly intervals in time period $t = 1$ at region $m = 4$ $(m \in M)$, then the parameter $y_{8n41} = 15$. Alternatively, if sensor type $b \in N$ cannot detect cloud ceilings between 1,000 and 2,000 feet altitude, but can detect surface visibility between 3,200 and 8,000 meters and the ACMES model calculates that surface visibility is between 3,200 and 8,000 meters, on average, during 10% of the hourly intervals in the time period $t = 1$ at region $m = 4$ $(m \in M)$, then the parameter $y_{8b41} = 10$.

The availability of these sets of modeled climatology statistics leads to a novel formulation of the BWSSP as a series of preemptive resource allocation Binary Integer Linear Programs. For the BWSSP, the decision is which type of sensor to deploy to which region at what time. Therefore, $x_{nmt} = 1$ if a sensor type $n \in N$ is deployed to region $m \in M$ at the start of time period $t \in T$ and 0, otherwise. All of the sets, parameters and decision variables in the BWSSP are summarized in Tables 2 - 4.

Table 2. Sets in the BWSSP Model Formulation

$T = \{1, 2, \ldots, \tau\}$	Set of time periods spanned by campaign plan
$M = \{1, 2, \ldots, \mu\}$	Set of potential sensor deployment regions
$H_t \subseteq M$	Set of regions that contain secure friendly installations at time period $t \in T$
$N = \{1, 2, \ldots, \nu\}$	Set of all available weather sensor types
$B \subset N$	Set of upper air profile sensors (balloons, etc)
$E \subset N$	Set of upper air sensors that behave like radars
$L_{nt} \subseteq T$	Set of time periods sensor $n \in N$ is expected to function if deployed at the start of time period $t \in T$
S	$\nu \times \tau$ matrix of s_{nt} parameters
A	$\nu \times \mu \times \tau$ tensor of a_{nmt} parameters
O_t	Set of military operation types planned during time period $t \in T$

Table 3. Parameters in the BWSSP Model Formulation

h_{mt}	Distance to the nearest region in set H_t from region $m \in M$ during time period $t \in T$
δ_n	Effective range of or minimum spacing for sensor type $n \in N$
d_{mq}	Physical distance between region $m \in M$ and region $q \in M$
s_{nt}	Number of $n \in N$ sensors arriving in theater during time period $t \in T$
a_{nmt}	1 if sensor $n \in N$ is deployable to region $m \in M$ at the start of time period $t \in T$, 0 otherwise
p_{ot}	Weighted priority of operation type $o \in O_t$ during time period $t \in T$
r_{onmt}	ACMES-derived percent hourly occurrence of one or more of the "red" weather conditions for operation type $o \in O_t$ during time period $t \in T$ in region $m \in M$ detectable by sensor type $n \in N$
y_{onmt}	ACMES-derived percent hourly occurrence of one or more of the "amber" weather conditions for operation type $o \in O_t$ during time period $t \in T$ in region $m \in M$ detectable by sensor type $n \in N$

Table 4. Decision Variables in the BWSSP Model Formulation

x_{nmt}	1 if sensor $n \in N$ deploys to region $m \in M$ at the start of time period $t \in T$, 0 otherwise

BWSSP Objective Function Formulation

The overarching objective of the BWSSP is to maximize the weather prediction capability over the entire battlefield. Specifically, the objective of the BWSSP is to maximize detection of operationally significant inclement weather conditions as frequently as possible for the longest amount of time possible, given the available set of weather sensors. The most operationally significant inclement weather conditions are those listed in the "red" column of the WEM for each time period. The "red" conditions constitute a significant threat to the safety of forces conducting that particular type of military operation. The "amber" conditions, though also significant, are conditions that are considered to present a degradation of military capability, but are not necessarily a threat to lives or equipment. Accordingly, any formulation for the BWSSP must prioritize detection of any occurrence of "red" inclement weather conditions over the detection of any "amber" inclement weather conditions.

In an environment where the "red" and "amber" percent occurrence parameters are fairly uniform, an additional goal of any sensing strategy should be to detect these parameters as soon as they enter or leave the battlefield. Therefore, if there are automated or clandestine tactical weather sensors available for deployment outside of secure installations in this type of uniform weather environment, it is advantageous to deploy them as far from friendly installations as possible. This leads to the formulation of the BWSSP as a multi-objective BILP with three objective functions where the first objective function must be maximized preemptively before the second and where the first and second must be maximized before the third.

The "Red" Objective Function

Let u_{nmt} be the usefulness associated with deploying sensor type $n \in N$ to region $m \in M$ at the start of time period $t \in T$ in terms of only the "red" weather conditions. This parameter is equal to the sum total of the frequency of occurrences of the "red" weather conditions over the operational lifespan of the sensor for every region within the effective range of the sensor (where effective range only applies to weather radars, as they are the only weather sensors able to scan the sky over other regions), weighted by the operational significance of the "red" weather conditions. Mathematically,

$$u_{nmt} = \begin{cases} a_{nmt} \left(\sum_{l \in L_{nt}} \sum_{o \in O_l} p_{ol} \cdot r_{onml} + \sum_{\{q \in M : d_{mq} \leq \delta_n\}} \sum_{l \in L_{nt}} \sum_{o \in O_l} p_{ol} \cdot r_{onql} \right) & \text{for } n \in E \subset N, m \in M, t \in T \\ a_{nmt} \sum_{l \in L_{nt}} \sum_{o \in O_l} p_{ol} \cdot r_{onml} & \text{for } n \in N \setminus E, m \in M, t \in T \end{cases} . \quad (1)$$

Combining this usefulness parameter with the decision variables yields the objective function:

$$\max \sum_{n \in N} \sum_{m \in M} \sum_{t \in T} u_{nmt} x_{nmt} . \quad (2)$$

Objective function (2) maximizes coverage of the operationally significant "red" weather conditions over the duration of the campaign plan.

The "Amber" Objective Function

Let v_{nmt} be the usefulness associated with deploying sensor type $n \in N$ to region $m \in M$ at the start of time period $t \in T$ in terms of the "amber" weather conditions. This parameter is equal to the sum total of the frequency of occurrences of the "amber" weather conditions over the operational lifespan of the sensor for every region within the effective range of the sensor (where effective range only applies to weather radars, as they are the only weather sensors able to scan the sky over other regions), weighted by the operational significance of the "amber" weather conditions. Mathematically,

$$v_{nmt} = \begin{cases} a_{nmt}\left(\sum_{l\in L_{nt}}\sum_{o\in O_l} p_{ol}\cdot y_{onml} + \sum_{\{q\in M: d_{mq}\le\delta_n\}}\sum_{l\in L_{nt}}\sum_{o\in O_l} p_{ol}\cdot y_{onql}\right) & \text{for } n\in E\subset N, m\in M, t\in T \\ a_{nmt}\sum_{l\in L_{nt}}\sum_{o\in O_l} p_{ol}\cdot y_{onml} & \text{for } n\in N\setminus E, m\in M, t\in T \end{cases}. \quad (3)$$

Combining this usefulness parameter with the decision variables yields the objective function:

$$\max \sum_{n\in N}\sum_{m\in M}\sum_{t\in T} v_{nmt}x_{nmt}. \quad (4)$$

Objective function (4) maximizes coverage of the operationally significant "amber" weather conditions over the duration of the campaign plan.

The Distance from Friendly Installation Objective Function

Recall h_{mt} is defined as the distance to the nearest region in set H_t from region $m \in M$ during time period $t \in T$. Therefore, the objective function that maximizes the distance from the nearest friendly installation for any sensor type that does not require a secure installation is:

$$\max \sum_{t\in T}\sum_{m\in M\setminus H_t} h_{mt}\sum_{n\in N} a_{nmt}\cdot x_{nmt}. \quad (5)$$

Combined Objective Function

Objective functions (2), (4) and (5) form a multi-objective BILP with three objective functions where objective function (2) must be maximized preemptively before objective function (4) and objective function (4) must be maximized preemptively before objective function (5). As stated in Chapter II, the BWSSP is NP-Hard and, in the worst case, cannot be solved in polynomial time (assuming $P \neq NP$). If each objective function is maximized one after the other, each BWSSP optimal solution would require executing three exponential algorithms.

This hazard could be avoided by solving the BWSSP as a standard multi-objective problem. However, the BWSSP objective functions are preemptive, not complementary (Das 1997). There is no tradeoff between the value of objective function (2) and objective function (4) or objective function (4) and objective function (5). Any solution that does not optimize objective function (2) is sub-optimal in the BWSSP.

If the three preemptive objective functions (2), (4) and (5) can be combined into a single, combined objective function in polynomial time, optimal solutions to the BWSSP can generally be found more efficiently. Sherali (1982) describes a simple algorithm that combines multiple, preemptive objective functions into a single objective function in polynomial time that maintains the preemptive nature of the original objective functions. Applying the Sherali algorithm to objective functions (2), (4) and (5) yields the objective function coefficients:

$$b_{nmt} = \left(1 + \sum_{t \in T} \sum_{m \in M \setminus H_t} \left|h_{mt}\right| \sum_{n \in N} \left|a_{nmt}\right|\right) \cdot v_{nmt} + h_{mt} \cdot a_{nmt} \quad \text{for } n \in N, m \in M, t \in T. \quad (6)$$

$$\pi_{nmt} = \left(1 + \sum_{n \in N} \sum_{m \in M} \sum_{t \in T} \left|b_{nmt}\right|\right) \cdot u_{nmt} + b_{nmt} \qquad \text{for } n \in N, m \in M, t \in T. \quad (7)$$

This yields the combined objective function:

$$\max \sum_{n \in N} \sum_{m \in M} \sum_{t \in T} \pi_{nmt} x_{nmt}. \quad (8)$$

The Sherali algorithm is a straightforward method for combining the three preemptive objective functions of the BWSSP into a single objective function that maintains the preemptive nature of the original objective functions. However, this method has its drawbacks. Combining the preemptive objective functions into a single objective function may cause the loss of some equally optimal solutions to the original set of preemptive objective functions. If discovering all possible optimal solutions is more valuable than obtaining a single optimal solution quickly, then this method may not be suitable. Ignizio and Thomas (1984) highlight an additional potential drawback of the Sherali algorithm that is applicable in the BWSSP formulation. Namely, values for π_{nmt} can become very large relative to the coefficients in the original objective functions. This brings in the potential for integer overflow errors or difficulty obtaining an optimal solution using commercial software such as the Excel® Premium Solver® Version 9.

Ignizio and Thomas (1984) propose a modification to the Sherali algorithm that compensates for this last potential drawback. Applying their modification to the Sherali algorithm, we wish to find a (p – 1)-dimensional vector (where p is the number of objective functions) of integer weights $\boldsymbol{\lambda}$ that minimizes the magnitude of the largest coefficient of the combined objective function. Under their algorithm, objective function (5) becomes:

$$\max \sum_{n\in N}\sum_{m\in M}\sum_{t\in T}\left(h_{mt}\cdot a_{nmt}-\lambda_2\cdot v_{nmt}\right)\cdot x_{nmt}\ . \quad (9)$$

If the Sherali algorithm is then applied to objective functions (4) and (9):

$$b'_{nmt}=\left(1+\sum_{n\in N}\sum_{m\in M}\sum_{t\in T}\left|h_{mt}\cdot a_{nmt}-\lambda_2\cdot v_{nmt}\right|\right)\cdot v_{nmt}+\left(h_{mt}\cdot a_{nmt}-\lambda_2\cdot v_{nmt}\right)\ \text{ for } n\in N, m\in M, t\in T\ . \quad (10)$$

The optimal value for the weight λ_2 is determined by finding:

$$\lambda_2^* = \min_{\lambda_2 \in \mathbb{Z}^*} \left\{ \max_{n \in N, m \in M, t \in T} \left\{ b'_{nmt} \right\} \right\}. \quad (11)$$

Equation (10) is convex in terms of λ_2, so the value for λ_2^* can be found using a discrete search scheme (Ignizio and Thomas 1984). Ignizio and Thomas (1984) suggest a Fibonacci search, but any efficient discrete search is sufficient. An efficient limiting form of the Fibonacci search is the Golden Section Method (Snyman 2005). In this research, the Improved Golden Section Method developed by Den Boef and Den Hertog (2007) is used. This method finds the optimal value for a known convex (or concave) black-box function using the Golden Section Method while taking advantage of the convexity of the function to reduce the number of required function evaluations (Den Boef and Den Hertog 2007). This method was chosen because it is at least as efficient as the Golden Selection Method and it does not require function gradient information (Den Boef and Den Hertog 2007). The Improved Golden Search Method is intended for continuous functions. Therefore, in this research, the nearest integer to the continuous optimal value found via the Improved Golden Section Method that has the smallest value for equation (11) is used as the value for λ_2^*. The VBA® code used to implement this algorithm is described in Appendix A.

Substituting in λ_2^* yields a new combined objective function that combines objective functions (4) and (5):

$$\max \sum_{n \in N} \sum_{m \in M} \sum_{t \in T} b'_{nmt} \, x_{nmt}. \quad (12)$$

Ignizio and Thomas (1984) can then be applied to combine preemptive objective functions (2) and (12) into a single objective function for the BWSSP that meets the preemptive objectives of objective functions (2), (4) and (5). Applying Ignizio and Thomas (1984), objective function (12) becomes:

$$\max \sum_{n \in N} \sum_{m \in M} \sum_{t \in T} \left(b'_{nmt} - \lambda_1 \cdot u_{nmt} \right) \cdot x_{nmt} \ . \quad (13)$$

Applying the Sherali algorithm to objective functions (2) and (13) gives:

$$c_{nmt} = \left(1 + \sum_{n \in N} \sum_{m \in M} \sum_{t \in T} \left| b'_{nmt} - \lambda_1 \cdot u_{nmt} \right| \right) \cdot u_{nmt} + \left(b'_{nmt} - \lambda_1 \cdot u_{nmt} \right). \quad (14)$$

The optimal value for the weight λ_1 is determined by finding:

$$\lambda_1^* = \min_{\lambda_1 \in \mathbb{Z}^*} \left\{ \max_{n \in N, m \in M, t \in T} \left\{ c_{nmt} \right\} \right\}. \quad (15)$$

Equation (14) is also convex in terms of λ_1, so the value for λ_1^* can also be found using the Improved Golden Section Method (Den Boef and Den Hertog 2007). This yields a new combined objective function that maintains the preemptive nature of objective functions (2), (4) and (5) while minimizing the magnitude of the combined objective function coefficients:

$$\max \sum_{n \in N} \sum_{m \in M} \sum_{t \in T} c_{nmt} x_{nmt} \ . \quad (16)$$

BWSSP Constraints

The first set of constraints required for the BWSSP ensure that no more than one surface sensor of any type is deployed to any region over the entire planning time period:

$$\sum_{n \in N \setminus \{B \cup E\}} \sum_{t \in T} x_{nmt} \leq 1 \text{ for each } m \in M \ . \quad (17)$$

Constraint (17) is summed over every time period of the campaign because the statistical distribution of the lifespan of a surface sensor is either very complex or unknown. The expected lifespan provides a reasonable estimate of sensor performance, but the standard deviation about the expected value can be very large. In the case of a clandestine sensor, there are also operational concerns. Returning to the site of a clandestine sensor would risk discovery by an adversary, thereby compromising the sensor and the personnel attempting to emplace it. As such, the BWSSP formulation assumes that any sensor, once deployed, will not be replaced for the rest of the planning time period. In other words, for all $x_{nmt} = 1$, $x_{nmt'} = 0$ for all $t' > t$ where $t \in T$ and $t' \in T$. It is assumed that the SMO will re-optimize the BWSSP instance in the event of the loss of a sensor in order to determine whether or not the current conditions and CONOPS advise replacement of the lost sensor.

The number of upper-air weather sensors and weather radars deployed to any particular region must also be restricted to one per each upper-air sensor type and weather radar type. In other words, a weather radar and weather balloon system may be deployed to the same region, but two weather radars or two upper-air sensors may not. This implies the following two constraints:

$$\sum_{n \in B} \sum_{t \in T} x_{nmt} \leq 1 \text{ for each } m \in M, \quad (18)$$

$$\sum_{n \in E} \sum_{t \in T} x_{nmt} \leq 1 \text{ for each } m \in M. \quad (19)$$

The next set of constraints ensures that the total number of sensors deployed does not exceed the total number of sensors available at each time period:

$$\sum_{m \in M} x_{nm1} \leq s_{n1} \text{ for each } n \in N \quad (20), \text{ and}$$

$$\sum_{m\in M} x_{nmt} - \sum_{k=1}^{t-1}\left(s_{nk} - \sum_{m\in M} x_{nmk}\right) \le s_{nt} \text{ for each } n \in N \text{ and } \{t \in T : t \ge 2\}. \quad (21)$$

Another consideration that must be accounted for in the BWSSP is the effective range of the sensors defined by the parameter δ_n. Air Force Manual 15-111 *Surface Weather Observations* (2008) defines a surface weather observation as "a measurement or evaluation (manual, automated, or augmented) of one or more meteorological elements that describe the state of the atmosphere at the region where the observation is taken." Surface observations are point observations, meaning the effective range of any surface weather sensor of any type is thought of as very small relative to the resolution of any climatological model. Therefore, the effective range of any surface weather sensor is just the small region where it is located, thereby implying that $\delta_n = 0$ for $n \in N \setminus \{E \cup B\}$.

Upper air observations that provide a vertical profile of the atmosphere, such as those provided by weather balloons, are also point observations. However, Federal Meteorological Handbook No. 3 - *Rawinsonde and Pibal Observations* (FMH-3 1997) suggests that these types of observing stations be spaced at least 250 km apart over large land regions and 1000 km over sparsely populated and oceanic regions. As such, constraint (22) is added as a constraint in situations where the suggestion in FMH-3 (1997) is followed.

$$\sum_{t\in T} x_{nmt} + \sum_{t\in T} x_{bqt} \le 1 \text{ for each } (n,b) \in B \text{ and } \{(m,q) \in M : d_{mq} < \max\{\delta_b, \delta_n\}\} \quad (22)$$

Weather radars of any type sweep the atmosphere over a relatively wide region and cannot be placed within the effective range of each other because of interference. Constraint (23) ensures no two radars are placed within each other's effective range.

$$\sum_{t \in T} x_{nmt} + \sum_{t \in T} x_{bqt} \leq 1 \text{ for each } (n,b) \in E \text{ and } \{(m,q) \in M : d_{mq} < \delta_b + \delta_n\} \quad (23)$$

As noted in Chapter II, FMH-3 (1997) mandates upper-air weather profile sensors be calibrated with surface weather sensors prior to launch. Though weather radars do not fall under this requirement, any METOC collection plan should co-locate any weather radar with a surface sensor to ensure that weather personnel are providing the maximum amount of information they can from any weather observation region. Constraints (24) and (25) ensure that upper-air sensors like weather balloons and weather radars are co-located with surface weather sensors.

$$\sum_{n \in B} x_{nmt} - \sum_{n \in N \setminus \{B \cup E\}} \sum_{k=1}^{t} x_{nmk} \leq 0 \text{ for each } m \in M, t \in T \quad (24)$$

$$\sum_{n \in E} x_{nmt} - \sum_{n \in N \setminus \{B \cup E\}} \sum_{k=1}^{t} x_{nmk} \leq 0 \text{ for each } m \in M, t \in T \quad (25)$$

Placing these constraints on objective function (16) yields the BWSSP.

BWSSP

$$\max \sum_{n\in N}\sum_{m\in M}\sum_{t\in T} c_{nmt}x_{nmt} \qquad (16)$$

Subject to: $$\sum_{n\in N\setminus\{B\cup E\}}\sum_{t\in T} x_{nmt} \le 1 \text{ for each } m\in M \qquad (17)$$

$$\sum_{n\in B}\sum_{t\in T} x_{nmt} \le 1 \text{ for each } m\in M \qquad (18)$$

$$\sum_{n\in E}\sum_{t\in T} x_{nmt} \le 1 \text{ for each } m\in M \qquad (19)$$

$$\sum_{m\in M} x_{nm1} \le s_{n1} \text{ for each } n\in N \qquad (20)$$

$$\sum_{m\in M} x_{nmt} - \sum_{k=1}^{t-1}\left(s_{nk} - \sum_{m\in M} x_{nmk}\right) \le s_{nt} \text{ for each } n\in N \text{ and } \{t\in T : t\ge 2\} \qquad (21)$$

$$\sum_{t\in T} x_{nmt} + \sum_{t\in T} x_{bqt} \le 1 \text{ for each } (n,b)\in B \text{ and } \{(m,q)\in M : d_{mq} < \max\{\delta_b,\delta_n\}\} \qquad (22)$$

$$\sum_{t\in T} x_{nmt} + \sum_{t\in T} x_{bqt} \le 1 \text{ for each } (n,b)\in E \text{ and } \{(m,q)\in M : d_{mq} < \delta_b + \delta_n\} \qquad (23)$$

$$\sum_{n\in B} x_{nmt} - \sum_{n\in N\setminus\{B\cup E\}}\sum_{k=1}^{t} x_{nmk} \le 0 \text{ for each } m\in M, t\in T \qquad (24)$$

$$\sum_{n\in E} x_{nmt} - \sum_{n\in N\setminus\{B\cup E\}}\sum_{k=1}^{t} x_{nmk} \le 0 \text{ for each } m\in M, t\in T \qquad (25)$$

$$x_{nmt} \in \{0,1\} \quad \forall\ n\in N, m\in M, t\in T$$

Solving the BWSSP

As discussed in Chapter II, the BWSSP where $\tau = 1$ is an instance of the SSP, which is NP-hard (Garey and Johnson 1979). Therefore, the general BWSSP is at least as hard as the SSP meaning that the BWSSP is NP-hard and worst case instances cannot be solved in polynomial time (assuming $P \neq NP$). Building the BWSSP constraints requires $O\left(\mu^2 \cdot \nu \cdot \tau\right)$ operations. Therefore, the BWSSP is a difficult combinatorial optimization problem.

In the next chapter, several BWSSP instances are created and solved to optimality using Excel 2007® with Premium Solver® Version 9. The instances are examined to determine if the combined objective function (16) maintains the preemptive nature of the original preemptive objective functions (2), (4) and (5) while solving the BWSSP more efficiently. Solutions are also examined to determine if the BWSSP creates a realistic METOC collection plan within a reasonable amount of time.

Interpretation of a BWSSP Optimal Solution

An optimal solution to the BWSSP is a METOC collection plan that places the available weather sensors at a set of regions $M^* \subseteq M$ in a schedule over the campaign time periods $t \in T$ that ensures all sensor types $n \in N$ are detecting operationally significant weather conditions as frequently as possible as far away from friendly installations as possible for the duration of their expected operational lifespans. This METOC collection plan is described by the set of decision variables equal to 1 in the

BWSSP optimal solution. Any value $x_{nmt} = 1$ in the solution indicates that one sensor type $n \in N$ should be deployed to region $m \in M$ at the start of time period $t \in T$.

Operationally significant weather conditions are viewed as a threat to friendly forces. Therefore, a METOC collection plan that detects those conditions as frequently as possible is most desirable. The value of the METOC collection plan created by an optimal solution to the BWSSP can be quantified in terms of the contribution it provides to the overall commander's CONOPS at each phase of the military campaign.

Recall that the coefficients of objective functions (2) and (4) are created from ACMES-derived percent hourly occurrences of weather conditions that either significantly or marginally impact friendly forces' ability to execute the operations in the CONOPS for each phase of the military campaign (operations $o \in O_t$ over all time periods $t \in T$). For every region $m \in M$ such that $\sum_{t \in T} \sum_{n \in N} x_{nmt} \geq 1$ in the BWSSP solution, a summary can be created depicting the number of hours each sensor type $n \in N$ at each region $m \in M$ can be expected to detect weather conditions that pose either a significant or marginal threat to friendly forces' ability to execute operations $o \in O_t$ during a time period $t \in T$. This summary relates the METOC collection plan to increased situational awareness on the battlefield, thereby increasing friendly forces' ability to conduct operations in favorable weather conditions that an adversary may not be aware of.

This analysis is useful during all four steps of JIPOE: define the battlefield environment, describe the battlefield's effects, evaluate the adversary and determine adversary potential courses of action (JPub 2-01.3 2000). General outlines of this type of analysis are given in Tables 5 and 6 below.

Table 5. BWSSP Solution Value as METOC Collection Plan at Time Period $t \in T$ (Significant "Red" Weather Conditions)

Expected Hours METOC Collection Plan Will Find "Red" Conditions For Operations Planned During Time Period $t \in T$			
	Regions		
(θ = Hours in time period t)	$q \in M : \sum_{k=1}^{t} \sum_{n \in N} x_{nqk} \geq 1$	$p \in M : \sum_{k=1}^{t} \sum_{n \in N} x_{npk} \geq 1$	……
Operation 1 $\in O_t$	$\frac{\theta \cdot \max_{n \in N} \left\{ r_{1nqt} \cdot \sum_{k=1}^{t} x_{nqk} \right\}}{100}$	$\frac{\theta \cdot \max_{n \in N} \left\{ r_{1npt} \cdot \sum_{k=1}^{t} x_{npk} \right\}}{100}$	……
Operation 2 $\in O_t$	$\frac{\theta \cdot \max_{n \in N} \left\{ r_{2nqt} \cdot \sum_{k=1}^{t} x_{nqk} \right\}}{100}$	$\frac{\theta \cdot \max_{n \in N} \left\{ r_{2npt} \cdot \sum_{k=1}^{t} x_{npk} \right\}}{100}$	……
. . .	. . .	. . .	……

Table 6. BWSSP Solution Value as METOC Collection Plan at Time Period $t \in T$ (Marginal "Amber" Weather Conditions)

Expected Hours METOC Collection Plan Will Find "Amber" Conditions For Operations Planned During Time Period $t \in T$			
	Regions		
(θ = Hours in time period t)	$q \in M : \sum_{k=1}^{t} \sum_{n \in N} x_{nqk} \geq 1$	$p \in M : \sum_{k=1}^{t} \sum_{n \in N} x_{npk} \geq 1$	……
Operation 1 $\in O_t$	$\frac{\theta \cdot \max_{n \in N} \left\{ y_{1nqt} \cdot \sum_{k=1}^{t} x_{nqk} \right\}}{100}$	$\frac{\theta \cdot \max_{n \in N} \left\{ y_{1npt} \cdot \sum_{k=1}^{t} x_{npk} \right\}}{100}$	……
Operation 2 $\in O_t$	$\frac{\theta \cdot \max_{n \in N} \left\{ y_{2nqt} \cdot \sum_{k=1}^{t} x_{nqk} \right\}}{100}$	$\frac{\theta \cdot \max_{n \in N} \left\{ y_{2npt} \cdot \sum_{k=1}^{t} x_{npk} \right\}}{100}$	……
. . .	. . .	. . .	……

Returning to the previous CAS example, again suppose Table 1 is the WEM for the first time period of a military campaign ($t = 1$ where $t \in T$). CAS is the eighth highest priority operation type in the list (o = 8 where $o \in O_1$). If sensor type $n \in N$ can detect cloud ceilings below 1,000 feet altitude as well as surface visibility less than 3,200 meters and the ACMES model calculates that one or both of those weather conditions occurs, on average, during 75% of the hourly intervals in time period t = 1 at region m = 4 $(m \in M)$, then the parameter $r_{8n41} = 75$. Alternatively, if sensor type $b \in N$ cannot detect cloud ceilings below 1,000 feet altitude, but can detect surface visibility less than 3,200 meters and the ACMES model calculates that surface visibility drops below 3,200 meters, on average, during 50% of the hourly intervals in time period t = 1 at region m = 4 $(m \in M)$, then the parameter $r_{8b41} = 50$. If a sensor type $b \in N$ and a sensor type $n \in N$ are operating at region m = 4 $(m \in M)$ during time period t = 1 $(t \in T)$, Table 5 would have the following values:

Table 7. Example BWSSP Solution as METOC Collection Plan at Time Period 1 (Significant "Red" Weather Conditions)

Expected Hours METOC Collection Plan Will Find "Red" Conditions For Operations Planned During Time Period 1			
(744 Hours in time period 1)	Regions		
	……………..	$m = 4$	……
. . .	. . .	. . .	……
CAS $(o = 8,\ o \in O_1)$	……………..	$\frac{744 \cdot \max\{r_{8n41} \cdot 1, r_{8b41} \cdot 1\}}{100} = \frac{744 \cdot \max\{75, 50\}}{100} = 558$	……
. . .	. . .	. . .	……

Staying with the previous CAS example, again suppose Table 1 is the WEM for the first time period of a military campaign (t = 1 where $t \in T$). CAS is the eighth highest priority operation type in the list (o = 8 where $o \in O_1$). If sensor type $n \in N$ can detect cloud ceilings between 1,000 and 2,000 feet altitude as well surface visibility between 3,200 and 8,000 meters and the ACMES model calculates that one or both of those weather conditions occurs, on average, during 15% of the hourly intervals in time period t = 1 at region m = 4 ($m \in M$) , then the parameter $y_{8n41} = 15$. Alternatively, if sensor type $b \in N$ cannot detect cloud ceilings between 1,000 and 2,000 feet altitude, but can detect surface visibility between 3,200 and 8,000 meters and the ACMES model calculates that surface visibility is between 3,200 and 8,000 meters, on average, during 10% of the hourly intervals in time period t = 1 at region m = 4 ($m \in M$) , then the parameter $y_{8b41} = 10$. If a sensor type $b \in N$ and a sensor type $n \in N$ are operating at region m = 4 $(m \in M)$ during time period t = 1 $(t \in T)$, Table 6 would have the following values:

Table 8. Example BWSSP Solution as METOC Collection Plan at Time Period 1 (Significant "Amber" Weather Conditions)

Expected Hours METOC Collection Plan Will Find "Amber" Conditions For Operations Planned During Time Period 1			
(744 Hours in time period 1)	Regions		
	………..	$m = 4$	……
. . .	. . .	. . .	……
CAS $(o = 8,\ o \in O_1)$	………… …..	$\frac{744 \cdot \max\{y_{8n41} \cdot 1, y_{8b41} \cdot 1\}}{100} = \frac{744 \cdot \max\{15,10\}}{100} = 111.6$	……
. . .	. . .	. . .	……

Also recall the coefficients of objective function (5) are the distances from a particular region $m \in M$ to the nearest friendly installation $q \in H_t \subseteq M$. If the average speed of large-scale weather systems is known, the distances h_{mt} for every region $m \in M$ such that $\sum_{t \in T} \sum_{n \in N} x_{nmt} \geq 1$ can be used to estimate the change in average time from detection to arrival of significant weather conditions at friendly installations at each time period $t \in T$.

BWSSP Sensitivity Analysis

Optimal solutions to different instances of the BWSSP can be compared to determine the merits of one METOC collection plan versus another. Comparisons may be made over any combination of the dimensions in the BWSSP: time, sensor type and region. If any parameter s_{nt} for $n \in N$ and $t \in T$ is modified or if a region is removed from either M or H_t for some time period $t \in T$, the degradation or improvement of the METOC collection plan due to that change can be quantified by the difference in the number of hours of expected operationally significant weather occurrences for each operation $o \in O_t$ over all time periods $t \in T$ using the appropriate r_{onmt} and y_{onmt} values.

Again returning to the previous CAS example, suppose a SMO needs to demonstrate the value of acquiring the sensor type $n \in N$ slated for allocation to region $m = 4$ $(m \in M)$ during time period t = 1 $(t \in T)$. If after re-optimizing the BWSSP with the value s_{n1} decreased by one, the sensor type $b \in N$ remains feasible for allocation to region $m = 4$ $(m \in M)$ and the new optimal solution contains the values $x_{b41} = 1$ and $x_{n41} = 0$, the values in Tables 7 and 8 change to:

Table 9. Modified BWSSP Solution/METOC Collection Plan at Time Period 1
(Significant "Red" Weather Conditions)

Expected Hours METOC Collection Plan Will Find "Red" Conditions For Operations Planned During Time Period 1			
(744 Hours in time period 1)	Regions		
	…………..	$m = 4$	……
. . .	. . .	. . .	……
CAS $(o = 8,\ o \in O_1)$	……………..	$\frac{744 \cdot \max\{r_{8b41} \cdot 1\}}{100} = \frac{744 \cdot 50}{100} = 372$	……
. . .	. . .	. . .	……

Table 10. Modified BWSSP Solution/METOC Collection Plan at Time Period 1
(Significant "Amber" Weather Conditions)

Expected Hours METOC Collection Plan Will Find "Amber" Conditions For Operations Planned During Time Period 1			
(744 Hours in time period 1)	Regions		
	……………..	$m = 4$	……
. . .	. . .	. . .	……
CAS $(o = 8,\ o \in O_1)$	……………..	$\frac{744 \cdot \max\{y_{8b41} \cdot 1\}}{100} = \frac{744 \cdot 10}{100} = 74.4$	……
. . .	. . .	. . .	……

Therefore, failing to acquire the sensor type $n \in N$ for allocation to region $m = 4$ $(m \in M)$ during time period t = 1 $(t \in T)$ causes a loss of 558 – 372 = 186 hours where friendly forces can expect to detect presently occurring "red" weather conditions for CAS and a loss of 111.6 – 74.4 = 37.2 hours where friendly forces can expect to detect presently occurring "amber" weather conditions for CAS. The same procedure can be easily performed in reverse to determine the value of gaining a new sensor type $n \in N$ at time period $t = 1$ $(t \in T)$.

Using this methodology, the SMO can analyze any changes in the BWSSP scenario by re-optimizing with the new parameter settings and comparing the new optimal solution to the old optimal solution using the number of expected hours of operationally significant weather detection gained or lost at each region $m \in M$. In summary, this analysis provides the overall commander with a quantifiable metric for evaluating how any changes to the METOC collection plan can impact his or her ability to proceed with the campaign plan during each time period $t \in T$.

Conclusion

This chapter presented the general formulation of the BWSSP. The parameters, sets and decision variables in the BWSSP were explained in terms of readily available data. These parameters were formulated first as a series of preemptive BILP objective functions and then as a combined, single BILP that maintained the preemptive nature of the original objectives. The constraints on the BWSSP BILP were described in detail in terms of the attributes of the BWSSP parameters. BWSSP complexity was discussed with a preview of the approach taken in Chapter IV to solve several BWSSP instances to optimality. Optimal solutions to the BWSSP were described as a METOC collection plan in terms of the BWSSP parameters. The chapter concluded with an approach to sensitivity analysis of differing BWSSP optimal solutions based on the improvement or degradation of a METOC collection plan in terms of the BWSSP parameters.

IV. Results

Introduction

In this chapter, several BWSSP instances are generated using VBA® code and solved to optimality using Excel 2007® with Premium Solver® Version 9. Each BWSSP instance is configured with parameter settings that depict realistic military campaign scenarios. The instances are used to verify that the combined objective function (16) maintains the preemptive nature of the original preemptive objective functions (2), (4) and (5) while solving the BWSSP more efficiently. Solutions to large instances of the BWSSP are also examined to determine if the BWSSP creates a realistic METOC collection plan within a reasonable amount of time. Sensitivity analysis is conducted on a single, smaller BWSSP instance to illustrate BWSSP solution interpretation as a METOC collection plan. Specifically, this chapter examines the overall effectiveness of the BWSSP as a model for creating a METOC collection plan.

Combined Objective Function Efficiency

This section examines the efficiency of solving the BWSSP using the combined objective function (16) versus preemptively solving objective functions (2), (4) and (5).

BWSSP Instance Generation

A total of 49 different BWSSP instances were created to test the computational efficiency and accuracy of objective function (16) versus preemptively solving objective functions (2), (4) and (5). The parameters of the BWSSP were generated from a list of input values. Some input values defined the specific value of a BWSSP parameter.

Other input values acted as an upper bound on a statistical distribution that created the randomized value for a BWSSP parameter.

The battlefield for each instance is defined by eight input parameters. The climate conditions simulate weather patterns in different parts of the globe. Climate conditions were defined as being either "Harsh," "Moderate" or "Tropical." A "Harsh" climate means that all of the r_{onmt} parameters were randomly generated as integer values from a uniform distribution between 25 and 50 for all $o \in O_t$, $n \in N$, $m \in M$ and $t \in T$. The corresponding y_{onmt} parameters were randomly generated as integers from a uniform distribution between 0 and (100 – r_{onmt}) for all $o \in O_t$, $n \in N$, $m \in M$ and $t \in T$. The "Harsh" climate was used to test scenarios where the primary objective function (2) coefficients are generally much larger than the secondary objective function (4) and tertiary objective function (5) coefficients.

A "Moderate" climate means that all of the r_{onmt} parameters were randomly generated as integer values from a uniform distribution between 0 and 25 for all $o \in O_t$, $n \in N$, $m \in M$ and $t \in T$. Each corresponding y_{onmt} parameter was randomly generated as an integer from a uniform distribution between 0 and (50 – r_{onmt}) for all $o \in O_t$, $n \in N$, $m \in M$ and $t \in T$. The "Moderate" climate was used to test scenarios where the secondary objective function (4) coefficients are generally much larger than the primary objective function (2) and tertiary objective function (5) coefficients.

A "Tropical" climate means that all of the r_{onmt} parameters were randomly generated as integer values from a uniform distribution between 0 and 15 for all $o \in O_t$, $n \in N$, $m \in M$ and $t \in T$. Each corresponding y_{onmt} parameter was randomly generated as an integer from a uniform distribution between 0 and ($15 - r_{onmt}$) for all $o \in O_t$, $n \in N$, $m \in M$ and $t \in T$. The "Tropical" climate was used to test scenarios where the tertiary objective function (5) coefficients are approximately the same order of magnitude as the primary objective function (2) and secondary objective function (4) coefficients.

The "Campaign Type" is a notional military campaign that would feature a particular mix of operation types and time periods. "Humanitarian Assistance" and "Peacekeeping" campaigns are of shorter duration and feature fewer operation types during each time period $t \in T$. "Conventional Warfare" features more operations over a longer set of time periods $t \in T$. "Advise/Train" campaigns are of longer duration, but feature fewer friendly installations or bases. "Pre-assault" campaigns are of very short duration (one time period) and weigh heavier towards clandestine weather sensors. This list is based on the author's personal experience as a staff weather officer participating in military operational planning. This is by no means the full list of all possible military campaign scenarios nor is it precisely how any campaign may be executed. The list is purely notional and the scenarios are selected to test the BWSSP's performance with a variety of parameter settings close to those commonly seen in real-world planning.

"Total Ops" is the total number of operations possible for a period $t \in T$. The size of each set O_t for all $t \in T$ was selected as a random integer value from a uniform distribution between 1 and the "Total Ops" value. "Model Points" and "Model Res" create a simulated ACMES model window covering an area the size of the "Ops Area Size." The parameter "Model Points" represents the number of simulated ACMES model data points in the "model" window, all of which are evenly spaced at the interval indicated by the "Model Res" parameter. A "Large" "Ops Area Size" is an area bigger than 200,000 square kilometers, a “Medium” “Ops Area Size” is an area between 30,000 and 70,000 square kilometers and the “Small” “Ops Area Size” is intended to represent an area between 6,000 and 20,000 square kilometers.

The “Number Periods” parameter determines the number of time periods in the BWSSP instance, τ. The “Number Bases” parameter indicates the number of friendly installations randomly placed within the simulated ACMES model window. The friendly installations are placed randomly in the ACMES model window by selecting simulated ACMES model window points at random from a uniform distribution until all installations are placed at different ACMES data points.

The parameters in A are determined by the sensor type. Sensors requiring a secure friendly location (fixed base) have $a_{nmt} = 1$ for all $m \in H_t$ for each $t \in T$ and $a_{nmt} = 0$ for all $m \notin H_t$ for each $t \in T$. For clandestine sensors, $a_{nmt} = 1$ for all $n \in N$, $m \in M$ and $t \in T$. Decision variables with objective function (16) coefficients equal to 0 were removed from the problem instance.

The settings for all of these input parameters are listed in Table 11 below:

Table 11. Parameters Defining the Simulated Battlefield for BWSSP Test Instances

Test Instance	Climate Conditions	Campaign Type	Operations Area Size	Total Ops	Number Regions	Model Res (km)	Number Periods	Number Bases
1	Harsh	Advise/Train	Large	15	130	45	6	5
2	Tropical	Advise/Train	Large	15	130	45	6	5
3	Moderate	Advise/Train	Large	15	130	45	6	5
4	Harsh	Advise/Train	Large	15	130	45	6	5
5	Tropical	Advise/Train	Large	15	130	45	6	5
6	Moderate	Advise/Train	Large	15	130	45	6	5
7	Moderate	Advise/Train	Medium	15	160	15	6	3
8	Moderate	Advise/Train	Small	15	160	7	6	2
9	Harsh	Advise/Train	Medium	15	160	15	6	3
10	Tropical	Advise/Train	Small	15	160	7	6	2
11	Tropical	Advise/Train	Medium	15	160	15	6	3
12	Harsh	Advise/Train	Small	15	160	7	6	2
13	Harsh	Peacekeeping	Large	6	220	45	6	10
14	Moderate	Peacekeeping	Large	6	220	45	6	10
15	Moderate	Advise/Train	Medium	15	160	15	6	3
16	Harsh	Advise/Train	Medium	15	160	15	6	3
17	Tropical	Advise/Train	Medium	15	160	15	6	3
18	Harsh	Peacekeeping	Large	6	300	45	6	10
19	Moderate	Advise/Train	Small	15	170	7	6	2
20	Tropical	Advise/Train	Small	15	170	7	6	2
21	Harsh	Advise/Train	Small	15	170	7	6	2
22	Tropical	Peacekeeping	Large	6	300	45	6	10
23	Moderate	Peacekeeping	Large	6	300	45	6	10
24	Harsh	Humanitarian Assistance	Large	4	540	45	6	25
25	Harsh	Humanitarian Assistance	Large	4	540	45	6	25
26	Tropical	Pre-Assault	Large	15	992	45	1	5
27	Moderate	Pre-Assault	Large	15	992	45	1	5
28	Tropical	Peacekeeping	Medium	6	290	15	6	5
29	Tropical	Peacekeeping	Medium	6	280	15	6	5
30	Moderate	Peacekeeping	Medium	6	280	15	6	5
31	Harsh	Peacekeeping	Medium	6	280	15	6	5
32	Tropical	Peacekeeping	Small	6	320	7	6	3
33	Moderate	Peacekeeping	Medium	6	290	15	6	5
34	Moderate	Peacekeeping	Small	6	320	7	6	3
35	Harsh	Peacekeeping	Medium	6	290	15	6	5
36	Tropical	Humanitarian Assistance	Large	4	540	45	6	25
37	Tropical	Humanitarian Assistance	Large	4	540	45	6	25
38	Moderate	Humanitarian Assistance	Large	4	540	45	6	25
39	Moderate	Humanitarian Assistance	Large	4	540	45	6	25
40	Harsh	Peacekeeping	Small	6	320	7	6	3
41	Harsh	Peacekeeping	Small	6	320	7	6	3
42	Harsh	Pre-Assault	Medium	15	995	15	1	3
43	Harsh	Conventional Warfare	Large	15	540	45	12	55
44	Harsh	Conventional Warfare	Large	15	540	45	13	50
45	Tropical	Pre-Assault	Medium	15	995	15	1	3
46	Moderate	Pre-Assault	Medium	15	995	15	1	3
47	Tropical	Conventional Warfare	Large	15	540	45	12	55
48	Tropical	Conventional Warfare	Large	15	540	45	13	50
49	Moderate	Conventional Warfare	Large	15	540	45	13	50

The set N consists of at most one type of clandestine surface weather sensor, one type of fixed base surface weather sensor, one type of clandestine upper-air weather sensor, one type of fixed base upper-air weather sensor and one type of weather radar. The operational lifespans (in time periods) for each sensor type $n \in N$ were chosen based on estimates of real-world behavior. The values were reduced in "Harsh" climates to capture the degradation in sensor capability due to wear and tear. A sensor lifespan equal to 0 indicates that the sensor type was not included in that particular problem instance.

The s_{nt} parameters indicating the number of sensors of any type $n \in N$ arriving at the start of time period $t \in T$, were determined by a draw from a uniform distribution between 0 and the "Max Sensor" value for each sensor type $n \in N$ and time period $t \in T$. The upper-air weather sensor spacing suggested by FMH-3 (1997) was observed in scenarios where it was appropriate. The effective range of the weather radar in each scenario was selected to be 150 kilometers in "Harsh" climates and 220 kilometers in other climates. This is very roughly based on the effective range of standard weather radars, though many factors influence the true effective range of any radar (Ahrens 2000). This difference in effective ranges models the change in radar settings due to weather conditions.

Table 12. Weather Sensor Parameters for BWSSP Test Instances

Test Instance	Clandestine Surface Lifespan	Base Surface Lifespan	Base Upper-Air Lifespan	Clandestine Upper-Air Lifespan	Radar Lifespan	Max Sensors	Upper-Air Spacing	Radar Range
1	3	4	2	0	6	20	0	150
2	7	9	6	0	12	20	0	220
3	5	6	4	0	8	20	0	220
4	3	4	2	0	6	20	250	150
5	7	9	6	0	12	20	250	220
6	5	6	4	0	8	20	250	220
7	5	6	4	0	8	20	0	220
8	3	6	4	0	8	10	0	220
9	3	4	2	0	6	20	0	150
10	5	9	6	0	12	10	0	220
11	7	9	6	0	12	20	0	220
12	7	4	2	0	6	10	0	150
13	3	4	2	0	6	5	250	150
14	5	6	4	0	8	5	250	220
15	5	6	4	0	8	20	250	220
16	3	4	2	0	6	20	250	150
17	7	9	6	0	12	20	250	220
18	3	4	2	0	6	5	250	150
19	3	6	4	0	8	10	250	220
20	5	9	6	0	12	10	250	220
21	7	4	2	0	6	10	250	150
22	7	9	6	0	12	5	250	220
23	5	6	4	0	8	5	250	220
24	0	4	2	0	6	20	250	150
25	0	4	2	0	6	20	250	150
26	7	9	6	1	12	20	0	220
27	5	6	4	1	8	20	0	220
28	7	9	6	0	12	5	250	220
29	7	9	6	0	12	5	250	220
30	5	6	4	0	8	5	250	220
31	3	4	2	0	6	5	250	150
32	5	9	6	0	12	5	250	220
33	5	6	4	0	8	5	250	220
34	3	6	4	0	8	5	250	220
35	3	4	2	0	6	5	250	150
36	0	9	6	0	12	20	250	220
37	0	9	6	0	12	20	250	220
38	0	6	4	0	8	20	250	220
39	0	6	4	0	8	20	250	220
40	7	4	2	0	6	5	250	150
41	7	4	2	0	6	5	250	150
42	3	4	2	1	6	20	0	150
43	0	4	2	0	6	20	0	150
44	0	4	2	0	6	20	0	150
45	7	9	6	1	12	20	0	220
46	5	6	4	1	8	20	0	220
47	0	9	6	0	12	20	0	220
48	0	9	6	0	12	20	0	220
49	0	6	4	0	8	20	0	220

Results

All 49 instances of the BWSSP were optimized using Microsoft® Excel 2007® with the Premium Solver® Version 9 add-in on a Microsoft® Windows XP® workstation featuring an AMD® Athlon™ 64 X2 Dual Core Processor 4800+ running at 2.49 GHz with 1.87 GB of RAM. Each instance was first optimized using objective function (16) as the sole objective function subject to constraints (17) through (25). The optimal objective function value for objective functions (2), (4) and (5) were then recorded using the solution found by optimizing objective function (16).

BWSSP was then solved as a preemptive BILP. The Premium Solver® was reset and objective function (2) was optimized subject to constraints (17) through (25). The objective function (2) optimal value was then recorded. The Premium Solver® was again reset and objective function (2) was added as an inequality constraint greater than or equal to its optimal objective function value. Objective function (4) was then optimized using Premium Solver® subject to the new constraint as well as constraints (17) through (25). The optimal objective function value for objective function (4) was then recorded and the Premium Solver® was reset. As before, objective functions (2) and (4) were added as inequality constraints greater than or equal to their respective optimal objective function values. Finally, objective function (5) was optimized using the Premium Solver® subject to the constraints determined by the two previous objective function optimal solutions as well as constraints (17) through (25). Its optimal objective function value was then recorded.

As depicted in Table 13, the combined objective function (16) returned an optimal objective function value equivalent to those returned by objective functions (2), (4) and

(5) when each was solved preemptively for every problem instance. This data indicates that the Improved Golden Section Method (Den Boef and Den Hertog 2007) as well as the Ignizio and Thomas (1984) modification to the Sherali (1982) algorithm were both implemented correctly in the VBA® code and that the methodology appears to be a valid approach to finding an optimal solution to the BWSSP.

Table 13. Objective Function Values for BWSSP Test Instances

Test Instance	Combined BILP				Pre-emptive BILP		
	Obj Func (16)	Obj Func (2)	Obj Func (4)	Obj Func (5)	Obj Func (2)	Obj Func (4)	Obj Func (5)
1	1.24466E+17	1135938	976216	11667	1135938	976216	11667
2	6.9121E+15	302404	148675	9122	302404	148675	9122
3	1.73322E+16	300978	434043	15006	300978	434043	15006
4	1.01693E+18	2003987	963044	11667	2003987	963044	11667
5	4.54854E+16	587649	147265	9122	587649	147265	9122
6	1.40106E+17	523103	428590	15006	523103	428590	15006
7	1.69262E+16	505197	745642	4637	505197	745642	4637
8	1.31563E+16	621496	913858	1925	621496	913858	1925
9	1.24816E+17	1725792	1482173	3517	1725792	1482173	3517
10	6.60088E+15	610410	307510	853	610410	307510	853
11	2.0125E+16	787288	389226	4781	787288	389226	4781
12	1.72423E+17	3084632	2590929	680	3084632	2590929	680
13	6.90981E+16	273849	92803	3252	273849	92803	3252
14	7.96818E+16	275007	187740	1594	275007	187740	1594
15	1.79809E+17	854510	741659	4637	854510	741659	4637
16	1.70633E+18	3060976	1471209	3517	3060976	1471209	3517
17	2.24858E+17	1384900	387222	4781	1384900	387222	4781
18	1.7053E+17	413836	143175	3690	413836	143175	3690
19	1.49558E+17	1130166	958903	2070	1130166	958903	2070
20	4.06074E+16	824574	229832	977	824574	229832	977
21	2.05119E+18	5025725	2768313	1400	5025725	2768313	1400
22	5.83979E+16	263439	60262	3105	263439	60262	3105
23	1.96138E+17	323560	169623	5253	323560	169623	5253
24	3.90365E+12	1232216	482949	0	1232216	482949	0
25	3.90365E+12	1232216	482949	0	1232216	482949	0
26	1.28466E+14	7931	2931	11564	7931	2931	11564
27	2.27214E+15	24112	32932	9673	24112	32932	9673
28	8.04154E+15	177963	40775	985	177963	40775	985
29	2.34549E+16	337798	71498	1378	337798	71498	1378
30	4.13857E+17	565224	478585	1426	565224	478585	1426
31	7.13604E+17	932936	352483	1157	932936	352483	1157
32	4.61048E+15	260924	71778	426	260924	71778	426
33	1.97196E+17	416144	257086	1349	416144	257086	1349
34	3.37978E+16	367718	290415	328	367718	290415	328
35	5.07662E+17	611181	229550	1574	611181	229550	1574
36	2.50221E+11	362634	62997	0	362634	62997	0
37	2.50221E+11	362634	62997	0	362634	62997	0
38	2.25502E+12	633784	353259	0	633784	353259	0
39	2.25502E+12	633784	353259	0	633784	353259	0
40	3.33282E+17	1244326	670755	1105	1244326	670755	1105
41	1.20939E+18	1765855	997463	1212	1765855	997463	1212
42	8.41629E+15	119968	98027	4095	119968	98027	4095
43	2.79091E+13	8206934	7990527	0	8206934	7990527	0
44	1.89065E+13	6713863	6533277	0	6713863	6533277	0
45	5.30556E+14	41406	20516	4094	41406	20516	4094
46	7.1827E+13	8411	12316	2682	8411	12316	2682
47	2.07052E+13	1861242	1102338	0	1861242	1102338	0
48	1.93794E+13	1895340	1124785	0	1895340	1124785	0
49	7.52243E+13	4079719	6953075	0	4079719	6953075	0

Table 14 contains the number of decision variables and constraints for each BWSSP instance. It also contains the optimal weights found via the Improved Golden Section Method (Den Boef and Den Hertog 2007) for the Ignizio and Thomas (1984) equations used to create the coefficients for objective function (16) in each BWSSP instance. The total time (in hours, minutes and seconds) to build the constraint set, build the Ignizio and Thomas (1984) equations and optimize the BWSSP instance (as both a combined (single) objective BILP and using preemptive objective functions) are listed in Table 14.

As discussed in Chapter III, there is a trade-off when using the combined objective function versus the series of preemptive objective functions. Namely, one may lose the ability to find other, equally optimal BWSSP solutions. However, the combined objective function (16) should, on average, provide an optimal solution to the BWSSP in a more computationally efficient manner than solving objective functions (2), (4) and (5) preemptively. The times (in hours, minutes and seconds) Premium Solver® took to achieve optimality for each BWSSP instance via both methods are listed in Table 14. The times (in hours, minutes and seconds) it took Excel 2007® to create all of the constraints for each BWSSP instance are also listed in Table 14.

Table 14. Problem Sizes and Times to Optimal Solutions for BWSSP Test Instances

Test Instance	Num Decision Variables	Num Constraints	λ_1^*	λ_2^*	Time (HH:MM:SS) Constraint Build	Ignizio Thomas	Combined Solver	Pre-Emptive Solve
1	828	202	0	95249	0:00:55	0:00:04	0:00:24	0:00:53
2	834	207	0	53144	0:01:01	0:00:03	0:00:21	0:00:55
3	840	212	0	214616	0:01:04	0:00:03	0:00:20	0:00:57
4	828	208	0	86320	0:01:06	0:00:04	0:00:25	0:00:52
5	834	210	0	50824	0:01:12	0:00:03	0:00:22	0:01:03
6	840	216	0	203724	0:01:17	0:00:04	0:00:21	0:00:59
7	996	217	0	100693	0:01:25	0:00:05	0:00:28	0:01:19
8	981	203	0	52578	0:01:27	0:00:04	0:00:29	0:01:19
9	996	217	0	55957	0:01:28	0:00:04	0:00:24	0:00:54
10	984	205	0	18462	0:01:28	0:00:04	0:00:24	0:01:07
11	996	217	0	44060	0:01:29	0:00:04	0:00:30	0:01:19
12	993	211	0	33817	0:01:30	0:00:04	0:00:30	0:01:20
13	1416	365	0	232973	0:01:36	0:00:08	0:00:27	0:00:41
14	1398	350	0	391842	0:01:54	0:00:08	0:00:23	0:00:34
15	996	220	0	100230	0:02:07	0:00:04	0:00:32	0:01:26
16	996	220	0	55456	0:02:09	0:00:03	0:00:25	0:01:01
17	996	220	0	43822	0:02:13	0:00:04	0:00:35	0:01:25
18	1896	423	0	344034	0:02:15	0:00:13	0:01:18	0:03:49
19	1041	214	0	55707	0:02:18	0:00:05	0:00:38	0:01:33
20	1047	218	0	19126	0:02:19	0:00:04	0:00:35	0:01:20
21	1047	218	0	40455	0:02:25	0:00:05	0:00:36	0:01:28
22	1938	471	0	177738	0:02:47	0:00:12	0:00:48	0:02:22
23	1914	449	0	691559	0:02:56	0:00:14	0:01:15	0:04:04
24	450	454	0	0	0:04:09	0:00:02	0:00:32	0:00:11
25	450	454	0	0	0:04:09	0:00:01	0:00:31	0:00:12
26	1999	2995	13	259684	0:04:28	0:00:12	0:00:43	0:01:16
27	1999	2993	2	853789	0:04:37	0:00:12	0:00:45	0:00:59
28	1785	373	0	55314	0:04:54	0:00:09	0:01:47	0:04:54
29	1734	369	0	55170	0:04:57	0:00:09	0:01:03	0:02:10
30	1731	368	0	276030	0:05:13	0:00:09	0:01:00	0:01:54
31	1734	369	0	134538	0:05:18	0:00:09	0:01:28	0:04:07
32	1935	366	0	30594	0:05:31	0:00:10	0:01:48	0:04:42
33	1788	375	0	294715	0:05:36	0:00:10	0:01:37	0:04:44
34	1941	370	0	79577	0:05:36	0:00:11	0:01:23	0:03:12
35	1797	382	0	161815	0:05:39	0:00:10	0:01:01	0:01:55
36	450	480	0	0	0:05:46	0:00:01	0:00:44	0:00:28
37	450	480	0	0	0:05:47	0:00:01	0:00:43	0:00:31
38	450	495	0	0	0:05:57	0:00:01	0:03:09	0:08:32
39	450	495	0	0	0:05:58	0:00:02	0:03:18	0:07:58
40	1944	372	0	87428	0:06:21	0:00:11	0:01:52	0:05:01
41	1956	380	0	115445	0:06:50	0:00:11	0:01:56	0:05:47
42	1999	2997	0	288815	0:06:57	0:00:11	0:01:10	0:01:13
43	1146	1189	0	1	0:08:04	0:00:09	0:14:43	0:15:33
44	1146	1122	0	1	0:08:45	0:00:10	0:13:54	0:18:18
45	1999	2998	2	147841	0:09:33	0:00:12	0:01:33	0:01:03
46	1999	2998	4	442093	0:09:36	0:00:12	0:01:40	0:01:39
47	1083	1294	0	1	0:11:25	0:00:04	0:22:45	0:26:00
48	1044	1173	0	1	0:12:09	0:00:05	0:10:57	0:18:58
49	1044	1196	0	2	0:12:18	0:00:04	0:06:53	0:19:59

If using the combined objective function (16) is a more computationally efficient method, on average, for optimizing the BWSSP than preemptively optimizing objective functions (2), (4) and (5), the mean of the Ignizio and Thomas (1984) optimization times added to the mean of the Premium Solver® optimization times for objective function (16) should be less than the total mean optimization time for objective functions (2), (4) and (5), when each is optimized preemptively. From Table 14, these values are calculated as:

Table 15. Mean and Variance of BWSSP Optimal Solution Times
(Hours, Minutes, Seconds)

	Combined	**Preemptive**
Mean	0:02:23	0:03:57
Variance	0:18:50	0:33:43

On first glance, it appears that the combined objective function (16) does, on average, find an optimal BWSSP solution significantly faster than preemptively optimizing objective functions (2), (4) and (5). However, this sample of BWSSP instances may not be representative of the larger population of all BWSSP instances.

A paired-*t* statistical hypothesis test can be performed on the sample data in Table 14 to see if the difference in the means of the two solution methods is statistically significant (Montgomery 2005). Assuming the hypothesis that the mean of the combined objective function times to optimality is equal to the mean of the preemptive objective function times to optimality, one can test the alternative hypothesis that the mean of the combined objective function times to optimality is less than the mean of the preemptive objective functions time to optimality using the test statistic (Montgomery 2005):

$$\left|t_0\right| = \frac{\frac{1}{49}\left|\sum_{j=1}^{49}\varphi_j\right|}{S_\varphi / \sqrt{49}} = \frac{1.57381}{2.377131 / \sqrt{49}} = 4.634437965 \quad (26)$$

Where φ_j is the difference (in minutes) between the *j*th preemptive objective functions total optimization time and the *j*th combined objective function optimization time and S_φ is the standard deviation of the sample of φ_j values.

Assuming a 99.9% confidence interval, the hypothesis that the two means are equal is rejected because $\left|t_0\right| = 4.634437965 > \mathrm{t}_{0.0005,48} = 3.733882843$ (Montgomery 2005). Therefore, we can conclude with a 99.9% level of confidence that the mean time to optimality for the combined objective function formulation is less than the mean time to optimality for the preemptive objective functions formulation. In Figure 11, the complexity of each BWSSP instance is approximated by the time required to build the instance constraints. The general trend indicates that as constraint build time increases (increasing BWSSP instance complexity), the combined objective function approach is less than or equal to the time required for the preemptive objective functions approach.

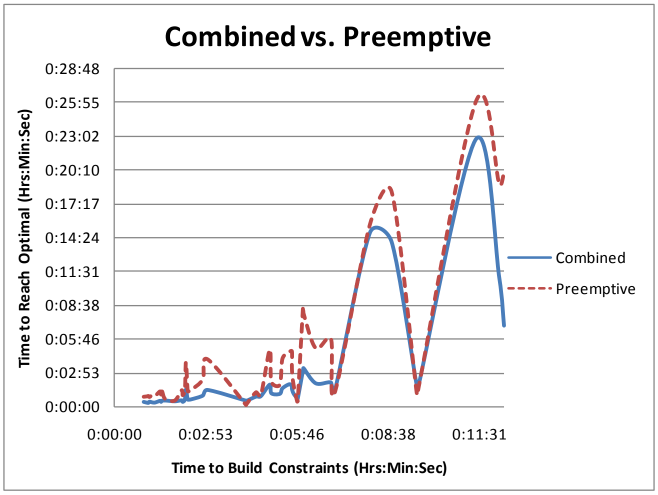


Figure 11. Combined Objective Function vs. Preemptive Objective Functions

Computational Efficiency Analytical Conclusions

Table 13 empirically demonstrates the optimal solution to the BWSSP found using the combined objective function (16) captures the preemptive relationship between objective functions (2), (4) and (5). The main drawback of the combined objective function approach, however, is that it may mask some alternative optimal solutions that would otherwise be found using the preemptive objective functions (2), (4) and (5). Therefore, if determining an optimal solution quickly is more important than finding alternative optimal solutions for the BWSSP, the combined objective function approach is preferred.

Large BWSSP Instances

For the BWSSP to be a practical methodology for creating METOC collection plans, SMOs must be able to find optimal solutions for the BWSSP in a reasonable amount of time using computer hardware and software that is readily available in a typical military headquarters. Eleven large instances of the BWSSP were solved on a

Hewlett-Packard HP Pavilion dv6500 Notebook PC with an Inter® Core™ 2 Duo CPU T7300 operating at 2.00GHz with 2.0 GB RAM running the Windows® Vista® operating system. The instances were created using the same VBA® code as before in Microsoft® Excel 2007® with the Premium Solver® Version 9 add-in. The results obtained from these instances are given in Table 16:

Table 16. Large BWSSP Instances Optimized on a Common Laptop Computer

Total Ops	μ	Model Res (km)	τ	Max $\|H_t\|$	Dec Var	Num Constr	v	Max s_{nt}	Objective Functions				λ^*_1	λ^*_2	Time (HH:MM:SS)		
									(16)	(2)	(4)	(5)			Ignizio Thomas	Combined Solver	Constraint Build
15	540	45	13	50	8154	1617	3	20	1.66766E+2	1054721	8898436	1701	0	966305	0:02:33	0:13:48	1:30:07
15	540	45	13	50	8154	1780	3	20	6.29185E+1	4684913	6917862	2422	0	1726734	0:03:00	0:24:52	0:21:47
15	540	45	13	50	7968	1646	3	20	1.65952E+1	4010450	2023006	2271	0	587940	0:01:49	0:04:48	0:15:56
15	540	45	12	55	7593	1650	4	20	1.58677E+2	1259185	1072905	1731	0	730711	0:01:40	0:20:15	0:10:43
15	540	45	12	55	7656	1867	4	20	3.99717E+1	4713436	7118371	1718	0	1295180	0:01:44	0:29:05	0:14:28
15	540	45	12	55	7362	1640	4	20	7.36242E+1	2490605	1238329	1697	0	585077	0:01:35	0:05:32	0:13:25
6	540	45	12	10	6654	737	4	5	1.57413E+1	1324465	472834	9089	0	1622548	0:01:30	0:21:50	0:08:33
6	540	45	12	10	6711	774	4	5	1.10301E+1	875942	705600	8313	0	2852157	0:01:36	0:21:55	0:11:15
6	540	45	12	10	6666	740	4	5	6.39607E+1	392491	95541	5008	1	795784	0:01:35	0:21:56	0:10:59
6	194	15	12	5	23358	2053	4	5	1.43409E+2	2192688	858852	1182	0	5971380	0:23:51	0:30:57	2:16:51
6	194	15	12	5	23397	2083	4	5	2.83084E+2	2543105	2227429	7037	0	7803005	0:25:14	0:30:25	4:23:29

Though this is by no means an exhaustive list of all possible BWSSP scenarios, the largest of the test instances had 23,397 decision variables with 2,083 constraints (decision variables with an objective function (16) coefficient equal to zero were removed in pre-processing). The first nine instances were optimized over a roughly "square" ACMES model window (23 × 23 model points) covering an area of about 900,000 square kilometers at 45 kilometers resolution. The last two instances were optimized over a "square" ACMES model window (13 × 13 model points) that is about 35,000 square kilometers at 15 kilometers resolution. Friendly installations were randomly inserted into the model window at random time periods.

An attempt was made to solve each instance preemptively using objective functions (2), (4) and (5). The Premium Solver® found the optimal solution in 1 hour and 13 minutes for the first instance and failed to return an optimal solution within an 8-hour time limit for the last 10 instances. The largest instance in the test set was optimized subject to constraints (17) – (25) using the single, combined objective function (16) in 5 hours and 19 minutes, including the time required to build constraints in Excel 2007®. This is well within the 24-hour time limit discussed in the introduction. This indicates that the BWSSP provides a useful methodology for creating a METOC collection plan for most real-world scenarios.

BWSSP Sensitivity Analysis

This section discusses a method for performing sensitivity analysis on BWSSP optimal solutions. A specific BWSSP instance is created and optimized. The optimal solution is interpreted as a METOC collection plan. The *S* matrix is modified and the BWSSP is re-optimized. The difference between the new and previous BWSSP optimal solutions is analyzed in terms of a gain or loss of significant weather condition detection in the METOC collection plan.

BWSSP Instance as a METOC Collection Plan

Consider a military campaign occurring over 16 regions $(\mu = 16)$ over a total of 2 time periods $(\tau = 2)$. Each region is at least 45 kilometers from the nearest region (simulating the ACMES model at 45 kilometer resolution). The CONOPS indicate that operation types Operation 1 and Operation 2 will be executed in that order of priority during time period 1 and operation type Operation 2 will be the only operation type executed during time period 2. There are five friendly installations located on the battlefield $H_1 = H_2 = \{2,7,13,14,15\}$. For the campaign, sensor types $n \in N$ are available according to the schedule in Table 17:

Table 17. Sensor Availability (*S* Matrix) for BWSSP Instance

N	**Sensor Type**	**δ_n (km)**	***T***	
			t = 1	**t = 2**
n = 1	**Clandestine SFC Sensor**	0	1	1
n = 2	**Base SFC Sensor**	0	1	1
n = 3	**Clandestine Upper-Air Sensor**	0	1	2
n = 4	**Upper-Air Sensor**	50	0	2
n = 5	**Weather Radar**	65	2	1

The simulated ACMES-derived hourly percent occurrence values of operationally significant weather conditions that can be detected by the sensor types $n \in N$ are given in Tables 18 and 19:

Table 18. "Red" Hourly Percent Occurrence Parameters for BWSSP Instance (r_{onmt})

Sensor	**n = 1**			**n = 2**			**n = 3**			**n = 4**			**n = 5**		
Time Period	**t = 1**		**t = 2**	**t = 1**		**t = 2**	**t = 1**		**t = 2**	**t = 1**		**t = 2**	**t = 1**		**t = 2**
Region	**o = 1**	**o = 2**	**o = 1**	**o = 1**	**o = 2**	**o = 1**	**o = 1**	**o = 2**	**o = 1**	**o = 1**	**o = 2**	**o = 1**	**o = 1**	**o = 2**	**o = 1**
1	40	41	30	44	46	33	40	41	30	44	46	33	44	46	33
2	23	30	27	26	33	30	23	30	27	26	33	30	26	33	30
3	36	32	40	40	35	44	36	32	40	40	35	44	40	35	44
4	33	45	42	37	50	47	33	45	42	37	50	47	37	50	47
5	29	38	25	32	42	28	29	38	25	32	42	28	32	42	28
6	37	33	43	41	37	48	37	33	43	41	37	48	41	37	48
7	37	41	37	41	46	41	37	41	37	41	46	41	41	46	41
8	29	30	42	32	33	47	29	30	42	32	33	47	32	33	47
9	29	26	40	32	29	44	29	26	40	32	29	44	32	29	44
10	41	43	30	46	48	33	41	43	30	46	48	33	46	48	33
11	41	34	34	46	38	38	41	34	34	46	38	38	46	38	38
12	36	40	41	40	45	46	36	40	41	40	45	46	40	45	46
13	45	36	26	50	40	29	45	36	26	50	40	29	50	40	29
14	43	31	40	48	34	45	43	31	40	48	34	45	48	34	45
15	28	44	32	31	49	36	28	44	32	31	49	36	31	49	36
16	38	31	35	42	34	39	38	31	35	42	34	39	42	34	39

Table 19. "Amber" Hourly Percent Occurrence Parameters for BWSSP Instance (y_{onmt})

Sensor	n = 1			n = 2			n = 3			n = 4			n = 5		
Time Period	t = 1		t = 2	t = 1		t = 2	t = 1		t = 2	t = 1		t = 2	t = 1		t = 2
Region	o = 1	o = 2	o = 1	o = 1	o = 2	o = 1	o = 1	o = 2	o = 1	o = 1	o = 2	o = 1	o = 1	o = 2	o = 1
1	41	1	53	6	12	19	46	10	44	25	34	36	50	18	51
2	68	23	13	15	21	65	58	2	17	22	15	58	42	44	33
3	24	48	56	42	3	40	20	12	50	58	55	4	25	53	8
4	20	16	19	32	22	50	55	12	28	1	19	37	52	32	19
5	21	20	5	12	19	35	51	30	8	44	31	49	49	14	70
6	9	10	17	58	39	17	56	44	41	36	0	13	53	26	33
7	33	19	41	28	20	47	14	43	52	31	8	28	9	20	29
8	16	33	9	0	63	3	25	33	21	2	55	41	53	50	4
9	42	68	53	29	52	5	39	24	10	49	32	47	45	62	48
10	21	2	11	16	3	47	21	42	49	17	41	60	49	51	3
11	52	50	29	41	21	14	12	66	42	53	29	28	45	33	25
12	31	4	55	53	21	20	37	28	53	17	4	14	47	53	11
13	10	40	35	41	34	55	35	12	60	18	14	24	39	27	8
14	39	47	59	9	16	12	8	54	14	17	2	42	25	50	33
15	54	44	29	8	48	22	4	10	61	24	14	55	51	48	62
16	38	28	50	1	28	17	39	11	54	14	25	11	33	54	3

Substituting these parameters into the BWSSP and maximizing objective function (16) yields the optimal solution $x_{1,14,1} = 1$, $x_{2,7,1} = 1$, $x_{3,14,1} = 1$, $x_{5,7,1} = 1$, $x_{5,14,1} = 1$, $x_{1,1,2} = 1$, $x_{2,2,2} = 1$, $x_{3,1,2} = 1$, $x_{4,2,2} = 1$, $x_{4,7,2} = 1$ with all other decision variables equal to 0. In plain language, this solution becomes the METOC collection plan in Table 20.

Table 20. METOC Collection Plan from BWSSP Instance Optimal Solution

Deploy One Sensor of Type	To Region	At Start of Time Period
Base Surface Sensor	7	1
Weather Radar	7	1
Clandestine Surface Sensor	14	1
Clandestine Upper-Air Sensor	14	1
Weather Radar	14	1
Clandestine Surface Sensor	1	2
Clandestine Upper-Air Sensor	1	2
Base Surface Sensor	2	2
Upper-Air Sensor	2	2
Upper-Air Sensor	7	2

Pictorially, this METOC collection plan looks like Figure 12:

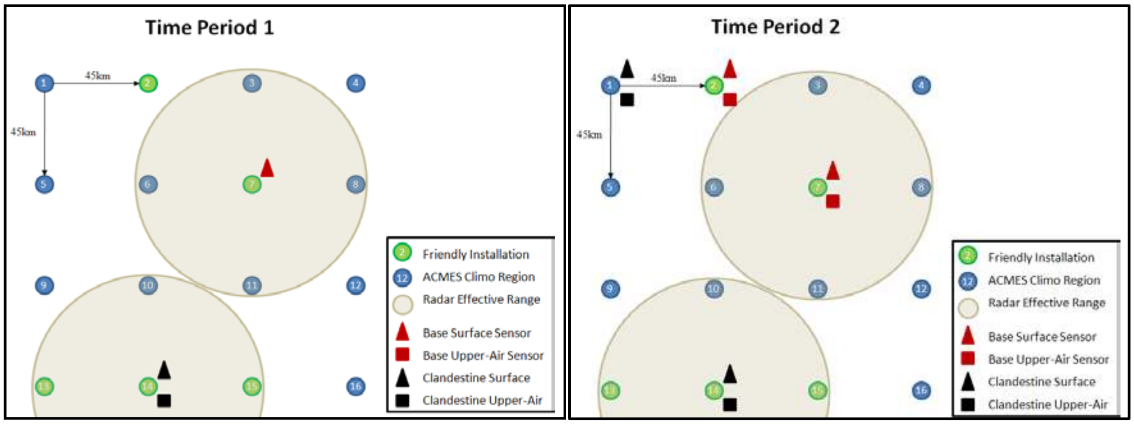


Figure 12. Pictorial Representation of METOC Collection Plan in Table 20

As outlined in Chapter III, the value of this METOC collection plan can be explained to the overall commander in terms of the military campaign plan using the r_{onmt} and y_{onmt} parameters for all $o \in O_t$, $n \in N$, $m \in M$ and $t \in T$ by following the format in Tables 5 and 6. Table 21 depicts the value of the METOC collection plan in Table 20 and Figure 12 in terms of the military campaign plan using the method described in Tables 5 and 6.

Table 21. METOC Collection Plan Value in Terms of Military Campaign

<table>
<tr><td colspan="3">Expected Hours METOC Collection Plan Will Find “Red” Conditions For Operations Planned During Time Period 1</td><td colspan="5">Expected Hours METOC Collection Plan Will Find “Red” Conditions For Operations Planned During Time Period 2</td></tr>
<tr><td rowspan="2">(744 Hours in Period 1)</td><td colspan="2">Regions</td><td rowspan="2">(744 Hours in Period 2)</td><td colspan="4">Regions</td></tr>
<tr><td>7</td><td>14</td><td>1</td><td>2</td><td>7</td><td>14</td></tr>
<tr><td>Operation 1</td><td>305.04</td><td>357.12</td><td>Operation 2</td><td>223.2</td><td>223.2</td><td>305.04</td><td>334.8</td></tr>
<tr><td>Operation 2</td><td>342.24</td><td>252.96</td><td colspan="5"></td></tr>
<tr><td colspan="3">Expected Hours METOC Collection Plan Will Find “Amber” Conditions For Operations Planned During Time Period 1</td><td colspan="5">Expected Hours METOC Collection Plan Will Find “Amber” Conditions For Operations Planned During Time Period 2</td></tr>
<tr><td rowspan="2">(744 Hours in Period 1)</td><td colspan="2">Regions</td><td rowspan="2">(744 Hours in Period 2)</td><td colspan="4">Regions</td></tr>
<tr><td>7</td><td>14</td><td>1</td><td>2</td><td>7</td><td>14</td></tr>
<tr><td>Operation 1</td><td>208.32</td><td>290.16</td><td>Operation 2</td><td>394.32</td><td>483.6</td><td>349.68</td><td>438.96</td></tr>
<tr><td>Operation 2</td><td>148.8</td><td>401.76</td><td colspan="5"></td></tr>
</table>

Sensitivity Analysis of a BWSSP Solution as a METOC Collection Plan

As mentioned in Chapter III, the methodology in the previous section can be used to perform sensitivity analysis on the BWSSP to determine the value of a particular sensor in the METOC collection plan (BWSSP optimal solution). Returning to the example in the previous section, if the clandestine surface weather sensor scheduled to become available at the start of time period 1 is delayed until the start of time period 2, the optimal solution to the BWSSP becomes $x_{2,7,1} = 1$, $x_{5,7,1} = 1$, $x_{1,1,2} = 1$, $x_{2,2,2} = 1$, $x_{1,14,2} = 1$, $x_{3,1,2} = 1$, $x_{3,14,2} = 1$, $x_{4,2,2} = 1$, $x_{4,7,2} = 1$, $x_{5,14,2} = 1$, and all other decision variables equal to 0. In plain language, this becomes the METOC collection plan in Table 22.

Table 22. METOC collection Plan from Modified BWSSP Instance Optimal Solution

Deploy One Sensor of Type	To Region	At Start of Time Period
Base Surface Sensor	7	1
Weather Radar	7	1
Clandestine Surface Sensor	1	2
Clandestine Upper-Air Sensor	1	2
Base Surface Sensor	2	2
Upper-Air Sensor	2	2
Upper-Air Sensor	7	2
Clandestine Surface Sensor	14	2
Clandestine Upper-Air Sensor	14	2
Weather Radar	14	2

Pictorially, this METOC collection plan looks like Figure 13:

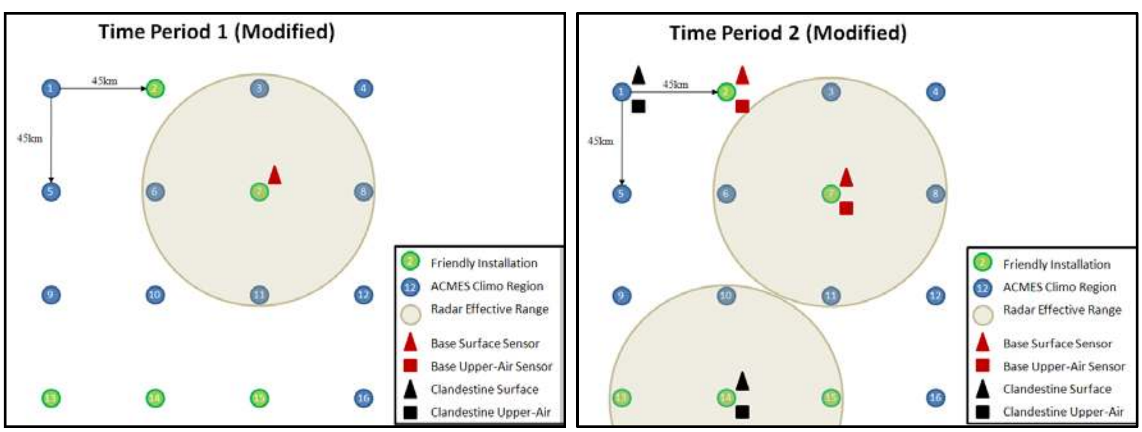


Figure 13. Pictorial Representation of METOC Collection Plan in Table 22

A comparison of Figures 12 and 13 reveals that the difference between the METOC collection plan in Table 20 and the METOC collection plan in Table 22 is represented by the expected hours of detection of significant weather conditions provided by the weather sensors allocated to region 14 during period 1 in the original METOC collection plan. An examination of Table 20 provides the loss of capability in the METOC collection plan due to the change in availability of the clandestine surface weather sensor (Table 23).

Table 23. Degradation of Capability in METOC Collection Plan (Sensitivity Analysis)

Expected Hours of Detection Lost Due to Clandestine Surface Weather Sensor Arrival Change (Time Period 1)			
"Red" Conditions		"Amber" Conditions	
(744 Hours in Period 1)	Region 14	(744 Hours in Period 1)	Region 14
Operation 1	357.12	Operation 1	290.16
Operation 2	252.96	Operation 2	401.76

Analytical Conclusions

In this chapter, several BWSSP instances were generated using VBA® and solved to optimality using Excel 2007® with Premium Solver® Version 9. Each BWSSP instance was configured with parameter settings that depict realistic military campaign scenarios. The combined objective function (16) maintained the preemptive nature of the original preemptive objective functions (2), (4) and (5) in all problem instances while, on average, solving the BWSSP in less time. Several large instances of the BWSSP were optimized on a standard laptop computer, demonstrating that the BWSSP creates a realistic METOC collection plan within a reasonable amount of time. Sensitivity analysis was conducted on a single, smaller BWSSP instance illustrating how a BWSSP solution can be interpreted as a METOC collection plan and how the merits of one METOC collection plan can be compared with those of another. In summary, the BWSSP provides an efficient mathematical model that creates optimal METOC collection plans for any type of military campaign anywhere on the globe.

V. Conclusions and Future Research

Introduction

This chapter concludes the research into the BWSSP. The research conclusions and contributions are discussed. Future BWSSP research areas are suggested.

Research Conclusions and Contributions

Though there are a wide range of sensor allocation schemes in the current literature, this research is the first approach that considers historical behaviors of the detection targets as a series of preemptive objectives to form a measure of the usefulness of a particular sensor placement. In the case of the BWSSP, gridded climatology models and historical weather observations provide reasonable estimates of when and where operationally significant weather conditions are expected to occur. Since the objective of the BWSSP is to maximize detection of these conditions, these estimates of past behaviors provide a reasonable estimate of target behaviors for the weather sensors allocated in the METOC collection plan. The methodology applied to the BWSSP can be expanded to other resource allocation problems with multiple, preemptive objectives where reasonable estimates of target behaviors are known and where time, space and the lifespan of the allocated resource are factors for consideration.

Additionally, this research creates the first mathematical model for a military METOC collection plan. The BWSSP model, as outlined in this research, can be used to compare METOC collection plans for differing supplies of weather sensors, thereby providing the SMO and overall commander with the ability to create a cost-benefit analysis for weather sensor procurement decisions. This type of analysis was not possible before this research.

Future Research

The BWSSP as described in this research only handles static weather sensors. One potential avenue of future research involves incorporating mobile weather sensors into the BWSSP as a means of filling in the gaps of sparse METOC collection plans. The BWSSP may also be applied to any type of sensing network where target detection abilities change over time. The methodology for creating the preemptive objectives of the BWSSP may also be applicable to other sensing networks where sensors are capable of performing multiple tasks, each with their own priority that changes over time.

Though the BWSSP can be optimized in a reasonable amount of time as a BILP, other approaches may bear fruit. A Dynamic Programming approach may be a more computationally efficient way to tackle the BWSSP. Also, in cases where there are a large number of friendly installations within a very small area, constraints 22 – 25 make the BWSSP much more difficult to optimize. A heuristic approach to this scenario may provide a sufficiently "good" basic feasible solution to the BWSSP in far less time than the BILP approach in this research.

Appendix A. Visual Basic for Applications Documentation

Code Description

The BWSSP instances in Chapter IV were created using the Microsoft® Visual Basic for Applications (VBA®) programming language. This code was implemented as a series of macros in Microsoft Excel 2007®. Each BWSSP instance BILP was solved to optimality using the Premium Solver® Version 9 Excel® add-in manufactured by Frontline Systems, Inc.

Source Code

VBA® is an object-oriented programming language embedded in the Microsoft Office 2007® suite. In VBA®, a statement preceded by a ‘ indicates a comment. The subroutines below create instances of the BWSSP based on user input, pass the BWSSP instance to Premium Solver® Version 9 and then store the optimal solution. Also included are VBA® subroutines that implement Ignizio and Thomas (1984) as well as the Improved Golden Section method developed by Den Boef and Den Hertog (2007).

BWSSP Test Subroutine

```
Option Base 1
Option Explicit

Sub Test()
' This subroutine runs several test instances of the BWSSP

Application.Calculation = xlCalculationManual       ' Turn off automatic calculation

Dim LOOP_COUNTER As Long        ' Loop counter
Dim StartTime As Date           ' Start time of the current solver run
Dim ProbStart As Date           ' Start time of the current problem instance
Dim M As Long                   ' Number of regions
Dim N As Long                   ' Number of sensor types
Dim T As Long                   ' Number of total periods
Dim k As Long                   ' Loop counter
Dim j As Long                   ' Loop counter

LOOP_COUNTER = 4                ' Initialize loop counter
```

```
' Clear old problem data
Do While Trim(Sheets("Campaign").Range("AF" & CStr(LOOP_COUNTER))) <> ""
  LOOP_COUNTER = LOOP_COUNTER + 1
Loop

Do While Trim(Sheets("Campaign").Range("A" & CStr(LOOP_COUNTER))) <> ""
' Loop through all problem instances

  Sheets("Campaign").Range("R1:AF1").Rows(LOOP_COUNTER).ClearContents
  Application.ScreenUpdating = False              ' Turn of screen updating (code runs faster)

  ' Record total number of locations
  M = Sheets("Campaign").Range("F" & CStr(LOOP_COUNTER))
  ' Record total number of sensor types
  N = Application.WorksheetFunction.CountIf( _
    Sheets("Campaign").Range("J" & CStr(LOOP_COUNTER) & ":N" & CStr(LOOP_COUNTER)), ">0")
  ' Record total number of timeperiods
  T = Sheets("Campaign").Range("H" & CStr(LOOP_COUNTER))

  Call DeleteSheets                           ' Delete old problem instance sheets

  Application.Calculation = xlCalculationManual     ' Turn off automatic calculation
  ProbStart = Now()                           ' Start the clock on the problem instance

  ' Create a matrix of distances between locations
  Call Distance_Matrix(M, Sheets("Campaign").Range("G" & CStr(LOOP_COUNTER)), True)

  ' Build randomized climate data based ont he scenario and format the data for the BILP
  Call Build_Random_Instance(Sheets("Campaign").Range("C" & CStr(LOOP_COUNTER)), _
    Sheets("Campaign").Range("B" & CStr(LOOP_COUNTER)), Sheets("Campaign").Range("E" & CStr(LOOP_COUNTER)), M, _
    T, Sheets("Campaign").Range("I" & CStr(LOOP_COUNTER)), _
    Sheets("Campaign").Range("J" & CStr(LOOP_COUNTER)), Sheets("Campaign").Range("K" & CStr(LOOP_COUNTER)), _
    Sheets("Campaign").Range("L" & CStr(LOOP_COUNTER)), Sheets("Campaign").Range("M" & CStr(LOOP_COUNTER)), _
    Sheets("Campaign").Range("N" & CStr(LOOP_COUNTER)), Sheets("Campaign").Range("O" & CStr(LOOP_COUNTER)), _
    Sheets("Campaign").Range("P" & CStr(LOOP_COUNTER)), Sheets("Campaign").Range("Q" & CStr(LOOP_COUNTER)))

  ' Create binary matrix of deployability values
  Call A_Matrix(N, T)

  ' Create matrix of deployability * distance to nearest base values (thrid obj function coefficients)
  Call H_Matrix(N, T)

  ' Start the clock for the combined objective function solve time
  StartTime = Now()
  ' Create the combined objective function for the bottom two obj functions
  Call IgnizioThomas("b_nm", "h_nm", "v_nm", N, T)
  ' Record the weight
  Sheets("Campaign").Range("AE" & CStr(LOOP_COUNTER)) = Sheets("b_nm1").Range("B2").Cells(1, (N + 1) * 2)

  ' Take the new combined objective function and merge it with the highest priority obj function
  Call IgnizioThomas("c_nm", "b_nm", "u_nm", N, T)

  Application.Calculation = xlCalculationManual

  ' Record the weight
  Sheets("Campaign").Range("AF" & CStr(LOOP_COUNTER)) = Sheets("c_nm1").Range("B2").Cells(1, (N + 1) * 2)

  ' Record the time it takes to make the combined objective functions
  Sheets("Campaign").Range("Y" & CStr(LOOP_COUNTER)) = Now() - StartTime

  ' Update the screen to prevent Excel from crashing
  Application.ScreenUpdating = True
  Sheets("Campaign").Select
  Application.ScreenUpdating = False
  Application.DisplayAlerts = False
  ' Formulate the constraints and solve the combined BILP
  Call BWSSP_BILP(N, T, LOOP_COUNTER, False, 0, "$B$3")
```

```
  ' Solve the preemptive BILP for just the u values
  Call BWSSP_BILP(N, T, LOOP_COUNTER, True, 0, "$D$3")

  ' Solve the preemptive BILP for just the v values
  Call BWSSP_BILP(N, T, LOOP_COUNTER, True, 1, "$F$3")

  ' Solve the preemptive BILP for just the h values
  If Sheets("Campaign").Range("U" & CStr(LOOP_COUNTER)) > 0 Then
    Call BWSSP_BILP(N, T, LOOP_COUNTER, True, 2, "$H$3")
  Else
    Sheets("Campaign").Range("X" & CStr(LOOP_COUNTER)) = 0
  End If

  Application.Calculation = xlCalculationManual

  ' Loop to next problem instance
  Sheets("Campaign").Select
  Sheets("Campaign").Range("AB" & CStr(LOOP_COUNTER)).Select
  Sheets("Campaign").Range("AB" & CStr(LOOP_COUNTER)) = Now() - ProbStart
  ActiveWorkbook.Save
  LOOP_COUNTER = LOOP_COUNTER + 1
  Application.ScreenUpdating = True
Loop

Application.Calculation = xlCalculationAutomatic      ' Turn off automatic calculation
Application.DisplayAlerts = True                      ' Restore alerts

End Sub
```

Delete BWSSP Instance Sheets Subroutine

```
Sub DeleteSheets()
' This algorithm deletes problem instance worksheets to make room for new problem runs

With Application
  .Calculation = xlCalculationManual
  .DisplayAlerts = False
  .ScreenUpdating = False
End With

Dim i As Long

' Delete all sheets except for the control sheet
i = 1

Do While i <= ThisWorkbook.Sheets.Count
  If Not Sheets(i).Name = "Campaign" Then
    Sheets(i).Delete
  Else
    i = i + 1
  End If
Loop

With Application
  .Calculation = xlCalculationAutomatic
  .DisplayAlerts = True
  .ScreenUpdating = True
End With

End Sub
```

BWSSP Random Instance Subroutine

```
Sub Build_Random_Instance(Campaign As String, Climate As String, TotalOps As Long, M As Long, T As Long, _
        NumBase As Long, SFCLifespan As Long, BaseSFCLifespan As Long, BaseUALifespan As Long, _
        UALifespan As Long, RadarLifespan As Long, MaxS As Long, UARange As Long, RadarRange As Long)
```

```
' This subroutine creates a random instance of the BWSSP
' It creates the list of bases and values for delta, s, u, and v

With Application
   .DisplayAlerts = False
   .ScreenUpdating = False
   .Calculation = xlCalculationManual
End With

Dim Array1() As Variant     ' Temporary array
Dim Array2() As Variant     ' Temporary array
Dim N As Long               ' Number of sensor types
Dim S As Range              ' Range of sensor supply numbers
Dim u As Range              ' Range of u objective function coefficients Equation (2)
Dim v As Range              ' Range of v objective function coefficients Equation (4)
Dim A As Range              ' Range of deployability parameters
Dim r As Range              ' r_onmt values
Dim y As Range              ' y_onmt values
Dim D As Range              ' Distance matrix
Dim Bases As Range          ' List of friendly installations in this isntance
Dim Delta As Range          ' Sensor effective ranges/minimum spacing (in km)
Dim i As Long               ' Loop counter
Dim j As Long               ' Loop counter
Dim k As Long               ' Loop counter
Dim p As Long               ' Loop counter
Dim q As Long               ' Loop counter
Dim o As Long               ' Loop counter
Dim g As Long               ' Loop counter
Dim Lifespan() As Long      ' Array of sensor life spans (in time periods)


' Count the number of sensor types
N = 0
If SFCLifespan > 0 Then N = N + 1
If BaseSFCLifespan > 0 Then N = N + 1
If BaseUALifespan > 0 Then N = N + 1
If UALifespan > 0 Then N = N + 1
If RadarLifespan > 0 Then N = N + 1

Set D = Sheets("d").Range("B2:" & Range("B2").Cells(M, M).Address)

' Remove old problem sheets, if there are any
i = 1
Do While i <= ThisWorkbook.Sheets.Count
   If Not Sheets(i).Name = "Campaign" And Not Sheets(i).Name = "d" Then
      Sheets(i).Delete
   Else
      i = i + 1
   End If
Loop

' Create random supply of sensors
Sheets.Add.Name = "s"
Set S = Sheets("s").Range("C2:" & Range("C2").Cells(N, T).Address)
S(1, 1).Offset(-1, -2) = "Sensor Type"
' Create sensor detection ranges
Sheets.Add.Name = "delta"
Set Delta = Sheets("delta").Range("A2:" & Range("B2").Cells(N, 1).Address)

For i = 1 To N Step 1
   S(i, 1).Offset(0, -1) = "n = " & CStr(i)
   For j = 1 To T Step 1
      If i = 1 Then S(i, j).Offset(-1, 0) = "t = " & CStr(j)
      S(i, j) = Round(MaxS * Rnd(), 0)
   Next j
Next i

k = N
```

```
If RadarLifespan > 0 Then
' Code the type of this sensor as a radar
   S(k, 1).Offset(0, -2) = 1
' Add a detection range
   Delta(k, 1) = "n = " & CStr(k)
   Delta(k, 2) = RadarRange
   k = k - 1
End If

If k < 1 Then k = 1

If BaseUALifespan > 0 Then
' Code the sensor type as base-based balloon
   S(k, 1).Offset(0, -2) = 4
' Add a detection range
   Delta(k, 1) = "n = " & CStr(k)
   Delta(k, 2) = UARange
   k = k - 1
End If

If k < 1 Then k = 1

If UALifespan > 0 Then
' Code the sensor type as non-base balloon
   S(k, 1).Offset(0, -2) = 6
' Add a detection range
   Delta(k, 1) = "n = " & CStr(k)
   Delta(k, 2) = 0
   k = k - 1
End If

If BaseSFCLifespan > 0 And k >= 1 Then
' Code the sensor type as base surface sensor
   S(k, 1).Offset(0, -2) = 3
   ' Add a detection range
   Delta(k, 1) = "n = " & CStr(k)
   Delta(k, 2) = 0
   k = k - 1
End If

If SFCLifespan > 0 And k >= 1 Then

For i = k To 1 Step -1
' Code the sensor type as non-base surface sensor
   S(i, 1).Offset(0, -2) = 7
' Add a detection range
   Delta(i, 1) = "n = " & CStr(i)
   Delta(i, 2) = 0
Next i

End If

' Randomly insert bases
Sheets.Add.Name = "Bases"
Set Bases = Sheets("Bases").Range("A1:" & Range("A1").Cells(NumBase + 1, N + 5).Address)
Bases(1, 1) = "Lat"
Bases(1, 2) = "Lon"
Bases(1, 3) = "m"
Bases(1, 4) = "Name"
Bases(1, 5) = "Start Period"
Range(Bases(1, 6), Bases(1, 5 + N)).Value = Application.Transpose(S.Columns(1).Offset(0, -1))
For k = 2 To NumBase + 1 Step 1
   Bases(k, 3) = Round(Rnd() * (M - 1) + 1, 0)
   Bases(k, 4) = "Base " & CStr(k - 1)
   If Campaign = "Humanitarian Assistance" Then
      Bases(k, 5) = 1
   Else
      Bases(k, 5) = Round(Rnd() * (T - 1) + 1, 0)
```

```
    End If
    Do While Application.WorksheetFunction.CountIf(Bases.Columns(3), "=" & CStr(Bases(k, 3).Value)) > 1
      Bases(k, 3) = Round(Rnd() * (M - 1) + 1, 0)
    Loop
    Range(Bases(2, 6), Bases(NumBase + 1, 5 + N)).Value = 0
Next k


Sheets.Add.Name = "Rules"
p = 0
' Create operations lists and respective weights
For k = 1 To T Step 1
    Sheets("Rules").Range("A1").Cells(1, 2 * k - 1) = "Phase " & CStr(k)
    Sheets("Rules").Range("A2").Cells(1, 2 * k - 1) = "Weight"
    Sheets("Rules").Range("A2").Cells(1, 2 * k) = "Operation"
    j = Round(Rnd() * (TotalOps - 1), 0) + 1
    For i = 1 To j Step 1
      Sheets("Rules").Range("A3").Cells(i, 2 * k) = Round(Rnd() * (TotalOps - 1), 0) + 1
      Do While Application.WorksheetFunction.CountIf(Sheets("Rules").Columns("A").Offset(0, 2 * k - 1), "=" & _
        CStr(Sheets("Rules").Range("A3").Cells(i, 2 * k))) > 1
        Sheets("Rules").Range("A3").Cells(i, 2 * k) = Round(Rnd() * (TotalOps - 1), 0) + 1
      Loop
    Next i
    If Application.WorksheetFunction.Count(Sheets("Rules").Columns("A").Offset(0, 2 * k - 1)) > p Then _
       p = Application.WorksheetFunction.Count(Sheets("Rules").Columns("A").Offset(0, 2 * k - 1))
Next k

' Add the weights to the operations lists
For k = 1 To T Step 1
    i = 1
    q = p
    Do While Trim(Sheets("Rules").Range("A3").Cells(i, 2 * k)) <> ""
      Sheets("Rules").Range("A3").Cells(i, 2 * k - 1) = q
      q = q - 1
      i = i + 1
    Loop
Next k

' Create randomized climatology statistics
For i = 1 To TotalOps Step 1
    Sheets.Add.Name = "r_" & CStr(i) & "2mt"
    Sheets("r_" & CStr(i) & "2mt").Range("A1") = "Region"
    Sheets("r_" & CStr(i) & "2mt").Range("B1") = "Lat"
    Sheets("r_" & CStr(i) & "2mt").Range("C1") = "Lon"
    D.Columns(1).Offset(0, -1).Copy Sheets("r_" & CStr(i) & "2mt").Range("A2:A" & CStr(M + 1))
    ReDim Array1(M, 12)
    For k = 1 To 12 Step 1
      Sheets("r_" & CStr(i) & "2mt").Range("D1").Cells(1, k) = k
      For j = 1 To M Step 1
        If Climate = "Harsh" Then
          Array1(j, k) = Round(Rnd() * 25, 0) + 25
        ElseIf Climate = "Moderate" Then
          Array1(j, k) = Round(Rnd() * 25, 0)
        Else
          Array1(j, k) = Round(Rnd() * 15, 0)
        End If
      Next j
    Next k
    Sheets("r_" & CStr(i) & "2mt").Range("D2:O" & CStr(M + 1)) = Array1
    Erase Array1
    Sheets("r_" & CStr(i) & "2mt").Copy after:=Sheets(ThisWorkbook.Sheets.Count)
    ActiveSheet.Name = "r_" & CStr(i) & "4mt"
    Sheets("r_" & CStr(i) & "2mt").Copy after:=Sheets(ThisWorkbook.Sheets.Count)
    ActiveSheet.Name = "r_" & CStr(i) & "5mt"
    Sheets("r_" & CStr(i) & "2mt").Copy after:=Sheets(ThisWorkbook.Sheets.Count)
    ActiveSheet.Name = "r_" & CStr(i) & "1mt"
    ReDim Array1(M, 12)
    For k = 1 To 12 Step 1
```

```
    For j = 1 To M Step 1
      Array1(j, k) = Round(Sheets("r_" & CStr(i) & "1mt").Range("D2").Cells(j, k) * 0.9, 0)
    Next j
  Next k
  Sheets("r_" & CStr(i) & "1mt").Range("D2:O" & CStr(M + 1)) = Array1
  Erase Array1
  Sheets("r_" & CStr(i) & "1mt").Copy after:=Sheets(ThisWorkbook.Sheets.Count)
  ActiveSheet.Name = "r_" & CStr(i) & "3mt"
  For p = 1 To N Step 1
    Sheets("r_" & CStr(i) & CStr(p) & "mt").Copy after:=Sheets(ThisWorkbook.Sheets.Count)
    ActiveSheet.Name = "y_" & CStr(i) & CStr(p) & "mt"
    ReDim Array1(M, 12)
    For k = 1 To 12 Step 1
      For j = 1 To M Step 1
        If Climate = "Harsh" Then
          Array1(j, k) = _
            Round(Rnd() * (100 - Sheets("r_" & CStr(i) & CStr(p) & "mt").Range("D2").Cells(j, k)), 0)
        ElseIf Climate = "Moderate" Then
          Array1(j, k) = _
            Round(Rnd() * (50 - Sheets("r_" & CStr(i) & CStr(p) & "mt").Range("D2").Cells(j, k)), 0)
        Else
          Array1(j, k) = _
            Round(Rnd() * (15 - Sheets("r_" & CStr(i) & CStr(p) & "mt").Range("D2").Cells(j, k)), 0)
        End If
      Next j
    Next k
    Sheets("y_" & CStr(i) & CStr(p) & "mt").Range("D2:O" & CStr(M + 1)).Value = Array1
    Erase Array1
  Next p
Next i

' Build the sensor life span array
i = 1
ReDim Lifespan(N)
If SFCLifespan > 0 Then
  Lifespan(i) = SFCLifespan
  i = i + 1
End If
If BaseSFCLifespan > 0 Then
  Lifespan(i) = BaseSFCLifespan
  i = i + 1
End If
If BaseUALifespan > 0 Then
  Lifespan(i) = BaseUALifespan
  i = i + 1
End If
If UALifespan > 0 Then
  Lifespan(i) = UALifespan
  i = i + 1
End If
If RadarLifespan > 0 Then
  Lifespan(i) = RadarLifespan
  i = i + 1
End If

' Create random u values from the climatology "statistics"
For k = 1 To T Step 1
  Sheets.Add.Name = "u_nm" & CStr(k)
  For j = 1 To N Step 1
    Set u = Sheets("u_nm" & CStr(k)).Range("B2:" & Range("B2").Cells(M, 2).Address).Offset(0, 2 * (j - 1))
    u(1, 2).Offset(-1, 0) = "n = " & CStr(j)
    u(1, 1).Offset(-1, 0) = "m"
    u.Columns(2) = 0
    g = k + Lifespan(j)
    If g > T Then g = T
    q = k
    Do While q > 12
      q = q - 12
```

```
    Loop
    For p = k To g Step 1
    ' Loop through the lifespan of this sensor
      o = 1
      ' Loop through operations in this phase
      Do While Trim(Sheets("Rules").Range("A3").Cells(o, 2 * p)) <> ""
        Set r = Sheets("r_" & CStr(o) & CStr(j) & "mt").Range("D2:D" & CStr(M + 1)).Offset(0, q - 1)
        Array1 = u.Columns(2).Value
        Array2 = Application.WorksheetFunction.MMult(r, Sheets("Rules").Range("A3").Cells(o, (2 * p) - 1))
        u.Columns(2).Value = MAdd(Array1, Array2)
        o = o + 1
      Loop
      q = q + 1
      If q = 13 Then q = 1
    Next p
    If Delta(j, 2) > 0 Then
      ReDim Array1(M, 1)
      For i = 1 To M Step 1
        Array1(i, 1) = Application.WorksheetFunction.SumIf(D.Columns(i), "<=" & CStr(Delta(j, 2)), u.Columns(2))
      Next i
      u.Columns(2).Value = Array1
      Erase Array1
    End If
    D.Columns(1).Offset(0, -1).Copy u.Columns(1)
  Next j
Next k

' Create random v values from the climatology "statistics"
For k = 1 To T Step 1
  Sheets.Add.Name = "v_nm" & CStr(k)
  For j = 1 To N Step 1
    Set v = Sheets("v_nm" & CStr(k)).Range("B2:" & Range("B2").Cells(M, 2).Address).Offset(0, 2 * (j - 1))
    v(1, 2).Offset(-1, 0) = "n = " & CStr(j)
    v(1, 1).Offset(-1, 0) = "m"
    v.Columns(2) = 0
    g = k + Lifespan(j)
    If g > T Then g = T
    q = k
    Do While q > 12
      q = q - 12
    Loop
    For p = k To g Step 1
    ' Loop through the lifespan of this sensor
      o = 1
      ' Loop through operations in this phase
      Do While Trim(Sheets("Rules").Range("A3").Cells(o, 2 * p)) <> ""
        Set y = Sheets("y_" & CStr(o) & CStr(j) & "mt").Range("D2:D" & CStr(M + 1)).Offset(0, q - 1)
        Array1 = v.Columns(2).Value
        Array2 = Application.WorksheetFunction.MMult(y, Sheets("Rules").Range("A3").Cells(o, (2 * p) - 1))
        v.Columns(2).Value = MAdd(Array1, Array2)
        o = o + 1
      Loop
      q = q + 1
      If q = 13 Then q = 1
    Next p
    If Delta(j, 2) > 0 And S(j, 1).Offset(0, -2) = 1 Then
      ReDim Array1(M, 1)
      For i = 1 To M Step 1
        Array1(i, 1) = Application.WorksheetFunction.SumIf(D.Columns(i), "<=" & CStr(Delta(j, 2)), v.Columns(2))
      Next i
      v.Columns(2).Value = Array1
      Erase Array1
    End If
    D.Columns(1).Offset(0, -1).Copy v.Columns(1)
  Next j
Next k

i = 1
```

```
' Remove the original climatology values (for speed in solving)
Do While i <= ThisWorkbook.Sheets.Count
  If Mid(Sheets(i).Name, 1, 2) = "r_" Or Mid(Sheets(i).Name, 1, 2) = "y_" Then
    Sheets(i).Delete
  Else
    i = i + 1
  End If
Loop

' Erase arrays
Erase Array1, Array2

Sheets("Campaign").Select
With Application
  .ScreenUpdating = True
  .DisplayAlerts = True
  .Calculation = xlCalculationAutomatic
End With

End Sub
```

Distance Matrix Subroutine

```
Sub Distance_Matrix(M As Long, ModelRes As Long, IsRandom As Boolean)
' This subroutine generates a distance matrix for the locations on the u variable pages

Application.Calculation = xlCalculationManual
Application.ScreenUpdating = False

Dim Distance As Range
Dim i As Long
Dim j As Long
Dim k As Long
Dim D() As Long
ReDim D(M, M)
Dim Loc() As Long
ReDim Loc(M)

' Remove old problem sheets, if there are any
On Error Resume Next
Sheets("d").Delete

' Create a distance matrix
Sheets.Add.Name = "d"
Set Distance = Sheets("d").Range("B2:" & Range("B2").Cells(M, M).Address)
Distance(1, 1).Offset(-1, -1) = "m, q"

Application.ScreenUpdating = True
Sheets("Campaign").Select
Application.ScreenUpdating = False

For i = 1 To M Step 1
  Loc(i) = i
  If IsRandom Then
  ' If we are creating a random distance matrix
    D(i, i) = 0
    k = 1
    For j = i + 1 To M Step 1
      D(i, j) = Round(((((j - i) Mod CLng(M ^ 0.5)) * ModelRes) ^ 2 + (CLng((j - i - 1) / CLng(M ^ 0.5)) * ModelRes) ^ 2) ^ 0.5, 0)
      D(j, i) = Round(((((j - i) Mod CLng(M ^ 0.5)) * ModelRes) ^ 2 + (CLng((j - i - 1) / CLng(M ^ 0.5)) * ModelRes) ^ 2) ^ 0.5, 0)
    Next j
  End If
  Application.ScreenUpdating = True
  Sheets("Campaign").Select
  Application.ScreenUpdating = False
Next i
```

```
Distance.Columns(1).Offset(0, -1).Value = Application.Transpose(Loc)
Distance.Rows(1).Offset(-1, 0).Value = Loc
Distance = D
Erase D
Erase Loc
Sheets("d").Columns("A").Cells.Font.Bold = True
Sheets("d").Rows("1").Cells.Font.Bold = True

Application.Calculation = xlCalculationAutomatic
Application.ScreenUpdating = True

End Sub
```

'A' Matrix Subroutine

```
Sub A_Matrix(N As Long, T As Long)
' This subroutine generates the a (deployability) matrix for the locations on the u variable pages

Application.Calculation = xlCalculationManual
Application.ScreenUpdating = False

Dim i As Long
Dim j As Long
Dim k As Long
Dim p As Long
Dim A As Range
Dim S As Range
Dim Bases As Range

' Remove old problem sheets, if there are any
i = 1
Do While i <= ThisWorkbook.Sheets.Count
  If Mid(Sheets(i).Name, 1, 2) = "a_" Then
    Sheets(i).Delete
  Else
    i = i + 1
  End If
Loop

i = Application.WorksheetFunction.Count(Sheets("Bases").Columns("C"))
Set Bases = Sheets("Bases").Range("A2:" & Range("A2").Cells(i, 5 + N).Address)

Set S = Sheets("s").Range("C2:" & Range("C2").Cells(N, T).Address)

For k = 1 To T Step 1
  Sheets("u_nm" & CStr(k)).Copy after:=Sheets(ThisWorkbook.Sheets.Count)
  ActiveSheet.Name = "a_nm" & CStr(k)
  For j = 1 To N Step 1
    Set A = Sheets("a_nm" & CStr(k)).Range("B2:C" & CStr(1 + Application.WorksheetFunction.Count( _
      Sheets("a_nm" & CStr(k)).Columns("B").Offset(0, 2 * (j - 1))))).Offset(0, 2 * (j - 1))
    A.Columns(2) = 1
    ' If this sensor type is base-only and it is not a base, we make the a value zero
    If CLng(S(j, 1).Offset(0, -2)) <= 4 Then
      For i = 1 To A.Rows.Count Step 1
        If Application.WorksheetFunction.CountIf(Bases.Columns(3), "=" & CStr(A(i, 1))) = 0 Then
          A.Rows(i).Cells.ClearContents
          Sheets("u_nm" & CStr(k)).Range(A.Rows(i).Address).Cells.ClearContents
          Sheets("v_nm" & CStr(k)).Range(A.Rows(i).Address).Cells.ClearContents
        Else
          p = 1
          ' Find the base
          Do While Bases(p, 3) <> A(i, 1) And p < Bases.Rows.Count
            p = p + 1
          Loop
          ' If the base doesn't exist yet or it is marked as undeployable for this sensor type, make a zero
```

```
          If Bases(p, 3) = A(i, 1) And _
            (Bases(p, 5) > k Or Bases(p, j + 6) < 0) Then
            A.Rows(i).Cells.ClearContents
            Sheets("u_nm" & CStr(k)).Range(A.Rows(i).Address).Cells.ClearContents
            Sheets("v_nm" & CStr(k)).Range(A.Rows(i).Address).Cells.ClearContents
          End If
        End If
      Next i
    End If
    ' Sort out deleted locations and sensors where a = 0
    With ActiveWorkbook.Worksheets("u_nm" & CStr(k)).Sort
      With .SortFields
        .Clear
        .Add Key:=Range(A.Columns(1).Address), SortOn:=xlSortOnValues, _
          Order:=xlAscending, DataOption:=xlSortNormal
      End With
      .SetRange Range(A.Address)
      .Header = xlNo
      .MatchCase = False
      .Orientation = xlTopToBottom
      .SortMethod = xlPinYin
      .Apply
      .SortFields.Clear
    End With
    With ActiveWorkbook.Worksheets("v_nm" & CStr(k)).Sort
      With .SortFields
        .Clear
        .Add Key:=Range(A.Columns(1).Address), SortOn:=xlSortOnValues, _
          Order:=xlAscending, DataOption:=xlSortNormal
      End With
      .SetRange Range(A.Address)
      .Header = xlNo
      .MatchCase = False
      .Orientation = xlTopToBottom
      .SortMethod = xlPinYin
      .Apply
      .SortFields.Clear
    End With
    With ActiveWorkbook.Worksheets("a_nm" & CStr(k)).Sort
      With .SortFields
        .Clear
        .Add Key:=Range(A.Columns(1).Address), SortOn:=xlSortOnValues, _
          Order:=xlAscending, DataOption:=xlSortNormal
      End With
      .SetRange Range(A.Address)
      .Header = xlNo
      .MatchCase = False
      .Orientation = xlTopToBottom
      .SortMethod = xlPinYin
      .Apply
      .SortFields.Clear
    End With
  Next j
Next k

Application.Calculation = xlCalculationAutomatic
Sheets("Campaign").Select
Application.ScreenUpdating = True
End Sub
```

‘H’ Matrix Subroutine

```
Sub H_Matrix(N As Long, T As Long)
' This subroutine generates the h_mt*a_mnt objective function coefficients

Application.Calculation = xlCalculationManual
```

```
Application.ScreenUpdating = False

Dim i As Long
Dim j As Long
Dim k As Long
Dim p As Long
Dim q As Long
Dim g As Long
Dim h() As Long
Dim HRange As Range
Dim S As Range
Dim D As Range
Dim Bases As Range

' Remove old problem sheets, if there are any
i = 1
Do While i <= ThisWorkbook.Sheets.Count
  If Mid(Sheets(i).Name, 1, 2) = "h_" Then
    Sheets(i).Delete
  Else
    i = i + 1
  End If
Loop

' Set the distance matrix
i = Application.WorksheetFunction.Count(Sheets("d").Columns("A"))
Set D = Sheets("d").Range("B2:" & Range("B2").Cells(i, i).Address)

' Set the base list
i = Application.WorksheetFunction.Count(Sheets("Bases").Columns("C"))
Set Bases = Sheets("Bases").Range("A2:" & Range("A2").Cells(i, 5 + N).Address)

Set S = Sheets("s").Range("C2:" & Range("C2").Cells(N, T).Address)

For k = 1 To T Step 1
  Sheets("a_nm" & CStr(k)).Copy after:=Sheets(ThisWorkbook.Sheets.Count)
  ActiveSheet.Name = "h_nm" & CStr(k)
  For j = 1 To N Step 1
    Set HRange = Sheets("h_nm" & CStr(k)).Range("B2:C" & CStr(1 + Application.WorksheetFunction.Count( _
      Sheets("h_nm" & CStr(k)).Columns("B").Offset(0, 2 * (j - 1))))).Offset(0, 2 * (j - 1))
    ReDim h(HRange.Rows.Count, 1)
    ' If this sensor type is base-only, we make the h_mt value zero
    If CLng(S(j, 1).Offset(0, -2)) > 4 Then
      For i = 1 To HRange.Rows.Count Step 1
        If Application.WorksheetFunction.CountIf(Bases.Columns(3), "=" & CStr(HRange(i, 1).Value)) = 0 And _
          HRange(i, 2) > 0 Then
          p = HRange(i, 1)
          For q = 1 To Bases.Rows.Count Step 1
            ' If the base exists during the period and it is closer, save the distance
            If (D(p, Bases(q, 3)) < h(i, 1) Or h(i, 1) = 0) And Bases(q, 5) <= k Then h(i, 1) = D(p, Bases(q, 3))
          Next q
        End If
      Next i
      HRange.Columns(2) = h
    Else
      HRange.Columns(2) = 0
    End If
  Next j
Next k

Sheets("Campaign").Select
Application.ScreenUpdating = True

End Sub
```

BWSSP BILP Formulation Subroutine

```
Sub BWSSP_BILP(N As Long, T As Long, LOOP_COUNTER As Long, IsPreEmp As Boolean, NumPreEmp As Integer, _
            MaxCell As String)
' This subroutine creates the constraints and solves the combined BWSSP BILP using premium solver 9

With Application
   .Calculation = xlCalculationManual
   .ScreenUpdating = False
   .DisplayAlerts = False
End With

Dim StartTime As Date           ' Start time of the solver
Dim i As Long                   ' Loop counter
Dim j As Long                   ' Loop counter
Dim k As Long                   ' Loop counter
Dim p As Long                   ' Loop counter
Dim q As Long                   ' Loop counter
Dim g As Long                   ' Loop counter
Dim w As Long                   ' Loop counter
Dim y As Long                   ' Loop counter
Dim M As Long                   ' Number of locations for this sensor type at this time
Dim x As Range                  ' Range containing decision variables
Dim c As Range                  ' Range containing c obj function coefficients (Equation (16))
Dim u As Range                  ' Range containing u obj function coefficients (Equation (2))
Dim S As Range                  ' Available sensor amounts
Dim Delta As Range              ' Sensor effective ranges/minimum spacing
Dim D As Range                  ' Distance matrix
Dim cx As String                ' Excel equation for Equation (16)
Dim ux As String                ' Excel equation for Equation (2)
Dim vx As String                ' Excel equation for Equation (4)
Dim hx As String                ' Excel equation for Equation (5)
Dim Constraint As Range         ' Objective function constraints
Dim SetConstraint As Boolean    ' True if constraints are set, false otherwise
SetConstraint = False

If Not IsPreEmp Then
   On Error Resume Next
   Sheets("ObjFunc").Delete
   On Error Resume Next
   Sheets("cx").Delete
   On Error Resume Next
   Sheets("ux").Delete
   On Error Resume Next
   Sheets("vx").Delete
   On Error Resume Next
   Sheets("hx").Delete
   ' Remove old problem sheets, if there are any
   i = 1
   Do While i <= ThisWorkbook.Sheets.Count
      If Mid(Sheets(i).Name, 1, 2) = "x_" Then
         Sheets(i).Delete
      Else
         i = i + 1
      End If
   Loop
   ' Create the sheet for the IP solution
   Sheets.Add.Name = "ObjFunc"
End If

cx = ""
ux = ""
vx = ""
hx = ""

' Set the distance matrix
i = Application.WorksheetFunction.Count(Sheets("d").Columns("A"))
```

```
Set D = Sheets("d").Range("B2:" & Range("B2").Cells(i, i).Address)

' Define the S and delta value
Set S = Sheets("s").Range("C2:" & Range("C2").Cells(N, T).Address)
Set Delta = Sheets("delta").Range("A2:" & Range("B2").Cells(N, 1).Address)

' Set the solver parameters
Sheets("ObjFunc").Select
Sheets("Objfunc").Range("A1") = LOOP_COUNTER - 3
Sheets("Objfunc").Range("B1") = Application.WorksheetFunction.Count(Sheets("Campaign").Columns("A"))
SolverReset                ' Reset all solver parameters
SolverOptions AssumeLinear:=True, Scaling:=True, BypassReports:=True
SolverModel Interpreter:=1, CheckFor:=4, SolveTransformed:=True, ShowTransformations:=False, _
  ShowExceptions:=False, DesiredModel:=1, Interactive:=False, UsePsiFunctions:=True, Engines:=1, ReqSmooth:=True, _
  FastSetup:=True, Sparse:=True, ActiveOnly:=False
SolverLPOptions Scaling:=True, AssumeNonNeg:=True, BypassReports:=True, Derivatives:=1, Presolve:=True

' Create decision variables and multiply them by the combined objective function coefficients
For k = 1 To T Step 1
  If Not IsPreEmp Then
    Sheets("u_nm" & CStr(k)).Copy after:=Sheets(ThisWorkbook.Sheets.Count)
    ActiveSheet.Name = "x_nm" & CStr(k)
    Sheets("x_nm" & CStr(k)).Copy after:=Sheets(ThisWorkbook.Sheets.Count)
    ActiveSheet.Name = "cx" & CStr(k)
    Sheets("cx" & CStr(k)).Copy after:=Sheets(ThisWorkbook.Sheets.Count)
    ActiveSheet.Name = "ux" & CStr(k)
    Sheets("x_nm" & CStr(k)).Copy after:=Sheets(ThisWorkbook.Sheets.Count)
    ActiveSheet.Name = "vx" & CStr(k)
    Sheets("x_nm" & CStr(k)).Copy after:=Sheets(ThisWorkbook.Sheets.Count)
    ActiveSheet.Name = "hx" & CStr(k)

    ' Set the objective function coefficients to their fixed values
    Sheets("u_nm" & CStr(k)).Select
    Cells.Select
    Selection.Copy
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
      :=False, Transpose:=False
    Sheets("h_nm" & CStr(k)).Select
    Cells.Select
    Selection.Copy
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
      :=False, Transpose:=False
    Sheets("v_nm" & CStr(k)).Select
    Cells.Select
    Selection.Copy
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
      :=False, Transpose:=False
    Sheets("c_nm" & CStr(k)).Select
    Cells.Select
    Selection.Copy
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
     :=False, Transpose:=False
  End If

  ' Build the objective function equations
  For j = 1 To N Step 1
    M = Application.WorksheetFunction.Count(Sheets("x_nm" & CStr(k)).Columns("B").Offset(0, (j - 1) * 2))
    If M > 0 Then
      Set x = Sheets("x_nm" & CStr(k)).Range(Range("B2").Cells(1, 2 * j - 1).Address & _
         ":" & Range("B2").Cells(M, 2 * j).Address)
      x.Columns(2).Value = 0
      If Not IsPreEmp Then
        For i = 1 To M Step 1
          Sheets("cx" & CStr(k)).Range(x(1, 2).Address).Cells(i, 1).Formula = _
            "=c_nm" & CStr(k) & "!" & x(i, 2).Address & "*" & "x_nm" & CStr(k) & "!" & x(i, 2).Address
          Sheets("ux" & CStr(k)).Range(x(1, 2).Address).Cells(i, 1).Formula = _
            "=u_nm" & CStr(k) & "!" & x(i, 2).Address & "*" & "x_nm" & CStr(k) & "!" & x(i, 2).Address
          Sheets("vx" & CStr(k)).Range(x(1, 2).Address).Cells(i, 1).Formula = _
```

```
            "=v_nm" & CStr(k) & "!" & x(i, 2).Address & "*" & "x_nm" & CStr(k) & "!" & x(i, 2).Address
          Sheets("hx" & CStr(k)).Range(x(1, 2).Address).Cells(i, 1).Formula = _
            "=h_nm" & CStr(k) & "!" & x(i, 2).Address & "*" & "x_nm" & CStr(k) & "!" & x(i, 2).Address
        Next i
        cx = cx & "SUM(" & "cx" & CStr(k) & "!" & x.Columns(2).Address & ")+"
        ux = ux & "SUM(" & "ux" & CStr(k) & "!" & x.Columns(2).Address & ")+"
        vx = vx & "SUM(" & "vx" & CStr(k) & "!" & x.Columns(2).Address & ")+"
        hx = hx & "SUM(" & "hx" & CStr(k) & "!" & x.Columns(2).Address & ")+"
      End If
      Sheets("ObjFunc").Select
      If Not SetConstraint Then
        ' Set the solver
        SolverOK SetCell:=MaxCell, MaxMinVal:=1, Valueof:=0, ByChange:="x_nm" & CStr(k) & "!" & x.Columns(2).Address, _
          Engine:=9, EngineDesc:="Large-Scale LP Solver"
        SetConstraint = True
      Else
        '  Add the decision variables for period k
        SolverAdd CellRef:="x_nm" & CStr(k) & "!" & x.Columns(2).Address, Comment:="Period " & CStr(k) & " Decision
Variables"
      End If
      ' Constrain the decision variables to be binary
      SolverAdd CellRef:="x_nm" & CStr(k) & "!" & x.Columns(2).Address, _
        Relation:=5, FormulaText:="", Comment:="Period " & CStr(k) & " Decision Variables Binary Constraint"
    End If
  Next j
Next k

If Not IsPreEmp Then
  i = 1
  ' Delete data sheets that we no longer need
  Do While i <= ThisWorkbook.Sheets.Count
    If Not Sheets(i).Visible Or (Not (Mid(Sheets(i).Name, 1, 1) = "c" Or Mid(Sheets(i).Name, 1, 1) = "u" Or Mid(Sheets(i).Name, 1,
1) = "v" Or _
      Mid(Sheets(i).Name, 1, 1) = "h" Or Mid(Sheets(i).Name, 1, 1) = "x") And _
      IsNumeric(Mid(Sheets(i).Name, Len(Sheets(i).Name), 1))) Then
      Sheets(i).Delete
    Else
      i = i + 1
    End If
  Loop

  ' Set up the combined objective function value
  Sheets("ObjFunc").Range("B2") = "Obj Function Value"
  Sheets("ObjFunc").Range("B3").Formula = "=" & Mid(cx, 1, Len(cx) - 1)

  ' Set up the u objective function value
  Sheets("ObjFunc").Range("D2") = "u Obj Function Value"
  Sheets("ObjFunc").Range("D3").Formula = "=" & Mid(ux, 1, Len(ux) - 1)

  ' Set up the u objective function value
  Sheets("ObjFunc").Range("F2") = "v Obj Function Value"
  Sheets("ObjFunc").Range("F3").Formula = "=" & Mid(vx, 1, Len(vx) - 1)

  ' Set up the u objective function value
  Sheets("ObjFunc").Range("H2") = "h Obj Function Value"
  Sheets("ObjFunc").Range("H3").Formula = "=" & Mid(hx, 1, Len(hx) - 1)

  ' Add in the constraints to the ObjFunc sheet
  Sheets("ObjFunc").Range("B5") = "Constraints"
  Set Constraint = Sheets("ObjFunc").Range("B8")
  Constraint(1, 1).Offset(-1, 0) = "m"
  Constraint(1, 2).Offset(-1, 0) = "Equation (17) - 1 SFC Per Loc"
  q = 1
  For i = 1 To D.Rows.Count Step 1
    For k = 1 To T Step 1
      For j = 1 To N Step 1
        M = Application.WorksheetFunction.Count(Sheets("c_nm" & CStr(k)).Columns("B").Offset(0, (j - 1) * 2))
        If M > 0 And (S(j, 1).Offset(0, -2) = 3 Or S(j, 1).Offset(0, -2) = 7) Then
```

```
          Set c = Sheets("c_nm" & CStr(k)).Range(Range("B2").Cells(1, 2 * j - 1).Address & _
            ":" & Range("B2").Cells(M, 2 * j).Address)
          If c(i, 1) = D(i, 1).Offset(0, -1) Then
            p = i
          Else
            p = 1
            Do While c(p, 1) < D(i, 1).Offset(0, -1) And p < M
              p = p + 1
            Loop
          End If
          If c(p, 1) = D(i, 1).Offset(0, -1) And c(p, 2) > 0 Then
            Constraint(q, 3) = Constraint(q, 3) & "x_nm" & CStr(k) & "!" & c(p, 2).Address & "+"
          End If
        End If
      Next j
    Next k
    If Not Trim(Constraint(q, 3)) = "" Then
      Constraint(q, 1) = D(i, 1).Offset(0, -1)
      Constraint(q, 2).Formula = "=" & Mid(CStr(Constraint(q, 3)), 1, Len(CStr(Constraint(q, 3))) - 1)
      Constraint(q, 3) = 1
      q = q + 1
    End If
  Next i

  Set Constraint = Constraint.Offset(0, 4)
  Constraint(1, 1).Offset(-1, 0) = "m"
  Constraint(1, 2).Offset(-1, 0) = "Equation (18) - 1 Balloon Per Loc"
  q = 1

  For i = 1 To D.Rows.Count Step 1
    For k = 1 To T Step 1
      For j = 1 To N Step 1
        M = Application.WorksheetFunction.Count(Sheets("c_nm" & CStr(k)).Columns("B").Offset(0, (j - 1) * 2))
        If M > 0 And S(j, 1).Offset(0, -2) Mod 2 = 0 Then
          Set c = Sheets("c_nm" & CStr(k)).Range(Range("B2").Cells(1, 2 * j - 1).Address & _
            ":" & Range("B2").Cells(M, 2 * j).Address)
          If c(i, 1) = D(i, 1).Offset(0, -1) Then
            p = i
          Else
            p = 1
            Do While c(p, 1) < D(i, 1).Offset(0, -1) And p < M
              p = p + 1
            Loop
          End If
          If c(p, 1) = D(i, 1).Offset(0, -1) And c(p, 2) > 0 Then
            Constraint(q, 3) = Constraint(q, 3) & "x_nm" & CStr(k) & "!" & c(p, 2).Address & "+"
          End If
        End If
      Next j
    Next k
    If Not Trim(Constraint(q, 3)) = "" Then
      Constraint(q, 1) = D(i, 1).Offset(0, -1)
      Constraint(q, 2) = "=" & Mid(CStr(Constraint(q, 3)), 1, Len(CStr(Constraint(q, 3))) - 1)
      Constraint(q, 3) = 1
      q = q + 1
    End If
  Next i

  Set Constraint = Constraint.Offset(0, 4)
  Constraint(1, 1).Offset(-1, 0) = "m"
  Constraint(1, 2).Offset(-1, 0) = "Equation (19) - 1 Radar Per Loc"
  q = 1
  For i = 1 To D.Rows.Count Step 1
    For k = 1 To T Step 1
      For j = 1 To N Step 1
        M = Application.WorksheetFunction.Count(Sheets("c_nm" & CStr(k)).Columns("B").Offset(0, (j - 1) * 2))
        If M > 0 And (S(j, 1).Offset(0, -2) = 1 Or S(j, 1).Offset(0, -2) = 5) Then
          Set c = Sheets("c_nm" & CStr(k)).Range(Range("B2").Cells(1, 2 * j - 1).Address & _
```

```
          ":" & Range("B2").Cells(M, 2 * j).Address)
        If c(i, 1) = D(i, 1).Offset(0, -1) Then
          p = i
        Else
          p = 1
          Do While c(p, 1) < D(i, 1).Offset(0, -1) And p < M
            p = p + 1
          Loop
        End If
        If c(p, 1) = D(i, 1).Offset(0, -1) And c(p, 2) > 0 Then
          Constraint(q, 3) = Constraint(q, 3) & "x_nm" & CStr(k) & "!" & c(p, 2).Address & "+"
        End If
      End If
    Next j
  Next k
  If Not Trim(Constraint(q, 3)) = "" Then
    Constraint(q, 1) = D(i, 1).Offset(0, -1)
    Constraint(q, 2) = "=" & Mid(CStr(Constraint(q, 3)), 1, Len(CStr(Constraint(q, 3))) - 1)
    Constraint(q, 3) = 1
    q = q + 1
  End If
Next i


Set Constraint = Constraint.Offset(0, 4)
Constraint(1, 1).Offset(-2, 0) = "Equations (20) and (21) - Resource Constraints"
For k = 1 To T Step 1
  Constraint(k, 1) = "t = " & CStr(k)
  For i = 1 To N Step 1
    M = Application.WorksheetFunction.Count(Sheets("x_nm" & CStr(k)).Columns("B").Offset(0, 2 * (i - 1)))
    If k = 1 Then
      Constraint(1, (i - 1) * 2 + 2).Offset(-1, 0) = "n = " & CStr(i)
      If M > 0 Then
        Constraint(k, (i - 1) * 2 + 2) = "=SUM(x_nm" & CStr(k) & "!" & Range(Range("B2").Cells(1, 2 * i), _
          Range("B2").Cells(M, 2 * i)).Address & ")"
      Else
        Constraint(k, (i - 1) * 2 + 2) = 0
      End If
    Else
      If M > 0 Then
        Constraint(k, (i - 1) * 2 + 3) = "-("
        For j = k - 1 To 1 Step -1
          Constraint(k, (i - 1) * 2 + 3) = Constraint(k, (i - 1) * 2 + 3) & _
            "(s!" & S(i, j).Address & "-SUM(x_nm" & CStr(j) & "!" & Range(Range("B2").Cells(1, 2 * i), _
            Range("B2").Cells(M, 2 * i)).Address & ")"
          If j > 1 Then
            Constraint(k, (i - 1) * 2 + 3) = Constraint(k, (i - 1) * 2 + 3) & ")+"
          Else
            Constraint(k, (i - 1) * 2 + 3) = Constraint(k, (i - 1) * 2 + 3) & "))"
          End If
        Next j
        Constraint(k, (i - 1) * 2 + 2) = "=SUM(x_nm" & CStr(k) & "!" & Range(Range("B2").Cells(1, 2 * i), _
          Range("B2").Cells(M, 2 * i)).Address & ")" & Constraint(k, (i - 1) * 2 + 3)
      Else
        Constraint(k, (i - 1) * 2 + 2) = 0
      End If
    End If
    Constraint(k, (i - 1) * 2 + 3) = "=s!" & S(i, k).Address
  Next i
Next k

Set Constraint = Constraint.Offset(0, 2 * (N + 1))
Constraint(1, 1).Offset(-1, 0) = "Equation (22) - Balloon Min Spacing"
If Application.WorksheetFunction.CountIf(S.Columns(1).Offset(0, -2), "=2") + _
  Application.WorksheetFunction.CountIf(S.Columns(1).Offset(0, -2), "=4") + _
  Application.WorksheetFunction.CountIf(S.Columns(1).Offset(0, -2), "=6") + _
  Application.WorksheetFunction.CountIf(S.Columns(1).Offset(0, -2), "=8") > 0 Then
  q = 1
```

```
' If there are balloons, find the first balloon with an effective range greater than zero
j = 1
Do While j <= N And (Delta(j, 2) = 0 Or S(j, 1).Offset(0, -2) Mod 2 <> 0)
  j = j + 1
Loop
If j <= N Then
' If we have found a balloon with effective range > 0, iterate through the distance matrix
  For w = 1 To D.Rows.Count - 1 Step 1
  ' Loop through rows of distance matrix
    For y = w + 1 To D.Columns.Count Step 1
    ' Loop through columns (upper half) of distance matrix
      For i = j To N Step 2
      ' Loop through remainder of sensor list, looking for balloons with effective range > 0
        If S(i, 1).Offset(0, -2) Mod 2 = 0 And Delta(i, 2) > 0 Then
        ' If sensor i is an upper air sensor with effective range greater than zero
          For g = i To N Step 1
          ' Loop through remainder of sensors, finding upper-air sensors with effective range > 0
          ' check current distance with combination of effective ranges
            If S(g, 1).Offset(0, -2) Mod 2 = 0 And (Delta(g, 2) > D(w, y) Or Delta(i, 2) > D(w, y)) Then
            ' If one of the balloon's minimum spacing requirements is greater than the distance
            ' between locations w and y set the constraint that w and y can only have one sensor between them
              For k = 1 To T Step 1
              ' Loop through time periods
                M = Application.WorksheetFunction.Count(Sheets("c_nm" & CStr(k)).Columns("B").Offset(0, 2 * (i - 1)))
                If M > 0 Then
                ' If there are any locations available for allocation for this sensor type and time period
                  Set c = Sheets("c_nm" & CStr(k)).Range(Range("B2").Cells(1, 2 * (i - 1) + 1).Address & _
                    ":" & Range("B2").Cells(M, 2 * i).Address)
                  ' Find the sensor type i at location w
                  If c(w, 1) = D(w, 1).Offset(0, -1) Then
                    p = w
                  Else
                    p = 1
                    Do While c(p, 1) < D(w, 1).Offset(0, -1) And p < M
                      p = p + 1
                    Loop
                  End If
                  If c(p, 1) = D(w, 1).Offset(0, -1) And c(p, 2) > 0 Then
                    Constraint(q, 2) = Constraint(q, 2) & "x_nm" & CStr(k) & "!" & c(p, 2).Address & "+"
                  End If
                End If
                M = Application.WorksheetFunction.Count(Sheets("c_nm" & CStr(k)).Columns("B").Offset(0, 2 * (g - 1)))
                If M > 0 Then
                  Set c = Sheets("c_nm" & CStr(k)).Range(Range("B2").Cells(1, 2 * (g - 1) + 1).Address & _
                    ":" & Range("B2").Cells(M, 2 * g).Address)
                  ' Find the sensor type g at location y
                  If c(y, 1) = D(y, 1).Offset(0, -1) Then
                    p = y
                  Else
                    p = 1
                    Do While c(p, 1) < D(y, 1).Offset(0, -1) And p < M
                      p = p + 1
                    Loop
                  End If
                  If c(p, 1) = D(y, 1).Offset(0, -1) And c(p, 2) > 0 Then
                    Constraint(q, 3) = Constraint(q, 3) & "x_nm" & CStr(k) & "!" & c(p, 2).Address & "+"
                  End If
                End If
              Next k
              If Not Trim(Constraint(q, 2)) = "" And Not Trim(Constraint(q, 3)) = "" Then
                Constraint(q, 2) = "=" & Mid(Constraint(q, 2) & Constraint(q, 3), 1, _
                  Len(Constraint(q, 2) & Constraint(q, 3)) - 1)
                Constraint(q, 3) = 1
                Constraint(q, 1) = "n = " & CStr(i) & " at m = " & D(w, 1).Offset(0, -1) & _
                           " + n = " & CStr(g) & " at m = " & D(y, 1).Offset(0, -1)
                q = q + 1
              Else
```

```
              Constraint(q, 2) = ""
              Constraint(q, 3) = ""
            End If
            If i <> g Then
            ' If the sensors types are not the same, we have to add a constraint for the reverse problem
              For k = 1 To T Step 1
              ' Loop through time periods
                M = Application.WorksheetFunction.Count(Sheets("c_nm" & CStr(k)).Columns("B").Offset(0, 2 * (g - 1)))
                If M > 0 Then
                ' If there are any locations available for allocation for this sensor type and time period
                  Set c = Sheets("c_nm" & CStr(k)).Range(Range("B2").Cells(1, 2 * g - 1).Address & _
                    ":" & Range("B2").Cells(M, 2 * g).Address)
                  ' Find the sensor type g at location w
                  If c(w, 1) = D(w, 1).Offset(0, -1) Then
                    p = w
                  Else
                    p = 1
                    Do While c(p, 1) < D(w, 1).Offset(0, -1) And p < M
                      p = p + 1
                    Loop
                  End If
                  If c(p, 1) = D(w, 1).Offset(0, -1) And c(p, 2) > 0 Then
                    Constraint(q, 2) = Constraint(q, 2) & "x_nm" & CStr(k) & "!" & c(p, 2).Address & "+"
                  End If
                End If
                M = Application.WorksheetFunction.Count(Sheets("c_nm" & CStr(k)).Columns("B").Offset(0, 2 * (i - 1)))
                If M > 0 Then
                  Set c = Sheets("c_nm" & CStr(k)).Range(Range("B2").Cells(1, 2 * i - 1).Address & _
                    ":" & Range("B2").Cells(M, 2 * i).Address)
                   ' Find the sensor type i at location y
                   If c(y, 1) = D(y, 1).Offset(0, -1) Then
                     p = y
                   Else
                     p = 1
                     Do While c(p, 1) < D(y, 1).Offset(0, -1) And p < M
                      p = p + 1
                    Loop
                  End If
                  If c(p, 1) = D(y, 1).Offset(0, -1) And c(p, 2) > 0 Then
                    Constraint(q, 3) = Constraint(q, 3) & "x_nm" & CStr(k) & "!" & c(p, 2).Address & "+"
                  End If
                End If
              Next k
              If Not Trim(Constraint(q, 2)) = "" And Not Trim(Constraint(q, 3)) = "" Then
                Constraint(q, 2) = "=" & Mid(Constraint(q, 2) & Constraint(q, 3), 1, _
                  Len(Constraint(q, 2) & Constraint(q, 3)) - 1)
                Constraint(q, 3) = 1
                Constraint(q, 1) = "n = " & CStr(g) & " at m = " & D(w, 1).Offset(0, -1) & _
                         " + n = " & CStr(i) & " at m = " & D(y, 1).Offset(0, -1)
                q = q + 1
              Else
                Constraint(q, 2) = ""
                Constraint(q, 3) = ""
              End If
            End If
          End If
        Next g
      End If
    Next i
  Next y
 Next w
 End If
End If

Set Constraint = Constraint.Offset(0, 4)
Constraint(1, 1).Offset(-1, 0) = "Equation (23) - Radar Effective Range"
If Application.WorksheetFunction.CountIf(S.Columns(1).Offset(0, -2), "=1") + _
  Application.WorksheetFunction.CountIf(S.Columns(1).Offset(0, -2), "=5") > 0 Then
```

```
q = 1
' If there are radars, find the first radar with an effective range greater than zero
j = 1
Do While j <= N And (Delta(j, 2) = 0 Or Not (S(j, 1).Offset(0, -2) = 5 Or S(j, 1).Offset(0, -2) = 1))
  j = j + 1
Loop
q = 1
If j <= N Then
' If we have found a radar with effective range > 0, iterate through the distance matrix
  For w = 1 To D.Rows.Count - 1 Step 1
  ' Loop through rows of distance matrix
    For y = w + 1 To D.Columns.Count Step 1
    ' Loop through columns (upper half) of distance matrix
      For i = j To N Step 2
      ' Loop through remainder of sensor list, looking for balloons with effective range > 0
        If (S(j, 1).Offset(0, -2) = 5 Or S(j, 1).Offset(0, -2) = 1) And Delta(i, 2) > 0 Then
        ' If sensor i is an upper air sensor with effective range greater than zero
          For g = i To N Step 1
          ' Loop through remainder of sensors, finding radars with effective range > 0
          ' check current distance with combination of effective ranges
            If (S(j, 1).Offset(0, -2) = 5 Or S(j, 1).Offset(0, -2) = 1) And Delta(g, 2) + Delta(i, 2) > D(w, y) Then
            ' If the combined sensor effective ranges is greater than the distance between locations w and y
            ' set the constraint that w and y can only have one sensor between them
              For k = 1 To T Step 1
              ' Loop through time periods
                M = Application.WorksheetFunction.Count(Sheets("c_nm" & CStr(k)).Columns("B").Offset(0, 2 * (i - 1)))
                If M > 0 Then
                ' If there are any locations available for allocation for this sensor type and time period
                  Set c = Sheets("c_nm" & CStr(k)).Range(Range("B2").Cells(1, 2 * (i - 1) + 1).Address & _
                    ":" & Range("B2").Cells(M, 2 * i).Address)
                  ' Find the sensor type i at location w
                  If c(w, 1) = D(w, 1).Offset(0, -1) Then
                    p = w
                  Else
                    p = 1
                    Do While c(p, 1) < D(w, 1).Offset(0, -1) And p < M
                      p = p + 1
                    Loop
                  End If
                  If c(p, 1) = D(w, 1).Offset(0, -1) And c(p, 2) > 0 Then
                    Constraint(q, 2) = Constraint(q, 2) & "x_nm" & CStr(k) & "!" & c(p, 2).Address & "+"
                  End If
                End If
                M = Application.WorksheetFunction.Count(Sheets("c_nm" & CStr(k)).Columns("B").Offset(0, 2 * (g - 1)))
                If M > 0 Then
                  Set c = Sheets("c_nm" & CStr(k)).Range(Range("B2").Cells(1, 2 * (g - 1) + 1).Address & _
                    ":" & Range("B2").Cells(M, 2 * g).Address)
                  ' Find the sensor type g at location y
                  If c(y, 1) = D(y, 1).Offset(0, -1) Then
                    p = y
                  Else
                    p = 1
                    Do While c(p, 1) < D(y, 1).Offset(0, -1) And p < M
                      p = p + 1
                    Loop
                  End If
                  If c(p, 1) = D(y, 1).Offset(0, -1) And c(p, 2) > 0 Then
                    Constraint(q, 3) = Constraint(q, 3) & "x_nm" & CStr(k) & "!" & c(p, 2).Address & "+"
                  End If
                End If
              Next k
              If Not Trim(Constraint(q, 2)) = "" And Not Trim(Constraint(q, 3)) = "" Then
                Constraint(q, 2) = "=" & Mid(Constraint(q, 2) & Constraint(q, 3), 1, _
                  Len(Constraint(q, 2) & Constraint(q, 3)) - 1)
                Constraint(q, 3) = 1
                Constraint(q, 1) = "n = " & CStr(i) & " at m = " & D(w, 1).Offset(0, -1) & _
                           " + n = " & CStr(g) & " at m = " & D(y, 1).Offset(0, -1)
                q = q + 1
```

```
                    Else
                      Constraint(q, 2) = ""
                      Constraint(q, 3) = ""
                    End If
                    If i <> g Then
                    ' If the sensors types are not the same, we have to add a constraint for the reverse problem
                      For k = 1 To T Step 1
                      ' Loop through time periods
                        M = Application.WorksheetFunction.Count(Sheets("c_nm" & CStr(k)).Columns("B").Offset(0, 2 * (g - 1)))
                        If M > 0 Then
                        ' If there are any locations available for allocation for this sensor type and time period
                          Set c = Sheets("c_nm" & CStr(k)).Range(Range("B2").Cells(1, 2 * g - 1).Address & _
                            ":" & Range("B2").Cells(M, 2 * g).Address)
                          ' Find the sensor type g at location w
                          If c(w, 1) = D(w, 1).Offset(0, -1) Then
                            p = w
                          Else
                            p = 1
                            Do While c(p, 1) < D(w, 1).Offset(0, -1) And p < M
                              p = p + 1
                            Loop
                          End If
                          If c(p, 1) = D(w, 1).Offset(0, -1) And c(p, 2) > 0 Then
                            Constraint(q, 2) = Constraint(q, 2) & "x_nm" & CStr(k) & "!" & c(p, 2).Address & "+"
                          End If
                        End If
                        M = Application.WorksheetFunction.Count(Sheets("c_nm" & CStr(k)).Columns("B").Offset(0, 2 * (i - 1)))
                        If M > 0 Then
                          Set c = Sheets("c_nm" & CStr(k)).Range(Range("B2").Cells(1, 2 * i - 1).Address & _
                            ":" & Range("B2").Cells(M, 2 * i).Address)
                          ' Find the sensor type i at location y
                          If c(y, 1) = D(y, 1).Offset(0, -1) Then
                            p = y
                          Else
                            p = 1
                            Do While c(p, 1) < D(y, 1).Offset(0, -1) And p < M
                              p = p + 1
                            Loop
                          End If
                          If c(p, 1) = D(y, 1).Offset(0, -1) And c(p, 2) > 0 Then
                            Constraint(q, 3) = Constraint(q, 3) & "x_nm" & CStr(k) & "!" & c(p, 2).Address & "+"
                          End If
                        End If
                      Next k
                      If Not Trim(Constraint(q, 2)) = "" And Not Trim(Constraint(q, 3)) = "" Then
                        Constraint(q, 2) = "=" & Mid(Constraint(q, 2) & Constraint(q, 3), 1, _
                          Len(Constraint(q, 2) & Constraint(q, 3)) - 1)
                        Constraint(q, 3) = 1
                        Constraint(q, 1) = "n = " & CStr(g) & " at m = " & D(w, 1).Offset(0, -1) & _
                                  " + n = " & CStr(i) & " at m = " & D(y, 1).Offset(0, -1)
                        q = q + 1
                      Else
                        Constraint(q, 2) = ""
                        Constraint(q, 3) = ""
                      End If
                    End If
                  End If
                Next g
              End If
            Next i
          Next y
        Next w
      End If
    End If

    Set Constraint = Constraint.Offset(0, 4)
    Constraint(1, 2).Offset(-2, 0) = "Equation (24) - Balloon/Surface Co-Location"
    For k = 1 To T
```

```
' Loop through time periods
  Constraint(1, 1).Offset(-1, 0) = "m"
  Constraint(1, 2).Offset(-1, 0) = "t = " & CStr(k)
  q = 1
  For i = 1 To D.Rows.Count Step 1
  ' Loop through locations in the distance matrix
    For j = 1 To N Step 1
    ' Loop through sensor types
      M = Application.WorksheetFunction.Count(Sheets("c_nm" & CStr(k)).Columns("B").Offset(0, 2 * (j - 1)))
      If M > 0 Then
      ' If there are any locations available for allocation for this sensor type and time period
        Set c = Sheets("c_nm" & CStr(k)).Range(Range("B2").Cells(1, 2 * j - 1).Address & _
           ":" & Range("B2").Cells(M, 2 * j).Address)
        If c(i, 1) = D(i, 1).Offset(0, -1) Then
          p = i
        Else
          p = 1
          Do While c(p, 1) < D(i, 1).Offset(0, -1) And p < M
            p = p + 1
          Loop
        End If
        If S(j, 1).Offset(0, -2) Mod 2 = 0 And c(p, 1) = D(i, 1).Offset(0, -1) And c(p, 2) > 0 Then
        ' If the sensor is an upper-air sensor
          Constraint(q, 2) = Constraint(q, 2) & "x_nm" & CStr(k) & "!" & c(p, 2).Address & "+"
        ElseIf (S(j, 1).Offset(0, -2) = 3 Or S(j, 1).Offset(0, -2) = 7) _
          And c(p, 1) = D(i, 1).Offset(0, -1) And c(p, 2) > 0 Then
        ' If the sensor is a surface sensor
          Constraint(q, 3) = Constraint(q, 3) & "x_nm" & CStr(k) & "!" & c(p, 2).Address & "+"
          For g = k - 1 To 1 Step -1
            Set c = Sheets("c_nm" & CStr(g)).Range(Range("B2").Cells(1, 2 * (j - 1) + 1).Address & _
              ":" & Range("B2").Cells(M, 2 * j).Address)
            If c(i, 1) = D(i, 1).Offset(0, -1) Then
              p = i
            Else
              p = 1
              Do While c(p, 1) < D(i, 1).Offset(0, -1) And p < M
                p = p + 1
              Loop
            End If
            If c(p, 1) = D(i, 1).Offset(0, -1) And c(p, 2) > 0 Then
              Constraint(q, 3) = Constraint(q, 3) & "x_nm" & CStr(g) & "!" & c(p, 2).Address & "+"
            End If
          Next g
          Set c = Sheets("c_nm" & CStr(k)).Range(Range("B2").Cells(1, 2 * (j - 1) + 1).Address & _
              ":" & Range("B2").Cells(M, 2 * j).Address)
        End If
      End If
    Next j
    If Not Trim(Constraint(q, 2)) = "" And Not Trim(Constraint(q, 3)) = "" Then
      Constraint(q, 2) = "=" & Mid(Constraint(q, 2), 1, Len(Constraint(q, 2)) - 1) & "-(" & _
        Mid(Constraint(q, 3), 1, Len(Constraint(q, 3)) - 1) & ")"
      Constraint(q, 3) = 0
      Constraint(q, 1) = D(i, 1).Offset(0, -1)
      q = q + 1
    Else
      Constraint(q, 2) = ""
      Constraint(q, 3) = ""
    End If
  Next i
  Set Constraint = Constraint.Offset(0, 4)
Next k

Constraint(1, 1).Offset(-1, 0) = "m"
Constraint(1, 2).Offset(-1, 0) = "Equation (25) - Radar/Surface Co-Location"
For k = 1 To T
' Loop through time periods
  Constraint(1, 1).Offset(-1, 0) = "m"
  Constraint(1, 2).Offset(-1, 0) = "t = " & CStr(k)
```

```
    q = 1
    For i = 1 To D.Rows.Count Step 1
    ' Loop through locations in the distance matrix
      For j = 1 To N Step 1
      ' Loop through sensor types
        M = Application.WorksheetFunction.Count(Sheets("c_nm" & CStr(k)).Columns("B").Offset(0, 2 * (j - 1)))
        If M > 0 Then
        ' If there are any locations available for allocation for this sensor type and time period
          Set c = Sheets("c_nm" & CStr(k)).Range(Range("B2").Cells(1, 2 * j - 1).Address & _
             ":" & Range("B2").Cells(M, 2 * j).Address)
          If c(i, 1) = D(i, 1).Offset(0, -1) Then
            p = i
          Else
            p = 1
            Do While c(p, 1) < D(i, 1).Offset(0, -1) And p < M
              p = p + 1
            Loop
          End If
          If (S(j, 1).Offset(0, -2) = 1 Or S(j, 1).Offset(0, -2) = 5) And _
            c(p, 1) = D(i, 1).Offset(0, -1) And c(p, 2) > 0 Then
          ' If the sensor is a radar
            Constraint(q, 2) = Constraint(q, 2) & "x_nm" & CStr(k) & "!" & c(p, 2).Address & "+"
          ElseIf (S(j, 1).Offset(0, -2) = 3 Or S(j, 1).Offset(0, -2) = 7) _
            And c(p, 1) = D(i, 1).Offset(0, -1) And c(p, 2) > 0 Then
          ' If the sensor is a surface sensor
            Constraint(q, 3) = Constraint(q, 3) & "x_nm" & CStr(k) & "!" & c(p, 2).Address & "+"
            For g = k - 1 To 1 Step -1
              Set c = Sheets("c_nm" & CStr(g)).Range(Range("B2").Cells(1, 2 * (j - 1) + 1).Address & _
                ":" & Range("B2").Cells(M, 2 * j).Address)
              If c(i, 1) = D(i, 1).Offset(0, -1) Then
                p = i
              Else
                p = 1
                Do While c(p, 1) < D(i, 1).Offset(0, -1) And p < M
                  p = p + 1
                Loop
              End If
              If c(p, 1) = D(i, 1).Offset(0, -1) And c(p, 2) > 0 Then
                Constraint(q, 3) = Constraint(q, 3) & "x_nm" & CStr(g) & "!" & c(p, 2).Address & "+"
              End If
            Next g
            Set c = Sheets("c_nm" & CStr(k)).Range(Range("B2").Cells(1, 2 * (j - 1) + 1).Address & _
                ":" & Range("B2").Cells(M, 2 * j).Address)
          End If
        End If
      Next j
      If Not Trim(Constraint(q, 2)) = "" And Not Trim(Constraint(q, 3)) = "" Then
        Constraint(q, 2) = "=" & Mid(Constraint(q, 2), 1, Len(Constraint(q, 2)) - 1) & "-(" & _
          Mid(Constraint(q, 3), 1, Len(Constraint(q, 3)) - 1) & ")"
        Constraint(q, 3) = 0
        Constraint(q, 1) = D(i, 1).Offset(0, -1)
        q = q + 1
      Else
        Constraint(q, 2) = ""
        Constraint(q, 3) = ""
      End If
    Next i
    Set Constraint = Constraint.Offset(0, 4)
  Next k
End If

' Now we use the Premium Solver to find a solution
' Make sure you have a reference to the Solver

For i = 1 To 3 Step 1
  M = Application.WorksheetFunction.Count(Sheets("ObjFunc").Columns("C").Offset(0, (i - 1) * 4))
  If M > 0 Then
    Set Constraint = Sheets("ObjFunc").Range("C8:" & Range("C8").Cells(M, 2).Address).Offset(0, (i - 1) * 4)
```

```
    ' If there are surface/balloon/radar sensors, add the sensor constraint (Equations 17,18,19)
      SolverAdd CellRef:=Constraint.Columns(1).Address, Relation:=1, FormulaText:=Constraint.Columns(2).Address, _
        Comment:="Equation (" & CStr(16 + i) & ")"
    End If
Next i

Set Constraint = Sheets("ObjFunc").Range("O8:" & Range("O8").Cells(T, 2).Address)
For j = 1 To N Step 1
' Add in the resource constraints (s_nt) - Equations (20) and (21)
    SolverAdd CellRef:=Constraint.Columns(1).Address, Relation:=1, FormulaText:=Constraint.Columns(2).Address, _
      Comment:="Resource Constraint for Sensor Type " & CStr(j)
    Set Constraint = Constraint.Offset(0, 2)
Next j

i = Application.WorksheetFunction.Count(Sheets("ObjFunc").Columns("O").Offset(0, 2 * (N + 1)))
If i = 0 Then i = 1
Set Constraint = Sheets("ObjFunc").Range("O8:" & Range("O8").Cells(i, 2).Address).Offset(0, 2 * (N + 1))

If Not Trim(Constraint(1, 1)) = "" Then
' If there are balloons with effective range that are greater than any distances between potential deployment locations
' add in the constraints
    SolverAdd CellRef:=Constraint.Columns(1).Address, Relation:=1, FormulaText:=Constraint.Columns(2).Address, _
      Comment:="Balloon constraints - Equation (22)"
End If

i = Application.WorksheetFunction.Count(Sheets("ObjFunc").Columns("O").Offset(0, 2 * (N + 1) + 4))
If i = 0 Then i = 1
Set Constraint = Sheets("ObjFunc").Range("O8:" & Range("O8").Cells(i, 2).Address).Offset(0, 2 * (N + 1) + 4)

If Not Trim(Constraint(1, 1)) = "" Then
' If there are radars with effective range that are greater than any distances between potential deployment locations
' add in the constraints
    SolverAdd CellRef:=Constraint.Columns(1).Address, Relation:=1, FormulaText:=Constraint.Columns(2).Address, _
      Comment:="Radar constraints - Equation (23)"
End If

For k = 1 To T Step 1
    i = Application.WorksheetFunction.Count(Sheets("ObjFunc").Columns("O").Offset(0, (2 * (N + 1)) + (4 * (k + 1))))
    If i = 0 Then i = 1
    Set Constraint = Sheets("ObjFunc").Range("O8:" & Range("O8").Cells(i, 2).Address).Offset(0, (2 * (N + 1)) + (4 * (k + 1)))

    If Not Trim(Constraint(1, 1)) = "" Then
    ' If there are radars with effective range that are greater than any distances between potential deployment locations
    ' add in the constraints
      SolverAdd CellRef:=Constraint.Columns(1).Address, Relation:=1, FormulaText:=Constraint.Columns(2).Address, _
        Comment:="SFC/Balloon co-location constraints - Equation (24)"
    End If
Next k

For k = 1 To T Step 1
    i = Application.WorksheetFunction.Count(Sheets("ObjFunc").Columns("O").Offset(0, (2 * (N + 1)) + (4 * (T + k + 1))))
    If i = 0 Then i = 1
    Set Constraint = Sheets("ObjFunc").Range("O8:" & Range("O8").Cells(i, 2).Address).Offset(0, (2 * (N + 1)) + 4 * (T + k + 1))

    If Not Trim(Constraint(1, 1)) = "" Then
    ' If there are radars with effective range that are greater than any distances between potential deployment locations
    ' add in the constraints
      SolverAdd CellRef:=Constraint.Columns(1).Address, Relation:=1, FormulaText:=Constraint.Columns(2).Address, _
        Comment:="SFC/Radar co-location constraints - Equation (25)"
    End If
Next k

If Not IsPreEmp Then
    ' Record number of decision variables
    Sheets("Campaign").Range("AC" & CStr(LOOP_COUNTER)) = SolverSizeGet(TypeNum:=1, SheetName:="Objfunc")

    ' Record number of constraints
    Sheets("Campaign").Range("AD" & CStr(LOOP_COUNTER)) = SolverSizeGet(TypeNum:=2, SheetName:="Objfunc")
```

```
Else
   If NumPreEmp >= 1 Then SolverAdd CellRef:="$D$3", Relation:=3, FormulaText:="$D$4", Comment:="Equation (2)"
   If NumPreEmp >= 2 Then SolverAdd CellRef:="$F$3", Relation:=3, FormulaText:="$F$4", Comment:="Equation (4)"
End If

Application.Calculation = xlCalculationAutomatic
StartTime = Now()
' Run the solver
SolverSolve UserFinish:=True, ShowRef:="ShowTrial"
SolverFinish KeepFinal:=1

If Not IsPreEmp Then
   ' Store the c obj function
   Sheets("Campaign").Range("R" & CStr(LOOP_COUNTER)) = Sheets("ObjFunc").Range("B3")
   ' Store the u obj function
   Sheets("Campaign").Range("S" & CStr(LOOP_COUNTER)) = Sheets("ObjFunc").Range("D3")
   ' Store the v obj function
   Sheets("Campaign").Range("T" & CStr(LOOP_COUNTER)) = Sheets("ObjFunc").Range("F3")
   ' Store the h obj function
   Sheets("Campaign").Range("U" & CStr(LOOP_COUNTER)) = Sheets("ObjFunc").Range("H3")
   ' Record the time to solve with the combined objective function values
   Sheets("Campaign").Range("Z" & CStr(LOOP_COUNTER)) = Now() - StartTime
Else
   ' Store the objective function value
   If NumPreEmp = 0 Then Sheets("Campaign").Range("V" & CStr(LOOP_COUNTER)) = Sheets("ObjFunc").Range(MaxCell)
   If NumPreEmp = 1 Then Sheets("Campaign").Range("W" & CStr(LOOP_COUNTER)) = Sheets("ObjFunc").Range(MaxCell)
   If NumPreEmp = 2 Then Sheets("Campaign").Range("X" & CStr(LOOP_COUNTER)) = Sheets("ObjFunc").Range(MaxCell)
   Sheets("ObjFunc").Range(MaxCell).Cells(2, 1) = Sheets("ObjFunc").Range(MaxCell)
   Sheets("Campaign").Range("AA" & CStr(LOOP_COUNTER)) = Sheets("Campaign").Range("AA" & CStr(LOOP_COUNTER)) +
(Now() - StartTime)
End If

Application.DisplayAlerts = True
Application.ScreenUpdating = True

End Sub
```

Ignizio and Thomas (1984) Subroutine

```
Sub IgnizioThomas(OutVar As String, LowVar As String, HighVar As String, N As Long, T As Long)
' This subroutine applies the Ignizio Thomas algorithm (1984)

Sheets("Campaign").Select
With Application
   .ScreenUpdating = False
   .Calculation = xlCalculationManual
End With

Dim i As Long
Dim j As Long
Dim k As Long
Dim M As Long
Dim temp As Variant

Dim OutRange As Range
Dim MidRange As Range
Dim HighRange As Range
Dim LowRange As Range
Dim UB As String          ' Upper bound range
Dim MaxRange As String
Dim Lambda As String
MaxRange = ""
UB = ""
Lambda = OutVar & "1!" & Range("B2").Cells(1, 2 * (N + 1)).Address

' Remove old problem sheets, if there are any
i = 1
```

```
Do While i <= ThisWorkbook.Sheets.Count
  If Mid(Sheets(i).Name, 1, Len(LowVar & HighVar)) = LowVar & HighVar Or _
    Mid(Sheets(i).Name, 1, Len(OutVar)) = OutVar Then
    Sheets(i).Delete
  Else
    i = i + 1
  End If
Loop


' Build the range containing the upper bound of the lower function
For k = 1 To T Step 1
  Sheets("u_nm" & CStr(k)).Copy after:=Sheets(ThisWorkbook.Sheets.Count)
  ActiveSheet.Name = OutVar & CStr(k)
  Sheets("u_nm" & CStr(k)).Copy after:=Sheets(ThisWorkbook.Sheets.Count)
  ActiveSheet.Name = LowVar & HighVar & CStr(k)
  Sheets(LowVar & HighVar & CStr(k)).Visible = False
  For j = 1 To 2 * N Step 2
    M = Application.WorksheetFunction.Count(Sheets(OutVar & CStr(k)).Columns("A").Offset(0, j))
    Set MidRange = Sheets(LowVar & HighVar & CStr(k)).Range(Range("B2").Cells(1, j).Address & _
      ":" & Range("B2").Cells(M, j + 1).Address)
    Set HighRange = Sheets(HighVar & CStr(k)).Range(Range("B2").Cells(1, j).Address & _
      ":" & Range("B2").Cells(M, j + 1).Address)
    Set LowRange = Sheets(LowVar & CStr(k)).Range(Range("B2").Cells(1, j).Address & _
      ":" & Range("B2").Cells(M, j + 1).Address)
    If M > 0 Then
      For i = 1 To M Step 1
        MidRange(i, 2).Formula = "=ABS(" & LowVar & CStr(k) & "!" & LowRange(i, 2).Address & "-(" & Lambda & _
          "*" & HighVar & CStr(k) & "!" & HighRange(i, 2).Address & "))"
      Next i
    End If
    MaxRange = MaxRange & "," & OutVar & CStr(k) & "!" & MidRange.Columns(2).Address
    UB = UB & "," & LowVar & HighVar & CStr(k) & "!" & MidRange.Columns(2).Address
  Next j
Next k

UB = "SUM(" & Mid(UB, 2) & ")"

' Build the final coefficient matrix
For k = 1 To T Step 1
  For j = 1 To 2 * N Step 2
    M = Application.WorksheetFunction.Count(Sheets(OutVar & CStr(k)).Columns("A").Offset(0, j))
    If M > 0 Then
      ReDim Equations(M, 1)
      Set OutRange = Sheets(OutVar & CStr(k)).Range(Range("B2").Cells(1, j).Address & _
        ":" & Range("B2").Cells(M, j + 1).Address)
      For i = 1 To M Step 1
        OutRange(i, 2).Formula = "=(1+" & UB & ")*" & HighVar & CStr(k) & "!" & OutRange(i, 2).Address & _
          "+(" & LowVar & CStr(k) & "!" & OutRange(i, 2).Address & "-" & Lambda & "*" & _
          HighVar & CStr(k) & "!" & OutRange(i, 2).Address & ")"
      Next i
    End If
  Next j
Next k

' Label the weights and solve for the optimal
Sheets(OutVar & CStr(1)).Select
Set OutRange = Sheets(OutVar & CStr(1)).Range("B2")
OutRange(1, N * 2).Offset(-1, 2) = "Lambda"
OutRange(1, N * 2).Offset(0, 2) = 0
OutRange(2, N * 2).Offset(0, 2) = "Maximum"
OutRange(3, N * 2).Offset(0, 2).Formula = "=MAX(" & Mid(MaxRange, 2) & ")"
Application.Calculation = xlCalculationAutomatic

'------------------------------------------------------------------------------------------
' Find the optimal value for lambda

j = 0
```

```
temp = OutRange(3, N * 2).Offset(0, 2)
Do While OutRange(3, N * 2).Offset(0, 2) <= temp And temp > 0
   OutRange(1, N * 2).Offset(0, 2) = 10 ^ j
   j = j + 1
Loop

OutRange(1, N * 2).Offset(0, 2) = 0

Call ImprovedGoldenSection(OutRange(1, N * 2).Offset(0, 2), 10 ^ j, 0, OutRange(3, N * 2).Offset(0, 2))

'------------------------------------------------------------------------------------------

Sheets("Campaign").Select
Application.ScreenUpdating = True

End Sub


Function MAdd(A() As Variant, B() As Variant) As Variant()
' This function performs matrix addition on two arrays

Dim i As Long
Dim j As Long
Dim c() As Variant

If UBound(A, 1) = UBound(B, 1) And UBound(A, 2) = UBound(B, 2) Then
   c = A
   For i = 1 To UBound(A, 1) Step 1
      For j = 1 To UBound(A, 2) Step 1
         c(i, j) = c(i, j) + B(i, j)
      Next j
   Next i
Else
   c(1, 1) = 0
End If

MAdd = c

End Function

Function ShowTrial(Reason As Integer)
' This function tells Premium Solver to accept the solution without input from the user
   ShowTrial = 0
End Function
```

Improved Golden Section Method Subroutine

```
Sub ImprovedGoldenSection(FX As Range, u As Double, L As Double, FunctionVal As Range)
' This subroutine implements the improved golden section line search method created
' by Edgar Den Boef and Dick Den Hertog (2007)

Dim tau As Double
tau = (5 ^ 0.5 - 1) / 2 ' Golden section interval
Dim x As Double
Dim x1 As Double        ' Variable
Dim x2 As Double        ' Variable
Dim FL As Double        ' Function value at variable value L
Dim FU As Double        ' Function value at variable value U
Dim i As Double         ' Loop counter
Dim FX1 As Double       ' Function value for x1
Dim FX2 As Double       ' Function value for x2

'--------------------------------------------------------------------------------
' Initialization (Algorithm 2 in the paper)

'Determine f(L) And f(U)
FX(1, 1) = L
```

```
FL = FunctionVal(1, 1)
FX(1, 1) = u
FU = FunctionVal(1, 1)

Do
  If FL < FU Then
    x = u - tau * (u - L)
    ' Determine f(x)
    FX(1, 1) = x
    FX1 = FunctionVal(1, 1)
  Else
    x = L + tau * (u - L)
    ' Determine f(x)
    FX(1, 1) = x
    FX1 = FunctionVal(1, 1)
  End If
  ' Determine new interval of uncertainty [L', U'] using convexity property
  If FX1 > FU Then
    L = x
    FL = FX1
  End If
  If FX1 > FL Then
    u = x
    FU = FX1
  End If
Loop Until u - L < 0.001 Or (FX1 <= FL And FX1 <= FU)

'-------------------------------------------------------------------------------
' The improved golden section method (Algorithm 1 in the paper)
FX2 = FX1

Do While u - L > 0.001
  If x = u - tau * (u - L) Then
    x1 = x
    x2 = L + tau * (u - L)
    ' Determine f(x2)
    FX(1, 1) = x2
    FX2 = FunctionVal(1, 1)
  ElseIf x = L + tau * (u - L) Then
    x1 = u - tau * (u - L)
    x2 = x
    ' Determine f(x1)
    FX(1, 1) = x1
    FX1 = FunctionVal(1, 1)
  End If
  ' Determine new interval of uncertainty [L', U'] using convexity property
  If FX2 > FX1 Then
    u = x2
    FU = FX2
    x2 = x1
    FX2 = FX1
  ElseIf FX2 < FX1 Then
    L = x1
    FL = FX1
    x1 = x2
    FX1 = FX2
  End If
  ' Stretch [L, U'] to [L'', U''] according to (1) and (2) to maintain golden selection property
  If x1 <= u - tau * (u - L) Then
    L = u - (1 / tau) * (u - x1)
    FX(1, 1) = L
    FL = FunctionVal(1, 1)
  ElseIf u - tau * (u - L) < x1 And x1 < (1 / 2) * (u + L) Then
    u = (1 / (1 - tau)) * (x1 - tau * L)
    FX(1, 1) = u
    FU = FunctionVal(1, 1)
  ElseIf (1 / 2) * (u + L) <= x1 And x1 < L + tau * (u - L) Then
    L = (1 / (1 - tau)) * (x1 - tau * u)
```

```
    FX(1, 1) = L
    FL = FunctionVal(1, 1)
  ElseIf x1 >= L + tau * (u - L) Then
    u = L + (1 / tau) * (x1 - L)
    FX(1, 1) = u
    FU = FunctionVal(1, 1)
  End If
  If L <= x1 And x1 <= u Then
    x = x1
  ElseIf L <= x2 And x2 <= u Then
    x = x2
  End If
Loop

'-------------------------------------------------------------------------------

' Golden search is for continuous functions, so now we search for the best integer value
' within the range [L, U]
FX(1, 1) = Application.WorksheetFunction.Floor(L, 1)
FL = FunctionVal(1, 1)
For i = Application.WorksheetFunction.Floor(L, 1) - 1 To Application.WorksheetFunction.Ceiling(L, 1) Step 1
  FX(1, 1) = i
  If FunctionVal(1, 1) <= FL Then
    FL = FunctionVal(1, 1)
    x = FX(1, 1)
  End If
Next i

FX(1, 1) = x

End Sub
```

Appendix B. "Blue Dart" Op-ED Column

In the early hours of January 24th, 2009, a convoy was travelling about 30 miles from an American firebase in the Afghan province of Farah. Around 1 a.m., the pavement beneath the fourth of the four vehicles detonated, flipping the six-ton humvee end-over-end three times and throwing the vehicle's engine thirty feet. In that vehicle were five Marines and SrA Alex Eudy, an Air Force Special Operations Weather Journeyman. The Marines in his patrol removed him from the wreckage. When he regained consciousness, SrA Eudy had dozens of fractures in both legs and a severe cut in his chin. He was severely wounded, but SrA Eudy went to work. Following his training, he checked his remaining weapon, an M-9 pistol, and then began applying his combat lifesaver training to himself and those around him. When the Marines in his patrol called for medical evacuation helicopters, SrA Eudy taught them how to use his weather observing equipment to pass detailed weather information to the inbound aircraft. In the harsh Afghan terrain, this information is critical to safe helicopter extraction. He was two months into his first deployment.

Thankfully, SrA Eudy survived his wounds. The Air Force News Service reports that he is in good spirits and is now starting the long road to recovery. SrA Eudy is an outstanding Airman, but he is not alone. He is at least the third special operations weatherman to receive the Purple Heart since September 11th, 2001. These brave Airmen are the 21st Century heirs to a tradition started by Office of Strategic Services weathermen in Axis-controlled regions of World War II Europe and continued by Air Commando weathermen in remote parts of the jungles of Vietnam. Modern special operations weathermen are the only personnel in the world trained to gather detailed, scientific environmental measurements from remote areas of the globe, where all other technical means cannot reach. The data they provide must be accurate and timely so that it can be included in theater-level numerical weather prediction computer models, thereby improving their accuracy. These computer models are used by all friendly forces in the operational theater. Additionally, when numerical weather prediction computer models are not available; these Airmen possess the technical and tactical knowledge to analyze limited environmental trend data to warn commanders of approaching conditions that can affect friendly and enemy forces' operations.

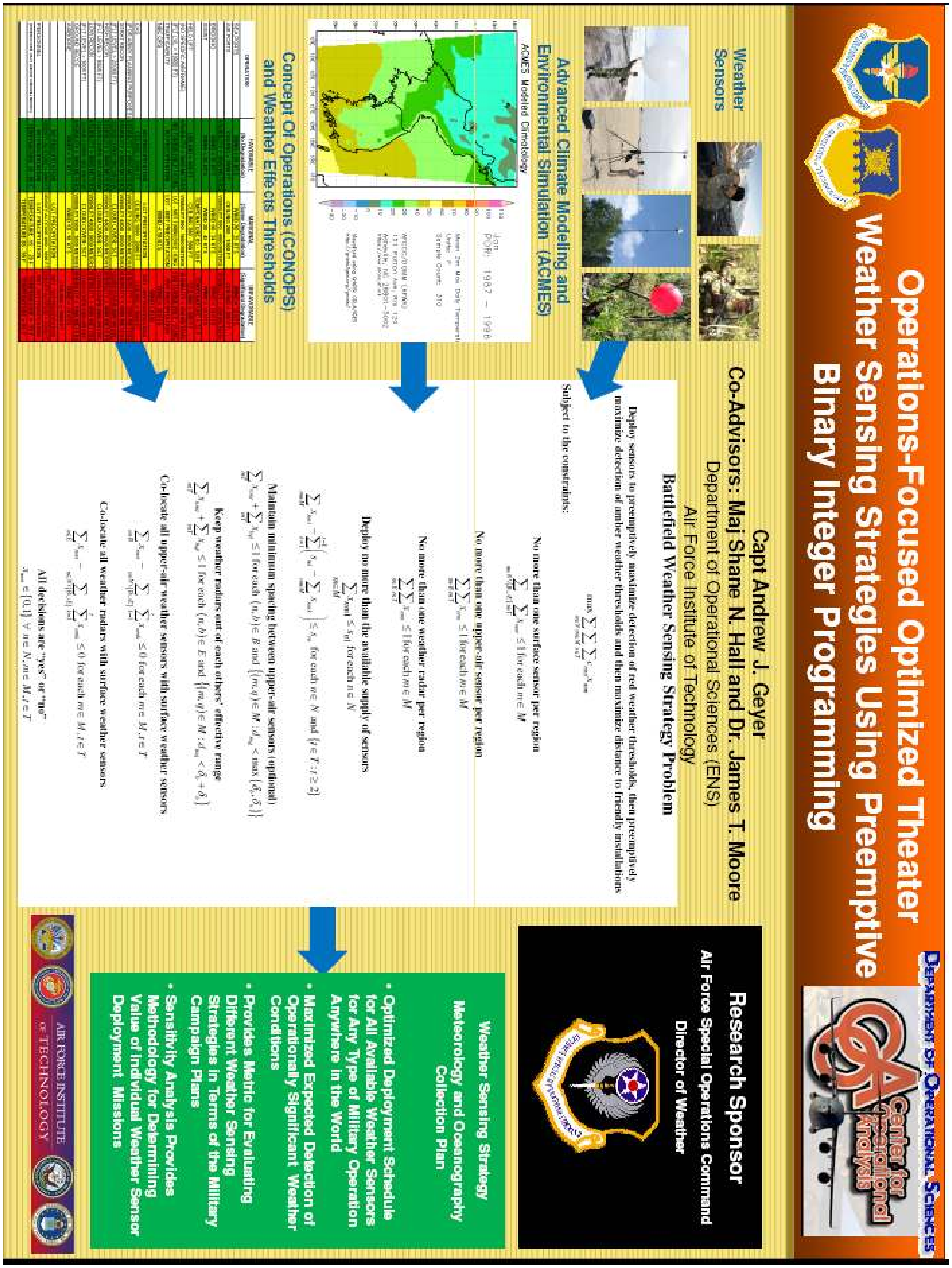
Gathering detailed measurements of battlefield environmental conditions is an important and dangerous military mission. Joint Publication 3-59 *Meteorological and Oceanographic Operations* requires that all meteorological data collection on the battlefield be coordinated in a single Joint plan. Each branch of the Department of Defense has its own set of meteorological sensors, each with its own detection capabilities, operational lifespan and operational requirements. Special operations weathermen are the only weather personnel authorized to deploy themselves or their sensors outside of the wire to gather meteorological data. These Airmen are rare and their equipment is expensive. Presently, there is no standardized methodology for determining the best time and place for the deployment of meteorological sensing equipment of any type. Any meteorological collection plan is left up to the judgment of the senior weather officer in the theater, with the approval of the overall operational commander. Consequently, there is no standardized metric for comparing the benefits of different collection plans. There is also no way to demonstrate any collection plan's contribution towards the overall campaign goals.

At the Air Force Institute of Technology's Center for Operational Analysis, a mathematical model was developed that takes any given military campaign plan, statistics compiled from climatological models, the numbers of available meteorological sensors and the capabilities of weather sensor types to create an optimized meteorological collection plan for any battlefield - anywhere. This model ensures that the available sensors are deployed in a manner that maximizes frequency of detection of operationally significant weather conditions over the longest period of time possible with the given set of sensors. This model also ensures that the meteorological collection plan is directly supporting the overall campaign plan. This model must be implemented as soon as possible so that Airmen like SrA Eudy are only ever put in harm's way for the best of reasons.

Captain Geyer is a former Special Operations Weather Team leader and graduate student in Operations Research at the Air Force Institute of Technology.

Feb 08

Appendix C. Story Board PowerPoint® Slide



Bibliography

14th Weather Squadron (AFWA) - USAF. "An Introduction to ACMES." *Strategic Climate Information Service.* https://notus2.afccc.af.mil/SCIS/data/acmes/ACMES_basics.doc (accessed September 12, 2008).

14th Weather Squadron (AFWA). *Strategic Climate Information Service.* https://notus2.afccc.af.mil/SCIS/spatial/acmes/data/products10/KOR4B/KOR4B01/monthly/KOR4B0125005.gif (accessed September 11, 2008).

AFMAN 15-111. "Surface Weather Observations." *AFMAN 15-111.* Washington, District of Columbia: United States Air Force, July 8, 2008.

AFMAN 15-129. "Air and Space Weather Operations - Processes and Procedures." *AFMAN 15-129.* Washington: United States Air Force, June 21, 2004.

Ahrens, C. Donald. *Meteorology Today - An Introduction to Weather, Climate and the Environment.* Pacific Grove, California: Brooks/Cole, 2000.

Central Intelligence Agency. *Afghanistan.* September 4, 2008. https://www.cia.gov/library/publications/the-world-factbook/geos/af.html (accessed September 13, 2008).

Das, Indraneel. *Multi-Objective Optimization.* January 22, 1997. http://www-new.mcs.anl.gov/otc/Guide/OptWeb/multiobj/ (accessed February 27, 2009).

Den Boef, Edgar, and Dick Den Hertog. "Efficient Line Search Methods for Convex Functions." *SIAM Journal of Optimization* 18, no. 1 (2007): 338-363.

Electronics. *Gizmag - Emerging Technology Magazine.* January 7, 2007. http://www.gizmag.com/pictures/hero/8593_7010865934.jpg (accessed September 11, 2008).

Emery, Charles. *Air Force Weather Agency WebPage.* May 2006. http://www.afweather.af.mil/shared/media/photodb/photos/051025-F-8732E-007.jpg (accessed September 11, 2008).

EWR Weather Radar Systems. "E600 Series II - Portable, Tactical Weather Radar System." *EWR Weather Radar Systems.* 2006. http://www.ewradar.com/library/images/E600.pdf (accessed September 11, 2008).

FMH-3. "Federal Meteorological Handbook No. 3 - Rawinsonde and Pibal

Observations." *Office Of The Federal Coordinator For Meteorological Services And Supporting Research.* Edited by Julian M., Jr. Wright. May 1997. http://www.ofcm.gov/fmh3/text/default.htm (accessed September 11, 2008).

Garey, M., and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* San Francisco: W.H. Freeman, 1979.

Holton, James R. *An Introduction to Dynamic Meteorology.* Edited by Renata Dmowska and James R. Holton. San Diego, California: Academic Press, 1992.

Ignizio, James P., and Lyn C. Thomas. "An Enhanced Conversion Scheme for Lexicographic, Multiobjective Integer Programs." *European Journal of Operational Research* (Elsevier Science Publishers B.V.) 18, no. 1 (October 1984): 57 - 61.

JPub 2-01.3. "Joint Publication 2-01.3 - Joint Tactics, Techniques, and Procedures for Joint Intelligence Preparation of the Battlespace." *Defense Technical Information Center.* May 24, 2000. http://www.dtic.mil/doctrine/jel/new_pubs/jp2_01_3.pdf (accessed September 12, 2008).

JPub 3-0. "Joint Publication 3-0 Joint Operations." *Defense Technical Information Center.* February 13, 2008. http://www.dtic.mil/doctrine/jel/new_pubs/jp3_0.pdf (accessed September 12, 2008).

JPub 3-59. "Joint Doctrine, Tactics,Techniques, and Procedures for Meteorological and Oceanographic Operations." *Defense Technical Information Center.* Edited by Department of Defense. September 24, 2008. http://www.dtic.mil/doctrine/jel/new_pubs/jp3_59.pdf (accessed January 13, 2009).

Kuykendall, Tabitha. *Airman Magazine Online.* May/June 2007. http://www.airmanonline.af.mil/shared/media/photodb/photos/080224-F-0232K-020.JPG (accessed September 12, 2008).

Montgomery, Douglas C. *Design and Analysis of Experiments.* 6. Hoboken, New Jersey: John Wiley & Sons, Inc., 2005.

National Weather Service. *National Weather Service - Eastern Region Headquarters.* May 22, 2002. http://www.erh.noaa.gov/aly/COOP/Equipment/standard_raingauge.jpg (accessed September 12, 2008).

Pizzocaro, Diego, Stuart Chalmers, and Alun Preece. *Sensor Assignment in Virtual*

Environments Using Constraint Programming. Vol. XV, chap. 8 in *Applications and Innovations in Intelligent Systems XV*, by Richard Ellis, Tony Allen and Miltos Petridis, 333-338. London: Springer London, 2007.

Ramadan, Rabie, Khaled Abdelghany, and Hesham El-Rewini. "SensDep: A Design Tool for the Deployment of Heterogeneous Sensing Devices." *Proceedings of the Second IEEE Workshop on Dependability and Security in Sensor Networks and Systems.* Washington, District of Columbia: IEEE Computer Society, 2006. 44 - 53 .

REI Outdoor. *www.rei.com.* September 12, 2008. http://media.rei.com/media/538399.jpg (accessed September 12, 2008).

Rowaihy, H., et al. "A Survey of Sensor Selection Schemes in Wireless Sensor Networks." *SPIE The International Society For Optical Engineering.* San Jose: International Society for Optical Engineering, 2007. 65621A-65621A.

Sherali, Hanif D. "Equivalent Weights for Lexicographic Multi-Objective Programs: Characterizations and Computations." *European Journal of Operations Research* (North-Holland Publishing Company), no. 11 (1982): 367-379.

Shih, Kuei-Ping, Yen-Da Chen, Chun-Wei Chiang, and Bo-Jun Liu. "A Distributed Active Sensor Selection Scheme for Wireless Sensor Networks." *11th IEEE Symposium on Computers and Communications, 2006.* Cagliari, Sardinia, Italy: IEEE Computer Society, 2006. 923- 928.

Sniper's Paradise. *Sniper's Paradise Web Site.* 2006. http://www.snipersparadise.com/equipment/pics/leica.1.jpg (accessed September 11, 2008).

Snyman, Jan A. *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-based Algorithms.* Cambridge, Massachusetts: Springer, 2005.

Tzu, Sun. *The Illustrated Art of War.* Translated by Samuel B. Griffith. New York, New York: Oxford University Press, 2005.

United States Air Force. *Air Force Weather Agency Webpage.* May 11, 2008. http://www.afweather.af.mil/shared/media/photodb/photos/030319-F-9999S-001.jpg (accessed September 11, 2008).

Xu, Xiaochun, and Sartaj Sahni. "Approximation Algorithms for Wireless Sensor Deployments." *Dr. Sartaj K Sahni's Web Site - University of Florida.* April 21, 2006. http://www.cise.ufl.edu/~sahni/papers/sensorDeployment.pdf (accessed September 10, 2008).

Vita

Captain Andrew J. Geyer graduated from Minot High School in Minot, North Dakota in May 1996. He entered undergraduate studies at North Dakota State University in Fargo, North Dakota where he graduated with a Bachelor of Science degree in Physics in May 2000. He was commissioned through Detachment 610, AFROTC at North Dakota State University.

His first assignment was to the Air Force Institute of Technology's Basic Meteorology Program at Texas A&M University in College Station, Texas. In September 2001, Capt Geyer was assigned to the 18th Weather Squadron at Fort Bragg, NC, where he served combat tours in Operations Enduring Freedom and Iraqi Freedom as a Combat Weather Team leader attached to the U.S. Army's 82nd Airborne Division. In May 2004, Capt Geyer reported to Kunsan Air Base, Republic of Korea, where he served as the weather flight commander for the 8th Fighter Wing. In July 2005, he reported to Fort Benning, Georgia where he served as Commander, Detachment 4, 10th Combat Weather Squadron and as the Staff Weather Officer for the U.S. Army's 75th Ranger Regiment. During that assignment, Capt Geyer deployed as the Joint METOC Officer assigned to Combined Joint Special Operations Task Force-Arabian Peninsula in Iraq. In August 2007, he entered the Graduate School of Engineering and Management, Air Force Institute of Technology. Upon graduation, he will be assigned to Headquarters, Air Force Weather Agency at Offutt Air Force Base, Nebraska.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
16-03-2009	**Master's Thesis**	Aug 2007 - Mar 2009

4. TITLE AND SUBTITLE	
OPERATIONS-FOCUSED OPTIMIZED THEATER WEATHER SENSING STRATEGIES USING PREEMPTIVE BINARY INTEGER PROGRAMMING	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	**5d. PROJECT NUMBER**
Geyer, Andrew J., Captain, USAF	**5e. TASK NUMBER**
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)	**8. PERFORMING ORGANIZATION REPORT NUMBER**
Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way, Building 642 WPAFB OH 45433-7765	AFIT/GOR/ENS/09-06
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSOR/MONITOR'S ACRONYM(S)**
Air Force Special Operations Command Director of Weather (AFSOC/A3W) Attn: Maj Jack Evans 100 Bartley St, Suite 153W DSN: 579-2502 Hurlburt Field, FL 32544 e-mail: jack.evans@hurlbut.af.mil	AFSOC/A3W
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Throughout history, successful military leaders have recognized that weather conditions on the battlefield can play a significant role in determining the victor. For this reason, the United States maintains and equips several different types of military units that are tasked to provide dedicated weather support to operational commanders. These units use a variety of types of sensors with differing capabilities to collect current weather conditions on the battlefield. In support of the commander's Concept of Operations (CONOP), United States military doctrine dictates that, as a part of general military campaign planning, a Meteorological and Oceanographic (METOC) collection plan be developed. This collection plan should specify the allocation of all weather sensing sources within the operational theater and throughout all phases of the military operation. This paper describes a methodology for creating a robust METOC collection plan in support of any given military campaign plan that deploys air, land, maritime, and special operations component weather sensing equipment in a way that maximizes expected detection of operationally significant weather conditions over the largest area possible for a given set of available weather sensors.

15. SUBJECT TERMS
Sensor Selection Problem, METOC Collection Plan, Weather Sensing Strategy, Preemptive Binary Integer Programming

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Shane N. Hall, Maj, USAF (ENS)
					19b. TELEPHONE NUMBER *(Include region code)*
U	U	U	UU	131	DSN: 785-3636 x4624; e-mail: Shane.Hall@afit.edu

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18